ReFer: Retrieval-Enhanced Vertical Federated Recommendation for Full Set User Benefit

Wenjie Li
Tsinghua Shenzhen International Graduate School, Tsinghua University
Shenzhen, China
Meituan
Shanghai, China
liwj20@mails.tsinghua.edu.cn

Zhongren Wang
Tsinghua Shenzhen International Graduate School, Tsinghua University
Shenzhen, China
Meituan
Shanghai, China
wcr23@mails.tsinghua.edu.cn

Jinpeng Wang
Tsinghua Shenzhen International Graduate School, Tsinghua University
Shenzhen, China
wjp20@mails.tsinghua.edu.cn

Shu-Tao Xia*
Tsinghua Shenzhen International Graduate School, Tsinghua University
Shenzhen, China
Research Center of Artificial Intelligence, Peng Cheng Laboratory
Shenzhen, China
xiast@sz.tsinghua.edu.cn

Jile Zhu
Mingjian Chen
Meituan
Shanghai, China
zhujile@meituan.com
chenmingjian@meituan.com

Jiangke Fan
Jia Cheng
Jun Lei
Meituan
Shanghai, China
jiangke.fan@meituan.com
jia.cheng.sh@meituan.com
leijun@meituan.com

ABSTRACT

As an emerging privacy-preserving approach to leveraging cross-platform user interactions, vertical federated learning (VFL) has been increasingly applied in recommender systems. However, vanilla VFL is only applicable to overlapped users, ignoring potential universal interest patterns hidden among non-overlapped users and suffers from limited user group benefits, which hinders its application in real-world recommenders.

In this paper, we extend the traditional vertical federated recommendation problem (VFR) to a more realistic Fully-Vertical federated recommendation setting (**Fully-VFR**) which aims to utilize all available data and serve full user groups. To tackle challenges in implementing Fully-VFR, we propose a **R**etrieval-**e**nhanced Vertical **Fe**derated **r**ecommender (**ReFer**), a groundbreaking initiative that explores retrieval-enhanced machine learning approaches in VFL. Specifically, we establish a general "retrieval-and-utilization" algorithm to enhance the quality of representations across all parties. We design a flexible federated retrieval augmentation (*RA*) mechanism for VFL: *(i) Cross-RA to complement field missing* and *(ii) Local-RA to promote mutual understanding between user groups.* We conduct extensive experiments on both public and industry datasets. Results on both sequential and non-sequential CTR prediction tasks demonstrate that our method achieves significant performance improvements over baselines and is beneficial for all user groups.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Online advertising**; • **Security and privacy** → *Privacy protections*; • **Computing methodologies** → *Cooperation and coordination.*

KEYWORDS

Vertical Federated Learning, Recommendation System, Online Advertising, Retrieval Augmentation Learning, Split Neural Network

ACM Reference Format:
Wenjie Li, Zhongren Wang, Jinpeng Wang, Shu-Tao Xia, Jile Zhu, Mingjian Chen, Jiangke Fan, Jia Cheng, and Jun Lei. 2024. ReFer: Retrieval-Enhanced Vertical Federated Recommendation for Full Set User Benefit. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657763

1 INTRODUCTION

Recommender system is an essential application in information retrieval [31, 44], which is quite ubiquitous in our daily lives, from reading news and watching movies to reviewing restaurants, online shopping, and identifying points of interest (POIs). In this context, user behaviors are recorded by many kinds of service platforms, which are diverse and complementary to depict user interests in different domains [43, 50]. Collectively leveraging these multi-platform user behaviors is helpful to achieve more fine-grained and comprehensive user interest modeling, which is beneficial to many tasks. However, it is nearly impossible to directly share raw data across platforms due to data privacy regulations [42] and commercial confidentiality among service agencies.

To address these issues, vertical federated learning [5, 30, 41, 45] has been adopted to utilize cross-platform attributes without compromising user privacy. It has been explored in various recommendation tasks [47], such as click-through prediction, conversion rate prediction, and item recommendation [7, 9, 12, 13]. In a typical

*Corresponding author.

SIGIR '24, July 14–18, 2024, Washington, DC, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0431-4/24/07.
https://doi.org/10.1145/3626772.3657763

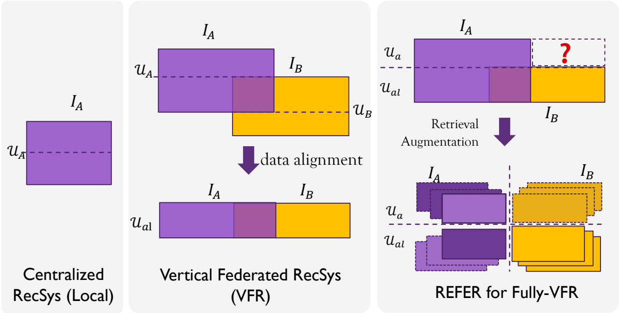

Figure 1: Conventional VFR setting only considers the overlapped dataset, suffering from a) incomplete user interest awareness and meanwhile b) cannot serve non-overlapped users. While extending the problem of VFR to Fully-VFR that serves for both user sets,we propose a retrieval-based federated framework to solve this problem.

VFL process, participants first execute a Private Set Intersection (PSI [33]) to obtain the aligned (i.e., overlapped) dataset and then perform distributed training. This essentially limits the models to only train and infer on aligned users. This constraint of ***narrowed data scope*** renders VFL impractical in two ways:

- **Incomplete interest awareness**: Aligned users for dissimilar businesses are often limited and constitute only a small portion of the user population. This reduced training set size can increase the risk of overfitting and result in low-quality embeddings and hidden representations, especially in sparse, high-dimensional recommendation datasets. In particular, item interactions of unaligned users are completely ignored, despite their potential to enhance item representations.
- **helpless for unaligned users**: The intrinsic field missing in passive parties makes it impossible for a federated model to make predictions for unaligned users, further undermining VFL's practicability. If a participant holds more unaligned users than aligned users, or weights unaligned users more heavily in their business, there may be insufficient motivation to join the federation. The expected performance gains on the aligned user set are not useful in this case. Although default filling with default values can alleviate this problem, it is superficial and fails to offer a meaningful information supplement.

In this paper, we propose a retrieval-enhanced approach named **ReFer** to tackle these issues (as depicted in Figure 1). Specifically, we design two kinds of retrieval augmentation strategies:

- **Cross-RA for field missing** For a target user in the unaligned user set, we retrieve relevant users in the aligned user set (only depending on their characters in the active party's domain) and then use the corresponding features in passive party's domain as supplemental features to fill the information missing gap.
- **Local-RA for mutual understanding** To enhance mutual correlation learning among aligned and unaligned users, we further retrieve relevant users in the entire user set for each user to enhance representations in the active party's feature domain.

To achieve these strategies, we devised a two-stage federated retrieval mechanism to conduct privacy-preserved distributed data augmentation. And proposed attention-based fusion modules to learn enhanced representations. With ReFer , training, and inference can be carried out for the full user set, while representation in all domains can be enhanced. In summary, our contributions are:

(1) We proposed the first retrieval-based algorithm for vertical federated learning to enable full user set modeling and enhanced representation learning.
(2) We proposed a general and effective "retrieve-and-fusion" framework to achieve both Local-side RA and Cross-side RA, which systematically enhances all parties' representation and promotes better predictions.
(3) We conducted extensive experiments on both public and industry datasets under both sequential and non-sequential click-through rate estimation tasks. The result shows that our method achieves significant performance lift against baseline models.

Table 1: The data utilization scopes of different methods. "UN-A" represents the unaligned data of party A, and "AL-B" represents the B-side part of aligned data. The text in brackets indicates how the method addresses the issue of missing B-side fields in inference.

	Training Stage		Serving Stage	
Method	UN-A	AL-B	UN-A	AL-B
Local	✓		✓	
Fed		✓		✓
FTL	✓	✓	✓	
FPD	✓	✓	✓	
Fed-Fill	✓	✓	✓(zero-filling)	✓
FedCVT	✓	✓	✓(hidden-syn)	✓
ReFer	✓	✓	✓(raw-factual)	✓

2 RELATED WORK

Retrieval Enhanced Machine Learning: The idea of retrieval-enhancement machine learning was first introduced in open-domain question answering [6, 15, 18, 21, 38] and have since been continuously adopted in large language modeling [3, 10, 19–21, 35, 37]. Inspired by these success of retrieval enhancement in natural language processing (NLP), [2, 26, 34] have adopted it in recommendation tasks. They design tailored retrievers for the recommender to search for relevant samples or users for data augmentation, thus utilizing cross-sample correlation to enhance user representation. For a more comprehensive understanding, interested readers can refer to a recent survey paper [48]. Our paper focuses on utilizing REML to address the challenges associated with VFL, which has not been considered in the works mentioned above.

Vertical Federated Recommendation Despite the popularity of designing federated recommender systems in horizontal FL setting [27, 28, 32, 47], such efforts on vertical FedRec [47] are still unsatisfactory [30, 45, 46]. With the similar purpose of tackling the field missing in VFL, FedCVT [17] complements both the missed representations and labels for unaligned samples. While this solution comprehensively uses all available data, it is computationally inefficient and not tailored for recommendation tasks. Three distillation-based works [23, 24, 36] propose to decouple the dependence of online serving with federation and meanwhile enable full user set inference. However, their goal of achieving local serving bans the use of B-side inputs completely for prediction, creating a significantly different and more challenging ill-pose setting from ours. [46] proposes a novel diffusion-based alternative training algorithm to utilize the unaligned data from active parties. However, its focus is only on improving performance for aligned samples, and it does not include unaligned samples in federated serving. A detailed comparasion of these works are summarized in Table 1. Despite their differences, our retrieval mechanism can be incorporated into these works to further enhance their performance.

3 PRELIMINARY

3.1 Concepts and Problem Formulation

We consider a typical VFR (**V**ertical **F**ederated **R**ecommender) system [5, 41] with an ***active party*** $\mathcal{A} = (\mathcal{U}_A, \mathcal{I}_A, \mathcal{X}_A, \mathcal{Y}_A)$ and a ***passive party*** $\mathcal{B} = (\mathcal{U}_B, \mathcal{I}_B, \mathcal{X}_B)$, where $\mathcal{U} = \{1, 2, \ldots, |\mathcal{U}|\}$, $\mathcal{I} = \{1, 2, \ldots, |\mathcal{I}|\}$ are id sets of users and items, $\mathcal{Y}_A = \{0, 1\}^{|\mathcal{U}_A| \times |\mathcal{I}_A|}$ is the label set of user-item interactions from the active party, and $\mathcal{X} = \{\mathbf{x}_{ij}\}_{i,j \in \mathcal{U} \times \mathcal{I}}$ is the feature set of interactions for each party. The active party is a recommender system that defines the learning task and holds labels and features, while the passive party is a data provider offering extra features about users, items, or contexts.

*Definition 3.1 (**Aligned Dataset**).* Given the aligned user set $\mathcal{U}_{al} = \mathcal{U}_A \cap \mathcal{U}_B$ and the union item set $\mathcal{I}_\cup = \mathcal{I}_A \cup \mathcal{I}_B$ of two parties, the aligned dataset is defined as $\mathcal{D}_{al} = \{\mathcal{D}^A_{al}, \mathcal{D}^B_{al}\}$, where:

$$\mathcal{D}^A_{al} = \{(\mathbf{x}^A, y^A)\}_{ij}, \mathcal{D}^B_{al} = \{\mathbf{x}^B\}_{ij};\;\; i \in \mathcal{U}_{al}, j \in \mathcal{I}_\cup \tag{1}$$

Here $\mathbf{x}^A_{ij} \in \mathcal{X}_A$ and $\mathbf{x}^B_{ij} \in \mathcal{X}_B$ are input vectors from two parties. Their field structure is introduced in detail in section 3.2.

*Definition 3.2 (**VFR Problem**).* A VFR system aims to train a recommender model by exploiting the aligned dataset $\mathcal{D}_{al} = \{\mathcal{D}^A_{al}, \mathcal{D}^B_{al}\}$ to make more comprehensive predictions for aligned users $\mathcal{U}_{al}$. The optimization goal and inference process are defined as:

$$\arg\min_{\theta_A,\theta_B} \mathcal{L}(\mathcal{Y}^A, f^{Fed}_{\theta_A,\theta_B}(\mathcal{U}_{al}, \mathcal{I}_\cup | \mathcal{D}^A_{al}, z(\mathcal{D}^B_{al}))) \tag{2}$$

$$\hat{y}^A_{ij} = f^{Fed}(i, j|\mathbf{x}^A_{ij}, z(\mathbf{x}^B_{ij})), \forall i, j \in \mathcal{U}_{al} \times \mathcal{I}_\cup \tag{3}$$

where $\mathcal{L}$ is the task loss function and $f^{Fed}_{\theta_A,\theta_B}$ is the distributed federated model partially owned by each party(see section 3.3 for details). $z(\cdot)$ is the data processing function transferring intermediate results between parties instead of the raw data. $\hat{y}^A_{ij}$ is the prediction. Both the model and data are private and invisible to other parties.

*Definition 3.3 (**Unaligned Dataset**[1]).* Given the unaligned users $\mathcal{U}_a = \mathcal{U}_A - \mathcal{U}_{al}$ of party $\mathcal{A}$, the unaligned dataset is defined as

$$\mathcal{D}^A_{un} = \{(\mathbf{x}^A, y^A)_{ij} | u \in \mathcal{U}_a, j \in \mathcal{I}_A\} \tag{4}$$

In practice, $\mathcal{D}_{al}$ is usually limited, and sometimes $|\mathcal{D}^A_{al}| \ll |\mathcal{D}^A_{un}|$, which restricts the effectiveness of VFR [17, 23, 24, 46]. Besides, VFR is not able to serve unaligned users regardless of its necessity and importance in real business. Thus, we define a more general problem setting:

*Definition 3.4 (**fully-VFR Problem**).* A fully VFR system aims to utilize the full labeled dataset $\mathcal{D}_{full} = \{\mathcal{D}^A_{un}, \mathcal{D}^A_{al}, \mathcal{D}^B_{al}\}$ to build a recommender model available **for all users** $\mathcal{U}_A = \{\mathcal{U}_a, \mathcal{U}_{al}\}$ **of the active party** to get better performance. The optimization goal and inference process are defined as :

$$\arg\min_{\theta_A,\theta_B} \mathcal{L}(\mathcal{Y}^A, f^{Fed}_{\theta_A,\theta_B}(\mathcal{U}_A, \mathcal{I}_\cup | \mathcal{D}^A_{un}, \mathcal{D}^A_{al}, z(\mathcal{D}^B_{al}))) \tag{5}$$

$$\hat{y}^A_{ij} = \begin{cases} f^{Fed}(i, j|\mathbf{x}^A_{ij}, z(\mathbf{x}^B_{ij})), & \forall i, j \in \mathcal{U}_{al} \times \mathcal{I}_\cup, \\ f^{Fed}(i, j|\mathbf{x}^A_{ij}, z'(\mathcal{D}^B_{al})), & \forall i, j \in \mathcal{U}_a \times \mathcal{I}_A. \end{cases} \tag{6}$$

where $z'(\cdot)$ is an alternative privacy-preserved processing function that extract i, j-relative information to assist the inference for unaligned users. For example in Table 1, Fed-Fill, FedCVT, and ReFer employ different strategies to implement this function, whereas FPD and FTL leave it as a null function.

As a core difference, VFR aims to learn a scoring function ***only work for aligned users***, while fully-VFR aims to learn a better scoring function ***work for all users*** and targeted to perform ***better than both local model and VFR model***.

3.2 Data Scenarios and Task Types

3.2.1 ***Attr-VFR Task***. Given two parties with identical item set $\mathcal{I}_A = \mathcal{I}_B = \mathcal{I}_\cup$, the **attr**ibute-**VFR** task aims to provide a better scoring prediction for a pair of user and candidate item (i, j) by utilizing ***additional data attributes*** from the passive party. Thus, a federated sample is a triplet $(\mathbf{x}^A_{ij}, y^A_{ij}, \mathbf{x}^B_{ij})$ as

$$\mathbf{x}^A_{ij} = [\mathbf{u}^A_i, \mathbf{v}^A_j, \mathbf{c}^A_{ij}], \forall i \in \mathcal{U}_A;\;\; \mathbf{x}^B_{ij} = [\mathbf{u}^B_i, \mathbf{v}^B_j, \mathbf{c}^B_{ij}], \forall i \in \mathcal{U}_{al} \tag{7}$$

where $j \in \mathcal{I}_\cup$, and $\mathbf{u}, \mathbf{v}, \mathbf{c} \in \mathcal{X}$ are concatenated vectors of values in attribute subsets of users, items, and context information. Attr-VFR stems from industry vertical federated advertising scenarios [46].

3.2.2 ***Seq-VFR Task***. Given two parties with non-identical item sets $\mathcal{I}_A \neq \mathcal{I}_B$, we define the behaviour-**seq**uence-**VFR** task which aims to improve the scoring for a time-identified pair of user and candidate item (i, j, t) by leveraging additional ***cross-view user behavior sequence*** from the passive party, where

$$\mathbf{x}^A_{ijt} = [\mathbf{u}^A_i, \mathbf{v}^A_j, \mathbf{s}^A_{it}], \forall i \in \mathcal{U}_A;\;\; \mathbf{x}^B_{it} = [\mathbf{u}^B_i, \mathbf{s}^B_{it}], \forall i \in \mathcal{U}_{al} \tag{8}$$

$$\mathbf{s}^A_{it} = \{\mathbf{v}^A_l \,| l \in \mathcal{I}^{i,t'}_A\};\;\; \mathbf{s}^B_{it} = \{\mathbf{v}^B_l \,| l \in \mathcal{I}^{i,t'}_B\};\;\; t' \leq t \tag{9}$$

where $j \in \mathcal{I}_A$ is the candidate item, $\mathbf{s}^A_{it}$ and $\mathbf{s}^B_{it}$ are the time-ordered user behaviour sequence vector, $\mathcal{I}^{i,t'}_A$ and $\mathcal{I}^{i,t'}_B$ are the item index sets interacted by user i before timestamp t in two domains. Note that the candidate item always comes from party A, while the item vector in the user behavior sequence comes from two parties. The potential for performance improvement derives from the diversity of user behavior sequences. Seq-VFR is a natural extension from the centralized sequential behavior CTR task [51] into a federated and multi-view setting.

[1] Here we refer the term in particular to party $\mathcal{A}$'s unaligned dataset. While one can analogously define $\mathcal{U}_b$ and unaligned dataset $\mathcal{D}^B_{un}$ for party $\mathcal{B}$, they are naturally excluded in our setting since they are out of the business service of the active party. We leave this minor extension as future work.

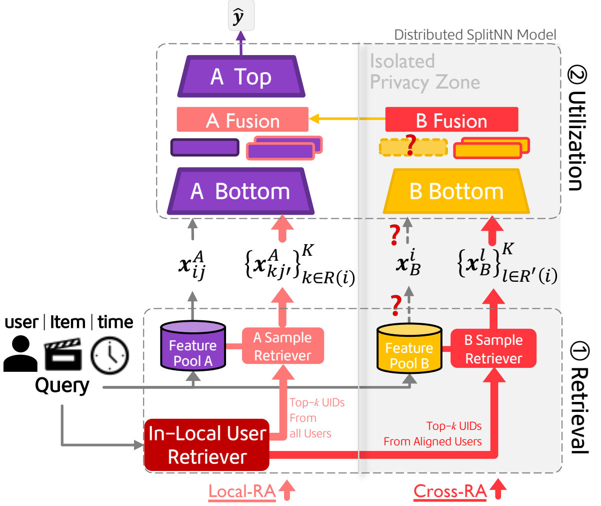

Figure 2: ReFer enables full-set user training and inference by enhancing representation learning on all parties. It contains a highly flexible federated retrieval mechanism and query-aware federated fusion modules.

3.3 Backbone Model: Vertical SplitNN

We follow the typical vertical Split Neural Network(SplitNN) [40] as the backbone model. Specifically, each party holds a ***bottom model*** (e.g., f_A, f_B) for extracting hidden representations, and the active party additionally holds a ***top model*** g_A to aggregate two sides of representations to make predictions $\hat{y}$ and computes loss $\mathcal{L}(\hat{y}, y)$. As like

$$\mathbf{h}_{ij}^A = f_A(\mathbf{x}_{ij}^A;\theta_A^1), \mathbf{h}_{ij}^B = f_B(\mathbf{x}_{ij}^B;\theta_B) \tag{10}$$

$$\hat{y}_{ij} = f^{Fed}_{\theta_A,\theta_B}(\mathbf{x}_{ij}^A, \mathbf{x}_{ij}^B) = g_A(\mathbf{h}_{ij}^A, \mathbf{h}_{ij}^B; \theta_A^2) \tag{11}$$

where $\theta_A = \{\theta_A^1, \theta_A^2\}$. Note that $\mathbf{h}_A$ and $\mathbf{h}_B$ are distributively computed in each party's server. Once $\mathbf{h}_B$ is ready, it will be sent to party A across the network. In the backward pass, the gradient $\mathbf{g}_B = \nabla_{\mathbf{h}_B}\mathcal{L}$ of $\mathbf{h}_B$ will be sent to party B for subsequent backward pass and parameter updates. *Only hidden representation* $\mathbf{h}_B$ *and corresponding gradient* $\mathbf{g}_B$ *are shared during training, thus original data is never directly exposed.* We consider another line of security-focused works [1, 22, 49] complementary to us and we focus on the aspect of effectiveness.

4 METHODOLOGY

As depicted in Figure 2, the overall framework is two-staged, consisting of *(1) Federated Retrieval Augmentation* and *(2) Federated Retrieval Utilization.*

4.1 Hierarchical Federated Retrieval

Given a query sample $\mathbf{x}_{ij}^A, i, j \in \mathcal{U}_A \times \mathcal{I}_A$ from the active party A, the federated retrieval $\mathcal{R}$ aims to return K relevant augmentation samples $\{\mathbf{x}_A^{k,j}\}_{k\in\mathcal{R}(i)}^K$ and $\{\mathbf{x}_B^{l,j}\}_{l\in\mathcal{R}'(i)}^K$ for both parties. The two sub-processes that retrieve data from two parties are called the *Local-RA* and *Cross-RA*. To implement an efficient and privacy-preserving retrieval mechanism that adheres to the principle of VFL, we divide the overall retrieval process $\mathcal{R}$ into two stages of user retriever $\mathcal{R}_u$ and sample retriever $\mathcal{R}_s$ where

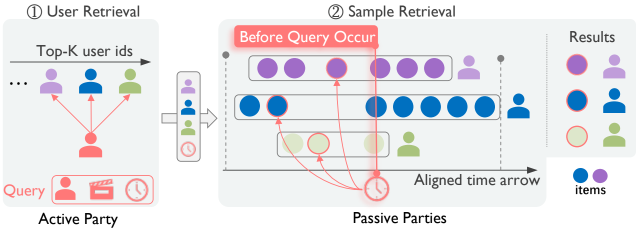

Figure 3: The federated retrieval process for cross-party retrieval augmentation (Cross-RA).

$$\mathcal{R}(i,j,t) = \mathcal{R}_s(\mathcal{R}_u(i;\mathbf{E}), j, t; \mathcal{D}_A, \mathcal{D}_{al}^B) \tag{12}$$

Here $\mathcal{R}_u$ and $\mathcal{R}_s$ denote the user and sample retriever functions respectively. $\mathbf{E}$ is the parameter of $\mathcal{R}_u$ while $\mathcal{R}_s$ is a non-parametric function. Both retrieval stages are conducted locally within each party to avoid cross-party raw data transfer, thereby ensuring privacy preservation and efficiency. Besides, only a subset of user-ids from the aligned user set and the interaction time of query item j are transferred across parties. This information is common knowledge to both parties and involves minimal communication burden.

4.1.1 ***In-Local User Retrieval.*** Given a query user id $i \in \mathcal{U}_A$, a candidate user-id pool $\mathcal{P}$ and a similarity function $sim(\cdot)$, the user retriever in (and only in) the active party returns the ids of the top-K similar users of i from $\mathcal{P}$, as like

$$\tilde{\mathcal{U}}_{\mathcal{P}}^i = \mathcal{R}_u(i;\mathbf{E}(\mathcal{P})); \quad |\tilde{\mathcal{U}}_{\mathcal{P}}^i| = K \tag{13}$$

$$\text{s.t. } sim(\mathbf{E}_i,\mathbf{E}_l) > sim(\mathbf{E}_i,\mathbf{E}_{l'}), \quad \forall l,l' \in \tilde{\mathcal{U}}_{\mathcal{P}}^i \times (\mathcal{P} - \tilde{\mathcal{U}}_{\mathcal{P}}^i) \tag{14}$$

Here $\mathbf{E} \in \mathbb{R}^{|\mathcal{U}_A| \times d}$ is a user embedding matrix, $\mathbf{E}(\mathcal{P})$ is the submatrix indexed by $\mathcal{P}$ and $\mathbf{E}_i$ is the embedding vector of user i. $\tilde{\mathcal{U}}_{\mathcal{P}}^i$ denotes for the result user id set. We separately use $\mathcal{U}_A$ and $\mathcal{U}_{al}$ as candidate pools for Local-RA and Cross-RA, and corresponding result sets are denoted as $\tilde{\mathcal{U}}_A^i$ and $\tilde{\mathcal{U}}_{al}^i$. The underlying reasons are elaborated as follows.

- **Local-RA with $\mathcal{P} = \mathcal{U}_A$**
  - *availability:* The ultimate goal of Local-RA is to obtain data from $\mathcal{X}_A$, which is available for all users. All users are useful candidates for this purpose.
  - *purpose:* Employing the complete set $\mathcal{U}_A$ as a candidate pool enables mutual retrieval between aligned and unaligned user sets. It ensures that regardless of the origin of the query (e.g., from $\mathcal{U}_a$ or $\mathcal{U}_{al}$), relevant users from the other set always stand a chance to be retrieved. When this strategy is further incorporated into the fusion learning module, it enhances mutual comprehension between user groups and alleviates the group bias.
- **Cross-RA with $\mathcal{P} = \mathcal{U}_{al}$**
  - *availability:* The ultimate goal of Cross-RA is to obtain data from $\mathcal{X}_B$, which is exclusively available to aligned users. Thus, only aligned users are available candidates for this purpose.

- *purpose:* For an unaligned user, we use similar aligned users as a bridge to access $\mathcal{X}_B$, thereby addressing the field missing problem in the raw data space. For an aligned user, augmentation with similar aligned users enables the utilization of cross-user correlation in field domain $\mathcal{X}_B$, which benefits representation learning.

4.1.2 ***Construction of User Query Embedding.*** To ensure semantically meaningful retrieval, it's crucial to use embedding vectors that accurately capture user similarity. Just as a pre-trained BERT model in NLP is recognized as a general sentence encoder, CF-based (collaborative-filtering) user embedding is acknowledged as an effective representation to depict user interests.

To make a practical and effective solution that can be readily applied in the industry, we pre-train an NCF model [11] to derive user embedding, relying solely on A-side user-item interaction data and excluding any side information.

$$\arg\min_{\mathbf{E},\boldsymbol{\Psi}} \mathcal{L}(y_{ij}, f^{NCF}(i, j; \mathbf{E}, \boldsymbol{\Psi})); \quad i, j \in \mathcal{U}_A \times \mathcal{I}_r \tag{15}$$

where f^{NCF} is the NCF model, $\mathbf{E}$ denotes the user embedding table and $\boldsymbol{\Psi}$ represents the set for all other model parameters. Only $\mathbf{E}$ is used by user retriever $\mathcal{R}_u$. $\mathcal{I}_r$ denotes the item set used for training user retriever, which should be available in the active party but not limited to be identical to $\mathcal{I}_A$.

We validate that the common id-based embedding is sufficiently enough to derive meaningful retrieval results. Despite our specific choice, our framework is flexible to other enhanced complex user embeddings. For example, one can use a totally different item domain (e.g., $\mathcal{I}_r \cap \mathcal{I}_A = \emptyset$) for informative diversity, additionally utilize side-information attributes $\mathcal{X}$ for richer representation, or adopt other kind of model architectures (e.g., MF, NGCF) tailored for their business. As a common and effective way in recommendation [2] and NLP [10], we use the inner product to measure the similarity between two users i and l:

$$sim(\mathbf{E}_i, \mathbf{E}_l) = \mathbf{E}_i \cdot \mathbf{E}_l \tag{16}$$

And to efficiently get Top-K ranked users, we adopt the Maximum Inner Product Search (MIPS) algorithm for accelerating the user retrieval. It has a sub-linear time complexity $O(log(|\mathcal{P}|))$ over the size of the candidate pool.

4.1.3 ***Distributed Sample Retrieval.*** Once we get the neighbor user set $\tilde{\mathcal{U}}^i_{\mathcal{P}}$ for a given query triplet (i, j, t), we use the sample retriever to acquire top-K related items. For a neighbour user l and a target item domain $C \in \{A, B\}$, we have:

$$\tilde{\mathcal{I}}^{lt}_C = \mathcal{R}_s(l, t; \mathcal{I}^{lt}_C); \quad l \in \tilde{\mathcal{U}}^i_{\mathcal{P}}, \quad |\tilde{\mathcal{I}}^{lt}_C| = K \tag{17}$$

$$\text{s.t.} \quad sim_I(j, j') > sim_I(j, j''), \forall j', j'' \in \tilde{\mathcal{I}}^{lt}_C \times (\mathcal{I}^{lt}_C - \tilde{\mathcal{I}}^{lt}_C) \tag{18}$$

Here $\tilde{\mathcal{I}}^{lt}_C \subset \mathcal{I}^{lt}_C \subset \mathcal{I}^l_C \subset \mathcal{I}_C$ is the result item subset, which is interacted by user l before timestamp t and with top-K relevance to the query item j. Due to the privacy constraint of VFL and task discrepancy between Attr-VFR and Seq-VFR, the sample retrieval of Local-RA and Cross-RA varies in multiple aspects: **(1) Local-RA with accessible $\mathbf{x}^A_{ij}$** Local-RA is conducted in the active party, thus with natural in-local data security, side-information $\mathcal{X}_A$ for both the query item j and candidate item set $\mathcal{I}^{lt}_A$ are available to use in the sample retrieval process of Local-RA. Besides, the process of Local-RA for both Attr-VFR and Seq-VFR remains the same. Furthermore, no network communication is needed in this process. **(2) Cross-RA with unaccessible $\mathbf{x}^A_{ij}, \mathbf{x}^B_{ij}$** Since Cross-RA is finally conducted in the passive party, the side information of the query is unavailable. $\mathbf{x}^A_{ij}$ can not be sent to party B for privacy protection reasons and $\mathbf{x}^B_{ij}$ is naturally missed for unaligned users. For Attr-VFR, item id j is common knowledge between parties and is thus available for retrieval. However, for Seq-VFR, only the interaction time t is available since the query item-id may not have corresponding field information in party B and is also confidential for party A. During the retrieval process, only the neighbor user set and interaction time need to be communicated, which are non-private information and involve a low communication payload.

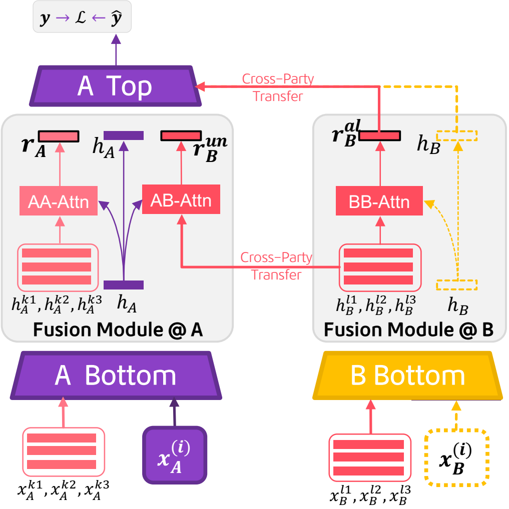

Figure 4: Architecture of Fusion Modules.

In our experiment, we use the closeness in time as the measure for item similarity, that is

$$sim_I(j, j') = e^{(t_{lj} - t_{lj'})^2} \tag{19}$$

Our experimental results show that even Top-1 time-latest item retrieval can yield satisfactory outcomes. Moreover, it only consumes log-level time complexity by adopting binary search on the time-ordered item sequence. Despite our choice, ReFer is also flexible to other kinds of sample retrievers.

4.2 Retrieval-Enhanced Modeling

4.2.1 ***Input Encoding.*** Once the retrieval is finished, we feed each query sample together with its retrieved samples into the model for enhanced learning. We reuse the embedding layer and bottom model (as stated in eq 10) to extract the initial representation for retrieved samples:

$$\tilde{\mathbf{H}}^A_{ij} = f_A(\mathbf{Z}^A_{ij}), \tilde{\mathbf{H}}^B_{ij} = f_B(\mathbf{Z}^B_{ij}) \tag{20}$$

Here $\mathbf{Z}$ and $\tilde{\mathbf{H}}$ are K-row matrices denoting the raw input and hidden features of query results for a given pair (i, j).

4.2.2 ***Query-Oriented Fusion.*** Now we can fuse the retrieved samples in the feature space. For the A-side segments of retrieved samples acquired by Local-RA, we use query-aware cross-attention to get a more compact fused representation:

$$\mathbf{r}_{ij}^{A} = attn_{A}(\mathbf{h}_{ij}^{A}, \tilde{\mathbf{H}}_{ij}^{A}\mathbf{W}); \quad \forall i, j \in \mathcal{U}_{A} \times \mathcal{I}_{A} \tag{21}$$

where $\mathbf{W} \in \mathbb{R}^{d_A \times d_A}$ is a learnable parameter adapting A-to-A domain transformation between query and retrieval results. The cross attention function $attn(\mathbf{a}, \mathbf{B})$ for a vector $\mathbf{a}$ and a value matrix $\mathbf{B} \in \mathbb{R}^{K \times d}$ is implemented as

$$attn(\mathbf{a}, \mathbf{B}) = \sum_{k=1}^{K} \alpha_k \mathbf{b}_k, \quad \alpha_k = \frac{exp(\mathbf{a}^{\top}\mathbf{B})}{\sum_{j=1}^{K} exp(\mathbf{a}^{\top}\mathbf{B})}, \tag{22}$$

where α_k is the attention weight. The subscript "A" in $attn_A$ indicates that the function is executed in party A. Analogously to Local-RA, we use a similar approach for Cross-RA in the passive party

$$\mathbf{r}_{ij}^{B} = \begin{cases} attn_{A}(\mathbf{h}_{ij}^{A}, \tilde{\mathbf{H}}_{ij}^{B} \cdot \mathbf{\Phi}), & \forall i, j \in \mathcal{U}_{a} \times \mathcal{I}_{A}, \\ attn_{B}(\mathbf{h}_{ij}^{B}, \tilde{\mathbf{H}}_{ij}^{B} \cdot \mathbf{\Theta}), & \forall i, j \in \mathcal{U}_{al} \times \mathcal{I}_{\cup}. \end{cases} \tag{23}$$

where $\mathbf{\Phi} \in \mathbb{R}^{d_B \times d_A}$ is the learnable parameter to capture B-to-A cross domain transformation while $\mathbf{\Theta} \in \mathbb{R}^{d_B \times d_B}$ is for B-to-B intra-domain transformation. It's worth noting that the modeling process for user groups differs significantly in three aspects:

- **Attention Key**: we adopt $\mathbf{h}_{ij}^{A}$ for unaligned users but $\mathbf{h}_{ij}^{B}$ for aligned users as the Attention key. Since $\mathbf{h}_{ij}^{B}$ does not exist for unaligned users, and thus $\mathbf{h}_{ij}^{A}$ is the only reliable, query-aware information we can utilize as the Attention key.
- **Learnable Parameter**: For unaligned users, to further mitigate the cross-domain discrepancy incurred by $\mathbf{h}_{ij}^{A}$, we introduce a different learnable parameter matrix $\mathbf{\Phi}$ to undertake the B-to-A domain mapping, ensuring the domain consistency of attentive correlation calculation. For aligned users, we also adopt a transformation, but only for fine-grained in-domain feature adoption.
- **Execution Party**: The B-side attention process for aligned users is conducted in-place in the passive party, while distributedly executed for unaligned users. Specifically, the retrieval feature matrix $\tilde{\mathbf{H}}_{ij}^{B}$ is firstly transformed in party B and then sent to party A for subsequent attentive fusion.

4.2.3 ***Prediction.*** After we get the fused representation for retrieved samples, we together feed them with the query sample's representations to the top model to get the final prediction:

$$\hat{y}_{ij}^{A} = \begin{cases} g_A([\mathbf{h}_{ij}^{A}; \mathbf{r}_{ij}^{A}; \vec{\mathbf{0}}; \mathbf{r}_{ij}^{B}]), & \forall i, j \in \mathcal{U}_{a} \times \mathcal{I}_{A} \\ g_A([\mathbf{h}_{ij}^{A}; \mathbf{r}_{ij}^{A}; \mathbf{h}_{ij}^{B}; \mathbf{r}_{ij}^{B}]), & \forall i, j \in \mathcal{U}_{al} \times \mathcal{I}_{\cup} \end{cases} \tag{24}$$

In the end, we calculate loss $\mathcal{L}(y^A, \hat{y}_A)$ in the active party with task-specific loss (e.g., cross-entropy loss for CTR task and mean square error for rating regression tasks). Throughout the training, we alternatively process unaligned and aligned data across epochs.

4.3 Practical Analysis

(1) $\{t, \tilde{\mathcal{U}}_{al}^{i}\}$ **in retrieval** are the only transmitted variables in the whole retrieval process. Here the timestamp t (usually at the date-level or hour-level) is used to avoid the time-traverse problem and conduct a time-related search in the retrieval process. $\tilde{\mathcal{U}}_{al}^{i}$ are just K user ID numbers from the aligned user set, which is common knowledge between the two parties. All these two kinds of variables do not violate privacy and only involve a small communication burden ($K + 1$ scalar values). **(2)** $\{\mathbf{r}^B, \mathbf{H}^B\}$ **in modeling** are the additional communicated variables introduced by ReFer in modeling process. They are isomorphic hidden vectors similar to $\mathbf{h}^B$, maintaining the same privacy protection level. Considering a $1:1$ ratio of aligned and unaligned users, They only involve $(K+1)/2$ vectors' communication burden. In a word, they maintain the same security level and only involve a few communication burdens. **(3) retrieval efficiency** The overall retrieval process only exhibits logarithmic $O(log(|\mathcal{U}|)) + O(log(|\mathcal{I}|))$ time complexity, which is highly efficient when dealing with millions of candidates. It is as efficient as a typical matching sub-system in a standard recommender.

Table 2: Statistics of all datasets and federated scenarios. y% means the ratio of positive label.

Dataset	Scenarios	User Group	#Users	#Items	#Samples|y%
Douban	book-movie	unaligned	344	2,478	1,776|0.69
		aligned	937	20,360	17,008|0.69
	book-music	unaligned	763	8,842	9,106|0.69
		aligned	518	14,283	9,691|0.69
	movie-book	unaligned	1,254	5,776	23,190|0.59
		aligned	1,234	15,200	26,378|0.57
	movie-music	unaligned	1,689	7,028	32,258|0.58
		aligned	799	10,298	17,277|0.57
ML-1M	Task1	unaligned	1,551	1,301	7,736|0.64
		aligned	2,529	3,460	58,554|0.57
	Task2	unaligned	1,719	1,170	4,949|0.64
		aligned	1,870	3,449	26,601|0.55
	Task3	unaligned	1,401	1,179	4,248|0.65
		aligned	2,536	3,477	43,700|0.58
Meituan	Search-B	unaligned	337,733	4618	1,625,741|0.01
		aligned	654,671	4772	2,382,273|0.02
	Browse-S	unaligned	340,755	4633	1,672,070|0.01
		aligned	651,649	4763	2,335,944|0.02

5 EXPERIMENTS

We conduct experiments aiming to answer the following questions:

- **RQ1**: How does ReFer perform compared to baselines?
- **RQ2**: How does ReFer perform on aligned/unaligned users?
- **RQ3**: How do different parts of ReFer contribute to the performance?

5.1 Experimental Settings

5.1.1 ***Dataset.*** We conduct experiments on two public datasets (Seq-VRec task) and one industry dataset (Attr-VRec task). Table 2. shows detailed statistics for all datasets. **(1) ML1M-Fed.** Using the original MovieLens1M dataset, we randomly divide it into two halves to create two parties, based on the movie genres field

(18 types in total). We generated three versions for party division and named their corresponding learning tasks Task1 through Task3. We follow [51] to transform the original rating data suitable for CTR prediction task. The final data fields include {movie_id, movie_cate_id, and rate_date} for the candidate movie and user-rated movie sequence. We use a sliding window on the time-sorted movie list of the user to generate samples. In each window, the final movie is treated as the candidate while the preceding movies are treated as historical behavior sequences. **(2) Douban-Fed.** The Douban dataset [50] is a natural multi-domain rating dataset containing 3 correlated item domains: movie, book, and music. We simulate 4 federated datasets by picking the 2 major domains movie and book as the active party and the rest 2 minor domains as passive parties. We preprocess raw data the same as ML1M-Fed to generate a sequential ctr prediction dataset. **(3) Meituan-Fed.** Meituan dataset is a click-through rate (CTR) dataset collected from a real-world advertising platform. The dataset comprises transaction records spanning nine days from Meituan platform[2]. It includes user profiles, item profiles, and two interaction domains: search actions and rec actions. We organize attributes of "user profile + item profile + search" into the **Search** domain, and attributes of rec actions into the **Rec** domain.

5.1.2 *Train & Test & Pool Splitting.*

For the two public datasets (Seq-VFR setting), we use ratio segmentation $[0.7|0.2|0.1]$ to split user behavior sequences for training, validation, and test. We use the training sequence as a retrieval pool for all parts of data and only use samples that occur before the query sample to avoid time traversing. For the meituan dataset (Attr-VFR setting), we use day segmentation $[2 \sim 6|7|8]$ for training/validation/test set and use day $[T-1|6|6]$ as retrieval pools for them to avoid future data leakage. Specifically, for a certain day in training or test data, only the past day's data serves as the retrieval pool.

5.1.3 *Evaluation Protocol.*

metrics: We use AUC and logloss to measure the performance for the CTR prediction task. We evaluated all methods on the full user set, the aligned user set, and the unaligned user set, respectively. For methods that originally ignored the unaligned dataset (e.g., Fed), we employed hidden feature zero-filling during inference to make them compatible.

5.1.4 *Baselines.*

We set up six methods for comparison:
Basic Group: (1) Local model is trained solely on the active party's features, without incorporating any attributes from the passive party. UN-Local and AL-Local are two variants of the Local model, each utilizing data from only one user group: UN-Local pertains to unaligned users, and AL-Local pertains to aligned users. **(2) Fed** This is a VFL model trained on aligned records. It only covers the aligned user set. **(3) Fed-Fill** This further exploits the active party's unaligned records, in which the missing fields of the passive party are filled with zeros.
Advanced Group: (4) FTL [29] is an end-to-end FL method for transferring knowledge to local samples. For shared samples, FTL maps different parties' raw features to a common feature space for knowledge transfer. **(5) FPD** Two pioneer works [24, 36] have adopted privileged distillation to achieve inference for all users. Here we collectively call them FPD methods. They learn a federated model on shared samples and then learn a local model (for local samples) by considering both ground-truth labels and soft labels produced by the federated model. **(6) FedCVT** [17] uses a similarity function to generate unaligned samples' hidden features. It combines unlabeled unaligned samples and labeled aligned samples with semi-supervised learning in a co-training fashion. We adopt it into our problem setting and re-implement it in a similar approximate way as in [46].

5.1.5 *Implementation Details.*

(1) retrieval: For public datasets, we use leave-out historical rating data to train user embeddings. For Meituan dataset, we use pre-trained common user embeddings in Meituan platform. To simplify the experimentation process, we pre-retrieved results for all records using FAISS [8, 16]. **(2) model structure**: We use the same model structure for all datasets, where bottom models use a 2-layer MLP with $64 \rightarrow 32$ units and the top models use a 2-layer MLP with $d_{cat} \rightarrow 16 \rightarrow 1$ units. Here d_{cat} is the summed size of all vectors produced by bottom models. Embedding dimensions for all fields are set to 10. **(3) training**: We use Adam optimizer with L_2 regularization to train models for 50 epochs, coupled with early stopping with a patience of 5 epochs. Batch sizes for Meituan, ML-1M, and Douban are 10000, 1000, and 512 respectively. **(4) hyper-parameter**: Grid search is used to find the best hyper-parameters, where $\eta \in \{0.005, 0.001, 0.0005, 0.0001\}$ and $\lambda_{L2} \in \{0.01, 0.001, 0.0001\}$. Finally, we report the mean results of the searched best hyper-parameter, under 3 random seeds.

5.2 Analysis on Different Baselines(RQ1)

Focusing on full user set performance shown in Table 3, ***ReFer outperforms all baselines on full-user-set performance*** across both metrics and all data scenarios, regardless of the significantly varied data discrepancies, including aligned/unaligned user ratios, VFR task types, and label ratios (as depicted in Table 2). For the aspect of efficiency, we employ approximate MIPS algorithms (e.g., ALSH) to achieve sub-linear time complexity. In our experiments, we use GPU-accelerated FAISS [16] as the retrieval engine. It only takes an average of ***1.02ms for per 1000-sample*** bunch query for Top-5 retrieval with a 32-d embedding vector. This is efficient and feasible in the industry RecSys. Compared to vanilla VFL, ReFer primarily involves extra millisecond-level time consumption.

Besides, we get some key findings by analyzing basic baselines: **(1) Discarding unaligned samples significantly harms the performance across the entire user set**: Local outperforms Fed on 9/9 scenarios for AUC and 7/9 scenarios on Logloss. By additionally observing Table 4 for that the AUC of aligned users on Fed is higher than AL-Local in 8/9 scenarios except for Movie-Book, we can see the extra fields are all useful. These two phenomenon jointly show that the benefit of introducing extra B-side fields is less than the lost benefit caused by neglecting unaligned samples. This significantly identifies the importance of extending VFR to Fully-VFR setting. **(2) Naive 0-filling is not sufficient**: Fed-Fill additionally uses B-side fields compared to Local, but it still underperforms Local in 2/9 scenarios on AUC (-0.0014 on Book-Music and -0.0013 on Movie-Book) and 2/9 on Logloss (+0.0009 on Movie-Book and +0.0002 on Movie-Music). Despite the only cases on Movie-Book where the B-side field is negative, there are still two cases showing Fed-Fill's

[2] https://www.meituan.com

Table 3: Overall results on full user set. ReFer achieves the best result in all scenarios on both AUC and logloss(denoted as NLL). The results are evaluated on full user sets. Green cells denote the best result and blue cells denote the secondary. "gap" denotes the gap between ReFer and the best baseline.

Dataset	Douban								MovieLens						Meituan			
Scenario	Book-Movie		Book-Music		Movie-Book		Movie-Music		Task1		Task2		Task3		Rec-Search		Search-Rec	
Method	AUC	NLL	AUC	NLL	AUC	NLL	AUC	NLL	AUC	NLL	AUC	NLL	AUC	NLL	AUC	NLL	AUC	NLL
UN-Local	0.6289	0.6011	0.6709	0.5865	0.7132	0.6431	0.7222	0.6192	0.7308	0.6149	0.7239	0.6241	0.7239	0.6074	0.5912	0.3628	0.6204	0.1627
AL-Local	0.6744	0.5948	0.6506	0.5835	0.7154	0.6260	0.6872	0.6427	0.7537	0.5780	0.7480	0.5813	0.7428	0.5782	0.6068	0.5084	0.6106	0.1676
Local	0.6784	0.5995	0.6818	0.5746	0.7350	0.6085	0.7256	0.6128	0.7584	0.5777	0.7545	0.5750	0.7493	0.5777	0.6182	0.1623	0.6305	0.1615
Fed	0.6532	0.5961	0.6534	0.5813	0.7109	0.6233	0.6843	0.6398	0.7547	0.5774	0.7401	0.5918	0.7303	0.6024	0.5376	0.1821	0.6001	0.1654
Fed-Fill	0.6795	0.5983	0.6804	0.5709	0.7337	0.6094	0.7263	0.6130	0.7641	0.5628	0.7545	0.5712	0.7505	0.5651	0.6413	0.1608	0.6439	0.1606
FTL	0.6179	0.6127	0.6296	0.6030	0.6948	0.6869	0.7034	0.6844	0.7081	0.6738	0.7172	0.6713	0.6792	0.6685	0.5913	0.1652	0.6018	0.1641
FPD	0.6823	0.5816	0.6828	0.5744	0.7405	0.6046	0.7319	0.5994	0.7615	0.5652	0.7565	0.5671	0.7498	0.5733	0.6184	0.1622	0.6321	0.1614
FedCVT	0.6818	0.5814	0.6738	0.5748	0.7374	0.5952	0.7268	0.6039	0.7590	0.5750	0.7544	0.5771	0.7509	0.5652	0.6187	0.1622	0.6316	0.1614
ReFer	0.6989	0.5673	0.6934	0.5629	0.7462	0.5916	0.7386	0.5934	0.7672	0.5581	0.7654	0.5637	0.7559	0.5610	0.6447	0.1606	0.6499	0.1602
gap*	**+0.0166**	**-0.0141**	**+0.0106**	**-0.0080**	**+0.0057**	**-0.0036**	**+0.0067**	**-0.0060**	**+0.0031**	**-0.0047**	**+0.0089**	**-0.0034**	**+0.0050**	**-0.0041**	**+0.0034**	**-0.0002**	**+0.0060**	**-0.0004**

* **It is worth noting that an AUC increase of 0.001 can be considered a significant improvement in CTR prediction [14, 25, 39, 51]**

Table 4: AUC results on aligned(AL) and unaligned(UL) users. ReFer achieves the best result in all scenarios in most cases.

Dataset	Douban								MovieLens						Meituan			
Scenario	Book-Movie		Book-Music		Movie-Book		Movie-Music		Task1		Task2		Task3		Rec-Search		Search-Rec	
Method	UL	AL	UL	AL	UL	AL	UL	AL	UL	AL	UL	AL	UL	AL	UL	AL	UL	AL
UN-Local	0.6681	0.6092	0.6747	0.6705	0.7180	0.7081	0.7257	0.7143	0.7173	0.7360	0.7034	0.7357	0.7101	0.7274	0.6061	0.5988	0.6412	0.5976
AL-Local	0.6241	0.6893	0.6225	0.6929	0.7128	0.7183	0.6894	0.6845	0.7185	0.7734	0.7142	0.7682	0.7079	0.7592	0.6061	0.6075	0.6330	0.6108
Local	0.6547	0.6841	0.6770	0.6887	0.7379	0.7322	0.7310	0.7132	0.7311	0.7729	0.7255	0.7716	0.7215	0.7601	0.6152	0.6106	0.6441	0.6130
Fed	0.6282	0.6924	0.6231	0.6961	0.7094	0.7179	0.6904	0.6868	0.7164	0.7855	0.7084	0.7756	0.7041	0.7656	0.5880	0.6303	0.6237	0.6296
Fed-Fill	0.6547	0.6855	0.6777	0.6846	0.7363	0.7332	0.7308	0.7152	0.7250	0.7886	0.7200	0.7793	0.7146	0.7697	0.6380	0.6332	0.6518	0.6308
FTL	0.6516	0.6451	0.6529	0.6007	0.7068	0.6998	0.7108	0.7038	0.6868	0.7575	0.6971	0.7639	0.6698	0.7206	0.5953	0.6028	0.6260	0.5987
FPD	0.6564	0.6890	0.6755	0.6938	0.7433	0.7374	0.7372	0.7208	0.7282	0.7799	0.7246	0.7750	0.7207	0.7613	0.6156	0.6104	0.6445	0.6153
FedCVT	0.6653	0.6862	0.6643	0.6965	0.7426	0.7379	0.7323	0.7160	0.7328	0.7745	0.7247	0.7737	0.7193	0.7660	0.6158	0.6109	0.6453	0.6141
ReFer	0.6705	0.7099	0.6780	0.7134	0.7554	0.7412	0.7410	0.7339	0.7299	0.7890	0.7369	0.7836	0.7226	0.7712	0.6429	0.6357	0.6605	0.6357
gap*	**+0.0024**	**+0.0175**	**+0.0003**	**+0.0169**	**+0.0121**	**+0.0033**	**+0.0038**	**+0.0131**	**-0.0029**	**+0.0004**	**+0.0114**	**+0.0043**	**+0.0011**	**+0.0015**	**+0.0049**	**+0.0025**	**+0.0087**	**+0.0049**

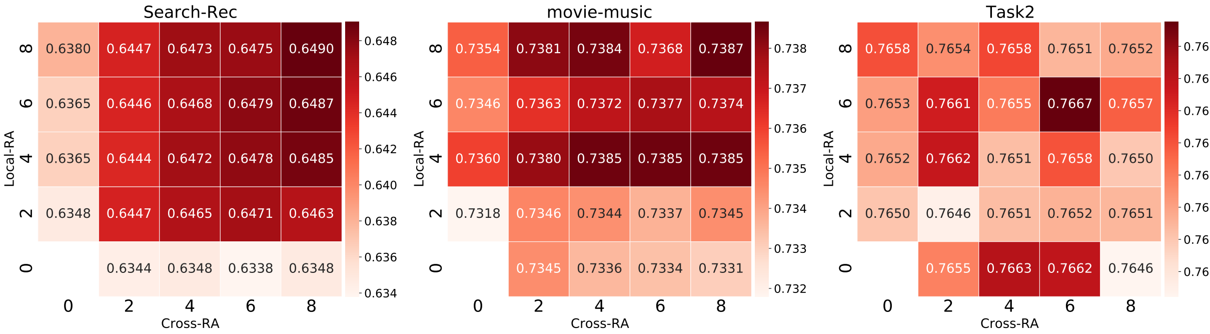

Figure 5: Ablation study on augmentation mechanisms with varying k. The heat map visualizes the value of AUC on the full user set. We show one scenario per dataset due to page limitation.

drawback. This indicates that using zero-filling could not entirely exploit the data advantage and may even incur a performance drop. Thus, developing a tailored method for more efficient full-set data utilization is necessary. **(3) User group performance bias generally exists**. There are 6 of 9 scenarios where AL-Local outperforms UN-Local on full-set user AUC while 3 of 9 for the rest, indicating the representative user group (who generalize better to all users) varies in different scenarios.

For *advanced baselines*, we observe that: **(1) Rigid feature alignment across heterogeneous domains is detrimental**. Contrary to Fed-Fill, FTL adds an extra loss term to directly minimize the discrepancy between the representations of the two parties. However, FTL consistently underperforms compared to Fed-Fill. This result suggests that direct alignment is impractical in recommendation systems. In our experiments, the data fields of the two parties neither overlap nor share the same item domain, demonstrating an intrinsic significant discrepancy. **(2) Distillation is effective but suffers from field missing**: FPD outperforms Local in all nine cases, indicating its effectiveness in retaining transferred federated knowledge. However, it underperforms compared to Fed-Fill in four out of nine cases, suggesting that a performance drop may occur when the shortage of field missing suppresses the benefit of field information distillation. **(3) Disentangled alignment and feature imputation are beneficial but not consistently stable**: FedCVT significantly outperforms FTL and occasionally achieves the best performance among all baselines (twice for Logloss and

Table 5: Ablation study of full user set AUC on the decreased volume of aligned users. ReFer consistently achieves better performance than baseline methods on all ratios. Green cells denote the best and bold font denote the secondary.

	Movie-Book				Movie-Music			
align ratio	Local	Fed-Fill	ReFer	gap	Local	Fed-Fill	ReFer	gap
0.1	0.7240	**0.7246**	0.7340	+0.0094	0.7272	**0.7275**	0.7353	+0.0078
0.25	**0.7274**	0.7271	0.7421	+0.0147	0.7287	**0.7287**	0.7348	+0.0061
0.5	0.7331	**0.7332**	0.7414	+0.0082	**0.7356**	0.7354	0.7423	+0.0067
0.75	0.7335	**0.7335**	0.7392	+0.0057	0.7274	**0.7276**	0.7356	+0.0080

once for AUC). Given that FedCVT employs a more refined feature alignment approach, similar to Domain Separation Networks [4], and actively imputes missing B-side data in the hidden space, these results are justified. The effectiveness of FedCVT's feature imputation module depends on the quality of feature disentanglement and alignment. However, ensuring this representation quality is quite challenging due to the heterogeneity and dynamicity of cross-domain user interests in federated recommendation environments, thus limiting FedCVT's effectiveness.

5.3 Analysis on Different User Groups(RQ2)

Performance on distinct user groups As illustrated in Table 4, we observed that ***ReFer achieves the best performance for both user groups*** (in all cases for aligned users' AUC and in 8 out of 9 cases for unaligned users' AUC). Although ReFer is not optimal for unaligned AUC in the case of Movielens-Task1, it remains the best for the full-set user AUC. Such phenomena can occur and may stem from the distribution bias between user groups (see Table 2 for cross-group user ratio and rating sparsity), with aligned users' advantage intensifying performance disparities of two user groups. In the case of Movielens-Task1, such a negative effect is also corroborated by the performance of the top two baselines for aligned users

Furthermore, we can also observe in Table 4 that, under certain data scenarios, Fed-Fill, FPD, and FedCVT also achieve a significant performance lift over basic baselines for specific user groups. However, due to the interest distribution bias and complex interplay between aligned and unaligned users, ***these baselines seldom achieve simultaneous benefit dominance for both user groups***. This in return justifies ReFer's advantage in achieving win-win performance benefits for user sets.

Effects on decreased aligned users Since ReFer relies on the aligned users to conduct cross-domain retrieval, we additionally investigate how aligned user's volume affects ReFer performance. As summarized in Table 5, ***ReFer works consistently well on varying sizes of aligned users***, maintaining its advantage of full-user-set AUC on all data ratios. This indicates that the cross-RA mechanism is robust to the size reduction of aligned users.

5.4 Components Ablation Study(RQ3)

Effects of Augmentation Mechanisms and Retrieval Size We present the impact of two augmentation mechanisms on the AUC metric for the full user set in Figure 5. We observe that: **(1) For a single mechanism, performance improves with increasing k until it reaches an optimal value.** Examining the heatmap either horizontally or vertically reveals a general trend: both augmentation mechanisms benefit from an increasing size of k, yet the improvement plateaus beyond a certain point. Intuitively, too few retrieval samples may lack sufficient information to aid the prediction process, while too many retrieval samples may introduce too many irrelevant noisy samples. **(2) Combining both mechanisms further enhances performance.** Despite the intricate interplay between the two mechanisms, integrating them with an appropriate retrieval size consistently yields the best performance, confirming the effectiveness of both.

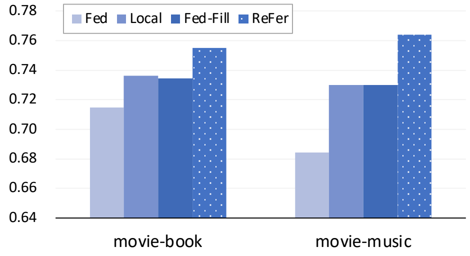

Figure 6: Ablation study on the distribution of training data for the user retriever. ReFer consistently exhibits an advantage over the baselines.

Effects of Different User Retrievers To showcase ReFer 's adaptability to diverse user embeddings, we trained an alternative user retriever on training dataset ratings, unlike the original which utilized left-out historical data. That is, the new retriever employs a different rating distribution to learn user similarity. The results depicted in Figure 6 indicate that ReFer still outperforms the baselines, by a substantial margin, highlighting its robustness in retriever selection. The essence of an effective retriever lies in accurately gauging cross-user similarity—a feature that is somewhat consistent across various rating data. Such adaptability underscores the practicality of ReFer in real-world applications, where a strong pre-trained retriever can be reused in multiple downstream tasks, reducing the burden of designing specific retrievers for each task.

6 CONCLUSION AND FUTURE WORK

In this paper, we introduce ReFer , ***the first retrieval-enhanced VFL algorithm*** designed to address the narrowed data scope problem of vertical federated recommendation. We propose a general "*retrieve-and-utilization*" framework to acquire enhanced representations for all parties, which results in improved performance. Our experimental results demonstrate that ReFer can achieve significant performance lifts on various datasets and tasks, revealing its potential in real applications and further research. In future work, we plan to explore new retrieval strategies and investigate the presence of popularity bias in the retrieval process.

ACKNOWLEDGMENTS

This work is supported by Meituan and partially supported by the NSFC Grant 62171248, the Shenzhen Science and Technology Program (JCYJ20220818101012025) and the PCNL KEY project (PCL2023AS6-1). Thanks to Mingyan Zhu for his valuable discussion, Jianghui Zhang and Chenghui Song for their early participation, Tao Dai and Bin Chen for their support in project management.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[2] Shuqing Bian, Wayne Xin Zhao, Jinpeng Wang, and Ji-Rong Wen. 2022. A Relevant and Diverse Retrieval-enhanced Data Augmentation Framework for Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2923–2932.

[3] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*. PMLR, 2206–2240.

[4] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. *Advances in neural information processing systems* 29 (2016).

[5] Iker Ceballos, Vivek Sharma, Eduardo Mugica, Abhishek Singh, Alberto Roman, Praneeth Vepakomma, and Ramesh Raskar. 2020. SplitNN-driven Vertical Partitioning. *CoRR* abs/2008.04137 (2020). arXiv:2008.04137 https://arxiv.org/abs/2008.04137

[6] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051* (2017).

[7] Tianyi Chen, Xiao Jin, Yuejiao Sun, and Wotao Yin. 2020. Vafl: a method of vertical asynchronous federated learning. *arXiv preprint arXiv:2007.06081* (2020).

[8] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. (2024). arXiv:2401.08281 [cs.LG]

[9] Fangcheng Fu, Huanran Xue, Yong Cheng, Yangyu Tao, and Bin Cui. 2022. BlindFL: Vertical Federated Machine Learning without Peeking into Your Data. In *Proceedings of the 2022 International Conference on Management of Data*. 1316–1330.

[10] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*. PMLR, 3929–3938.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[12] Yaochen Hu, Di Niu, Jianming Yang, and Shengping Zhou. 2019. FDML: A collaborative machine learning framework for distributed features. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2232–2240.

[13] Mingkai Huang, Hao Li, Bing Bai, Chang Wang, Kun Bai, and Fei Wang. 2020. A federated multi-view deep learning framework for privacy-preserving recommendations. *arXiv preprint arXiv:2008.10808* (2020).

[14] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 169–177.

[15] Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282* (2020).

[16] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[17] Yan Kang, Yang Liu, and Xinle Liang. 2022. FedCVT: Semi-supervised Vertical Federated Learning with Cross-view Training. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 4 (2022), 1–16.

[18] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[19] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172* (2019).

[20] Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566* (2021).

[21] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[22] Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, and Chong Wang. 2021. Label Leakage and Protection in Two-party Split Learning. In *International Conference on Learning Representations*.

[23] Wenjie Li, Qiaolin Xia, Hao Cheng, Kouyin Xue, and Shu-Tao Xia. 2022. Vertical semi-federated learning for efficient online advertising. *arXiv preprint arXiv:2209.15635* (2022).

[24] Wenjie Li, Qiaolin Xia, Junfeng Deng, Hao Cheng, Jiangming Liu, Kouying Xue, Yong Cheng, and Shu-Tao Xia. 2022. Semi-Supervised Cross-Silo Advertising with Partial Knowledge Transfer. *arXiv preprint arXiv:2205.15987* (2022).

[25] Xiangyang Li, Bo Chen, HuiFeng Guo, Jingjie Li, Chenxu Zhu, Xiang Long, Sujian Li, Yichao Wang, Wei Guo, Longxia Mao, et al. 2022. IntTower: the Next Generation of Two-Tower Model for Pre-Ranking System. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3292–3301.

[26] Yushen Li, Jinpeng Wang, Tao Dai, Jieming Zhu, Jun Yuan, Rui Zhang, and Shu-Tao Xia. 2024. RAT: Retrieval-Augmented Transformer for Click-through Rate Prediction. In *Companion Proceedings of the ACM Web Conference 2024*.

[27] Feng Liang, Weike Pan, and Zhong Ming. 2021. Fedrec++: Lossless federated recommendation with explicit feedback. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4224–4231.

[28] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Dongxiao Yu, Jun Ma, Maarten de Rijke, and Xiuzhen Cheng. 2020. Meta matrix factorization for federated rating predictions. In *SIGIR*. 981–990.

[29] Yang Liu, Yan Kang, Chaoping Xing, Tianjian Chen, and Qiang Yang. 2020. A secure federated transfer learning framework. *IEEE Intelligent Systems* 35, 4 (2020), 70–82.

[30] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. 2022. Vertical Federated Learning. *arXiv preprint arXiv:2211.12814* (2022).

[31] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. 2012. Recommender systems. *Physics reports* 519, 1 (2012), 1–49.

[32] Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. 2020. Fedfast: Going beyond average for faster training of federated recommender systems. In *SIGKDD*. 1234–1242.

[33] Daniel Peterson, Pallika Kanani, and Virendra J Marathe. 2019. Private federated learning with domain adaptation. *arXiv preprint arXiv:1912.06733* (2019).

[34] Jiarui Qin, Weinan Zhang, Rong Su, Zhirong Liu, Weiwen Liu, Ruiming Tang, Xiuqiang He, and Yong Yu. 2021. Retrieval & interaction machine for tabular data prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1379–1389.

[35] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083* (2023).

[36] Zhenghang Ren, Liu Yang, and Kai Chen. 2022. Improving Availability of Vertical Federated Learning: Relaxing Inference on Non-overlapping Data. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2022).

[37] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567* (2021).

[38] Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. *Advances in Neural Information Processing Systems* 34 (2021), 25968–25981.

[39] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1161–1170.

[40] Jiankai Sun, Xin Yang, Yuanshun Yao, Aonan Zhang, Weihao Gao, Junyuan Xie, and Chong Wang. 2021. Vertical Federated Learning without Revealing Intersection Membership. *arXiv preprint arXiv:2106.05508* (2021).

[41] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. 2018. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564* (2018).

[42] Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10, 3152676 (2017), 10–5555.

[43] Jinpeng Wang, Ziyun Zeng, Yunxiao Wang, Yuting Wang, Xingyu Lu, Tianxiang Li, Jun Yuan, Rui Zhang, Hai-Tao Zheng, and Shu-Tao Xia. 2023. Missrec: Pre-training and transferring multi-modal interest-aware sequence representation for recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 6548–6557.

[44] Jinpeng Wang, Jieming Zhu, and Xiuqiang He. 2021. Cross-batch negative sampling for training two-tower recommenders. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 1632–1636.

[45] Kang Wei, Jun Li, Chuan Ma, Ming Ding, Sha Wei, Fan Wu, Guihai Chen, and Thilina Ranbaduge. 2022. Vertical federated learning: Challenges, methodologies and experiments. *arXiv preprint arXiv:2202.04309* (2022).

[46] Penghui Wei, Hongjian Dou, Shaoguo Liu, Rongjun Tang, Li Liu, Liang Wang, and Bo Zheng. 2023. FedAds: A Benchmark for Privacy-Preserving CVR Estimation with Vertical Federated Learning. *arXiv preprint arXiv:2305.08328* (2023).

[47] Liu Yang, Ben Tan, Vincent W. Zheng, Kai Chen, and Qiang Yang. 2020. *Federated Recommendation Systems*. Springer International Publishing, Cham, 225–239. https://doi.org/10.1007/978-3-030-63076-8_16

[48] Hamed Zamani, Fernando Diaz, Mostafa Dehghani, Donald Metzler, and Michael Bendersky. 2022. Retrieval-enhanced machine learning. *arXiv preprint arXiv:2205.01230* (2022).

[49] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. 2020. {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning. In *2020 USENIX annual technical conference (USENIX ATC 20)*. 493–506.

[50] Chuang Zhao, Hongke Zhao, Ming He, Jian Zhang, and Jianping Fan. 2023. Cross-domain recommendation via user interest alignment. In *Proceedings of the ACM Web Conference 2023*. 887–896.

[51] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.