Federated Recommendation with Explicitly Encoding Item Bias

Zhihao Wang[1*], He Bai[2*], Wenke Huang[1*], Duantengchuan Li[1], Jian Wang[1†], Bing Li[13†]

[1]School of Computer Science, Wuhan University
[2]School of Journalism and Information Communication, Huazhong University of Science and Technology
[3]Hubei Luojia Laboratory, Wuhan, China
{zhihao_wang, jianwang, bingli}@whu.edu.cn

Abstract

With the development of federated learning techniques and the increased need for user privacy protection, the federated recommendation has become a new recommendation paradigm. However, most existing works focus on user-level federated recommendation, leaving platform-level federated recommendation largely unexplored. A significant challenge in platform-level federated recommendation scenarios is severe label skew. Users behave in various ways on different platforms, bringing up the rating and item bias problem. In this work, we propose *FREIB* (Federated Recommendation with Explicitly Encoding Item Bias). The core idea is explicitly encoding item bias during federated learning, addressing the problem of fuzzy item bias, and achieving consistent representation in label skew scenarios. We achieve this by utilizing global knowledge guidance to model common rating patterns and by aligning feature prototypes to enhance item encoding at the same rating level. Extensive experiments conducted on three public datasets demonstrate the superiority of our method over several state-of-the-art approaches.

Introduction

Federated recommendation, a new recommender system paradigm incorporating advanced federated learning techniques (McMahan et al. 2017a), has demonstrated significant potential in protecting user privacy and providing personalized recommendations. In scenarios where data sources for recommender systems are decentralized and come from different clients, federated recommendation models deploy algorithms on each client for training and co-tuning on the server side. This schema ensures user privacy while delivering personalized recommendations.

In addition to user-level federated recommendation, real-world scenarios also involve platform-level federated recommendation tasks, where users interact on different platforms, exhibiting label skew (Kairouz et al. 2019). As shown in Fig. 1, users behave differently on various platforms, leading to the item bias and rating bias learning problem. Items tend to get similar ratings across platforms because of their characteristic, which indicates item bias. Users score strictly on some platforms and are prone to low scores, while they score leniently on others and are prone to high scores, showing the rating bias. However, the co-existence of item bias and rating bias will confuse item embedding learning. For example, the classic movie Titanic should be widely liked and have high ratings on different platforms. However, this is susceptible to the rating bias of different platforms, where items learned on platforms with strict scoring habits have lower scores, affecting the item representation and causing inconsistent item bias. Consequently, the gradient of the model on each platform falls in a different direction, making it challenging for the server to learn accurate item representations and make effective rating predictions.

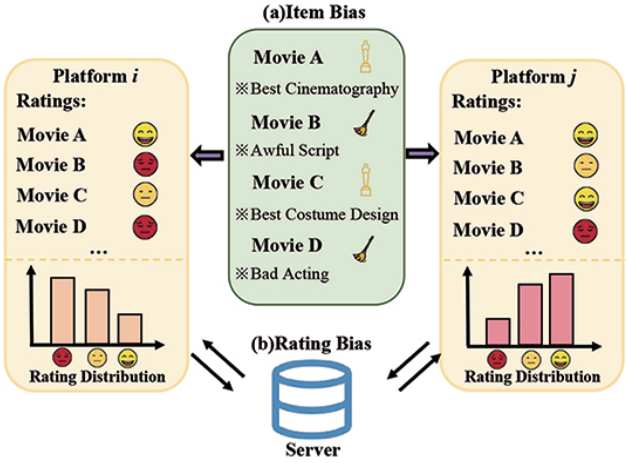

Figure 1: **Illustration of platform-level federated learning**. Users behave differently on various platforms, leading to the label skew problem. (a) Users give different ratings according to the characteristics of movies, which shows item bias. (b) The tendency of user scoring varies across platforms, resulting in rating bias between platforms.

Previous work on federated recommender systems has focused on privacy protection at the individual user level and deploying models to user clients for recommendation (Lin et al. 2021; Chai et al. 2021; Zhang et al. 2024c,b). However, these approaches face new challenges in platform-level federated recommendation scenarios requiring accurate gen-

*Equal contribution.
†Corresponding author

eralized recommendations. Firstly, the user and item encoders constructed by these methods model users and items at a coarse-grained level, ignoring the intrinsic bias of the items. Items attract users not only due to user-specific interests but also due to their intrinsic qualities, which are platform-independent. Secondly, these methods fail to effectively mine the potential associations among items with the same-level ratings in a rating task.

When modeling items, their objective existence is easily affected by different platforms, obfuscating the consistency that exists across platforms. To address this, we introduce an additional explicit item bias encoder to characterize item features at a finer granularity. By explicitly modeling item bias, we can get rid of inconsistent item embedding learned across platforms. In addition, considering that user behaviors interacting with and rating items should be similar across platforms, we utilize global knowledge guidance. Specifically, we fix the global server's model after each round of communication epoch and utilize it to guide the current client during training the local recommendation model. This training mode allows our model to learn a generalized pattern for users to rate items and platform-independent item bias.

Furthermore, items that receive the same ratings should have similar feature embedding of item bias due to their potential relationships. Therefore, we construct feature prototypes of item bias based on their ratings on the local platform, aggregate them and generate global prototypes on the server. While training the local model on different platform clients, we align the local prototypes of item bias to their corresponding global prototypes.

In a nutshell, we propose a novel method named Federated Recommendation with Explicitly Encoding Item Bias (*FREIB*). Our contributions are summarized as follows:

- We focus on the label skew problem in the platform-level federated recommendation and demonstrate that it hinders the generalization of federated learning methods, leading to performance degradation.
- We present a simple yet effective model *FREIB* to handle item bias and rating bias in the platform-level federated recommendation. Our model explicitly models the item bias, utilizes global knowledge guidance, and aligns feature prototypes.
- We conduct extensive experiments on three public datasets, and the results show the superiority of our model over state-of-the-art methods.

Related Work

In this section, we review related work of federated learning and federated recommendation.

Federated Learning

Federated learning has become a popular field being researched with the increase in data heterogeneity scenarios and privacy preservation needs (Konečný et al. 2016; Kairouz et al. 2019; Li et al. 2020a; Long et al. 2020; Huang, Ye, and Du 2022; Hao et al. 2022; Liang et al. 2023; Hu et al. 2024c; Wang et al. 2024a,b; Hu et al. 2024b; Liang et al. 2024). FedAvg (McMahan et al. 2017b) pioneers the training of global models from decentralized data by aggregating the parameters of local models. However, it performs poorly on non-i.i.d (identically and independently distributed) data, which attracts a lot of work to explore further (Luo et al. 2021; Li et al. 2022; Ma et al. 2022; Wu et al. 2023; Dai et al. 2023a; Tan et al. 2023; Hong et al. 2023; Hu et al. 2023a, 2024a). Based on FedAvg, existing models mainly utilize global penalty terms to solve the data heterogeneity problem. SCAFFOLD (Karimireddy et al. 2020) addresses 'client-drift' using control variates, reducing communication rounds and demonstrating robustness to data heterogeneity and client sampling. FedProx (Li et al. 2020b) considers the data heterogeneity and system heterogeneity and proposes a proximal term to guarantee the aggregation of the partial information of those incomplete computations in FedAvg. FedStar (Tan et al. 2023) exploits and exchanges the common latent structure information for inter-graph federated learning tasks. Besides, pFedME (T. Dinh, Tran, and Nguyen 2020), FedDyn (Acar et al. 2021) also employ distinct mechanisms for computing global parameter stiffness, thereby exerting control over the disparities that may arise among the distributed models. From another perspective, MOON (Li, He, and Song 2021), FedUFO (Zhang et al. 2021), FedProto (Tan et al. 2022), FedProc (Mu et al. 2023), FedNH (Dai et al. 2023b) enhance feature-level consensus between local and global models by prioritizing them. This emphasis on consistency of features at the micro level promotes robustness and consistency within the federation framework. For the label skew problem, FedConcat (Diao, Li, and He 2024) addresses it by concatenating local models, leveraging clustering for collaborative training within client groups based on label distributions. In this paper, we focus on the label skew problem of platform-level scenes in federated recommendation.

Federated Recommendation

With the increase of distributed data scenarios and privacy protection requirements, the federated recommendation becomes a new recommendation paradigm (Sun et al. 2024; Zhang et al. 2024a). FCF (Ammad-ud-din et al. 2019) first introduces a collaborative filtering method in the federated recommendation setting and employs FedAvg to train the global model. FedRec (Lin et al. 2021) proposes user averaging and hybrid filling strategies to protect the information of rating records. FedRec++ (Liang, Pan, and Ming 2021) further introduces an innovative lossless federated recommendation method that allocates certain denoising clients to eliminate noise. FedMF (Chai et al. 2021) utilizes the matrix factorization and further protects user privacy with homomorphic encryption techniques during the updating process. P-NSMF (Hu et al. 2022) introduces group-wise concealing and aggregates in a secure way to conduct non-sampling matrix factorization. FedNCF (Perifanis and Efraimidis 2022) applies NCF (He et al. 2017) in the federated recommendation, using a neural network to learn user and item embedding. FedPerGNN (Wu et al. 2022) encrypts the information of user neighbors to the third-party server to construct a graph neural network on each client. PerFedRec

Notation	Description
M	number of platforms
θ_m	private model of the m^{th} participant
θ^E	global model
D	set of rating records
$\mathbb{P}_m(r)$	label distribution of platform m
E	number of communication epochs
T	number of local rounds
d	vector dimension
$\mathcal{P}^m$	local feature prototypes of item bias in the m^{th} participant
$\mathcal{G}$	global feature prototypes of item bias
$r_{u,i}$	rating of item i by user u
$\mathcal{L}^m_{CE}$	loss of the basic NCF model
$\mathcal{L}^m_{bias}$	loss of the predictions with item bias
$\mathcal{L}^m_{distill}$	loss of global knowledge guidance
$\mathcal{L}^m_{proto}$	loss of feature prototype alignment
$\mathcal{L}^m$	loss of the m^{th} participant

Table 1: Notations and descriptions used in the paper.

(Luo, Xiao, and Song 2022) jointly learns the representation through a collaborative graph and performs users clustering to generate personalized recommendation. P-GCN (Hu et al. 2023b) utilizes item-based user representation and privacy-preserving graph convolution approach to handle federated item recommendation. P2FCDR (Chen et al. 2023) utilizes an optimizable orthogonal mapping matrix to transform the knowledge across domains and provides privacy protection by applying the local differential privacy technique. FPPDM (Liu et al. 2023) exploits user preferences in the local modeling and combines user characteristics across domains in the global server. F2PGNN (Agrawal et al. 2024) integrates personalized graph neural networks (GNNs) with differential privacy techniques to mitigate inherent bias across demographic groups. PFedRec (Zhang et al. 2023) only shares the item encoder in the communication epoch and learns the score function locally. Our work focuses on the scenario of platform-level data distribution, which has not been well explored.

Method

In this section, we propose a novel framework named Federated Recommendation with Explicitly Encoding Item Bias (*FREIB*). We will define the platform-level federated recommendation and describe the modules in the following subsections.

Problem Formulation

For the federated recommendation in a platform-level setting, we follow the typical federated learning framework. Suppose there are M platforms (indexed by m). Each platform has a local model θ_m and a set of rating records $D_m = \{(u, i, r_{u,i}) | u \in \mathbb{R}^{N_u}, i \in \mathbb{R}^{N_i}, r_{u,i} \in \mathbb{R}^{N_r}\}$, where N_u and N_i denote the numbers of users and items, respectively, and N_r denotes the rating levels. The label skew exists in this scenario, meaning that label distribution $\mathbb{P}_m(r)$ on the clients is distinct. To mimic the label skew, we follow the common experiments setting and use Dirichlet sampling (Balakrishnan, Kotz, and Johnson 2019). The primary goal is to optimize the models θ_m of each platform and aggregate them into the server-side model θ^E using certain strategies for better generalization and recommendation in the test set $D_0 = \{(u, i, r_{u,i})\}$. We describe the notations used in the paper in Tab. 1.

As NCF (He et al. 2017) has demonstrated its superiority in recommendation with a simple framework, we adopt it as our basic backbone. The calculation of NCF can be simplified as:

$$\hat{r}^o_{u,i} = NCF(u, i) = f^o(GMF(u, i) \oplus MLP(u, i)), \quad (1)$$

where $\oplus$ denotes the concatenate operation, GMF and MLP stand for the generalized matrix factorization model and the multi-layer perceptron, and $\hat{r}^o_{u,i}$ is the prediction score for user u on item i. Further, the loss of model can be formulated as:

$$\mathcal{L}^m_{CE} = \sum_{(u,i)\in D_m} r_{u,i} - \hat{r}^o_{u,i}. \quad (2)$$

Explicit Item Bias Encoder

Roughly modeling an item as a whole tends to ignore the inherent bias within the item itself. The overall embedding of the item can be distorted by the label distribution of different platforms, making it difficult to learn a generalizable representation through gradient descent. Therefore, we introduce the extra Explicit Item Bias Encoder (EIBE) to portray the intrinsic item bias, which remains consistent across different platforms.

First, for each item i, we explicitly construct the item bias embedding, formulated as:

$$\mathbf{E}_{bias} = Embedding(i_1, i_2, \cdots, i_{N_i}), \quad (3)$$

where $\mathbf{E}_{bias} \in \mathbb{R}^{d_{bias}}$ is the item bias embedding matrix for items, d_{bias} is the dimension of the item bias embedding vector.

Before the score function takes user features and item features as input and generates predictions, we introduce an additional score function for item bias embedding,

$$\hat{r}^i_{bias} = S_{bias}(\mathbf{E}^i_{bias}), \quad (4)$$

where $\hat{r}_{bias}$ is the item bias score for item i, and S_{bias} is the item bias score function. After acquiring the item bias score function, the prediction for rating of user u and item i with item bias can be formulated as:

$$\hat{r}^{bias}_{u,i} = \hat{r}^o_{u,i} + \hat{r}^i_{bias}, \quad (5)$$

where $\hat{r}^o_{u,i}$ is the original prediction score.

For the loss function of the predictions with item bias, we adopt the MSE loss, which for the m-th participant can be formulated as:

$$\mathcal{L}^m_{bias} = \sum_{(u,i)\in D_m} r_{u,i} - \hat{r}^{bias}_{u,i}. \quad (6)$$

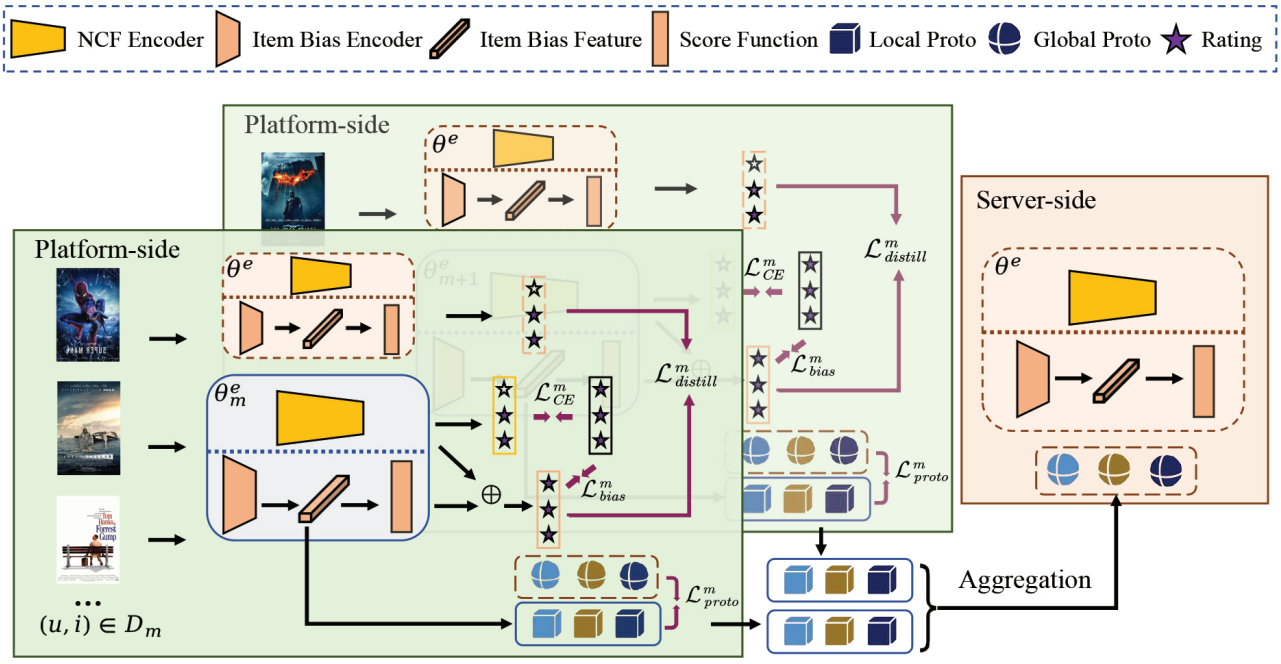

Figure 2: **Architecture illustration** of Federated Recommendation with Explicitly Encoding Item Bias (*FREIB*). To learn the consistent and platform-independent item bias, we introduce the extra Explicit Item Bias Encoder (EIBE) to generate predictions with bias. In each communication epoch, we fix the global model and utilize the Global Knowledge Guidance (GKG) to learn the common behavior mode of users. Besides, we construct local feature prototypes of item bias during local rounds. Participants upload local prototypes to the server, and the server aggregates them to generate global prototypes, which are used for Feature Prototype Alignment (FPA) in the next communication epoch. Best viewed in color. Zoom in for details.

Global Knowledge Guidance

Although the label skew causes label distributions to differ across platforms, the Global Knowledge Guidance (GKG) can still direct the training process. The common behavior mode of users rating items and the platform-independent item bias can be well adjusted through the aggregation of global knowledge. In particular, we utilize the server-side model parameters after the first communication epoch. We fix the server-side model during local training and generate predictions, and the process can be formulated as:

$$Net_{fixed} = \theta^E, \tag{7}$$

$$(\hat{r}^{bias}_{u,i})_{fixed} = Net_{fixed}(u,i), \tag{8}$$

where $(\hat{r}^{bias}_{u,i})_{fixed}$ is the prediction of the server-side model. By applying the MSE loss between the participant's predictions and the server's predictions, the global knowledge can guide the local training process of participant m:

$$\mathcal{L}^m_{distill} = \sum_{(u,i)\in D_m} \hat{r}^{bias}_{u,i} - (\hat{r}^{bias}_{u,i})_{fixed}. \tag{9}$$

Feature Prototype Alignment

Naturally, items within the same rating level should exhibit some potential similarities. Inspired by prototype learning, we construct feature prototypes of item bias according to their labels and conduct the Feature Prototype Alignment (FPA). In the communication epoch, each participant learns the local feature prototypes of item bias,

$$\mathcal{P}^m_k = \frac{1}{|N_k|} \sum_{i\in D_m, r_{u,i}=k} e^i_{bias}, \tag{10}$$

$$\mathcal{P}^m = \{\mathcal{P}^m_1, \mathcal{P}^m_2, \cdots, \mathcal{P}^m_K\}, \tag{11}$$

where N_k denotes the number of items rated as k and K is the total rating levels. The global server then aggregates all the local prototypes during the updating process:

$$\mathcal{G}_k = \frac{1}{M} \sum_{m=1}^{M} \mathcal{P}^m_k, \tag{12}$$

$$\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_K\}. \tag{13}$$

Given the global prototypes of item bias, the participant can align the features of the item bias during local training:

$$\mathcal{L}^m_{proto} = \sum_{k=1}^{K} \sum_{i\in D_m, r_{u,i}=k} e^i_{bias} - \mathcal{G}_k. \tag{14}$$

Considering the proposed modules above, the final optimization loss function for participant m can be formulated as:

$$\mathcal{L}^m = \mathcal{L}^m_{CE} + \mathcal{L}^m_{bias} + \mathcal{L}^m_{distill} + \tau\mathcal{L}^m_{proto}, \tag{15}$$

where τ is the weight hyper-parameter. The optimization process of *FREIB* is demonstrated in Algorithm 1.

Algorithm 1: Model training in *FREIB*

Input: Communication epochs E, local rounds T, number of participants M, the m^{th} participant private data $D_m(u, i, r_{u,i})$, private model θ_m
Output: The final global model θ^E

for $e = 1, 2, ..., E$ **do**
  Participant Side;
  for $m = 1, 2, ..., N$ **in parallel do**
    $\theta^e_m, \mathcal{P}_m \leftarrow$ **LocalUpdating**$(\theta^e, \mathcal{G})$
  Server Side;
  $\theta^{e+1} \leftarrow \frac{1}{M}\sum_{m=1}^{M} \theta^e_m$
  /* Global prototypes */
  $\mathcal{G} = \frac{1}{M}\sum_{m=1}^{M} \mathcal{P}^m$

LocalUpdating$(\theta^e, \mathcal{G})$:
$\theta^e_m \leftarrow \theta^e$; // Distribute global parameter
$Net_{fixed} \leftarrow \theta^e$; // Fix global parameter
for $t = 1, 2, ..., T$ **do**
  for $(u, i) \in D_m$ **do**
    e^i_{bias} in Eq. (3)
    $\hat{r}^o_{u,i} \leftarrow NCF$
    $\hat{r}^{bias}_{u,i}$ in Eq. (5)
    $(\hat{r}^{bias}_{u,i})_{fixed}$ in Eq. (8)
    $\mathcal{L}^m_{CE} \leftarrow (\hat{r}^o_{u,i}, r_{u,i})$ in Eq. (2)
    $\mathcal{L}^m_{bias} \leftarrow (\hat{r}^{bias}_{u,i}, r_{u,i})$ in Eq. (6)
    $\mathcal{L}^m_{distill} \leftarrow (\hat{r}^{bias}_{u,i}, (\hat{r}^{bias}_{u,i})_{fixed})$ in Eq. (9)
    $\mathcal{L}^m_{proto} \leftarrow (e^i_{bias}, \mathcal{G})$ in Eq. (14)
    $\mathcal{L}^m \leftarrow (\mathcal{L}^m_{CE}, \mathcal{L}^m_{bias}, \mathcal{L}^m_{distill}, \mathcal{L}^m_{proto})$in Eq. (15)
    $\theta^e_m \leftarrow \theta^e_m - \eta\nabla\mathcal{L}^m$
$\mathcal{P}_m = \{\}$; // Initialize local prototypes
/* Local prototypes */
for $k = 1, 2, ..., K$ **do**
  $\mathcal{P}^m_k = \frac{1}{|N_k|}\sum_{i \in D_m, r_{u,i}=k} e^i_{bias}$
return $\theta^e_m, \mathcal{P}^m$

Experiment

We set up various experiments on commonly used datasets to evaluate *FREIB*, search the optimal hyper-parameter, conduct the ablation study to verify the effectiveness of proposed modules, and explore the robustness of our method with local differential privacy in this section.

Experimental Setup

Datasets. We evaluate our proposed method on three public datasets: MovieLens-100K, MovieLens-1M(Harper and Konstan 2015), and Amazon-Beauty (Ni, Li, and McAuley 2019). MovieLens-100K contains 100,000 ratings of 1,682 movies from 943 users while MovieLens-1M contains 1 million ratings of 3,952 movies by 6,040 users. The Amazon dataset contains user reviews, ratings, and other metadata of products from Amazon.com. We select Amazon-Beauty according to the category, which has about 2,000,000 records and 260,000 items. The statistics of datasets are listed in Tab. 2. We remove users with less than five interactions to ensure the federated learning setting of label skew. The datasets are randomly split with the ratio of 8:2 into the training set and the test set, following the common setting in machine learning.

Dataset	MovieLens-100K	MovieLens-1M	Beauty
#Users	943	6,040	40,226
#Items	1,682	3,952	54,542
#Interactions	100,000	100,000,209	353,962

Table 2: Statistics of datasets

Comparison Methods. We compare *FREIB* with methods in platform-level federated settings, all utilizing only rating records for information. For federated learning methods, we compare with FedProx (Li et al. 2020b) (MLSys'20), FedMF (Chai et al. 2021) (IEEE Intell Syst'21), FedProto (Tan et al. 2022) (AAAI'22), FedNCF (Perifanis and Efraimidis 2022) (KBS'22), FedPerGNN (Wu et al. 2022) (Nat. Commun.'22), as well as the recently proposed PFedRec (Zhang et al. 2023) (IJCAI'23).

Evaluation Metrics. We adopt two widely used metrics for rating prediction: MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error). MAE evaluates the absolute deviation between rating predictions and ground-truth labels, while RMSE measures the variance of the deviation. Both of them measure the model's performance on the rating prediction task.

Implementation Details. For a fair comparison, we set the number of participants to 5, conduct the communication epochs $E = 50$, and perform 10 local rounds ($T = 10$) for the federated setting. We adopt the linear layer as the score function. Besides, the initial embedding size is fixed at 32 for all methods, except the embedding size of item bias is set as 10. We use the SGD (Robbins and Monro 1951) optimizer with a learning rate $lr = 0.001$ except PFedRec (Zhang et al. 2023), which employs a larger learning rate with the item encoder based on the scale of datasets. The weight decay is set to $1e{-}5$ and the momentum to 0.9. The training batch size is 64. For the weight hyper-parameter, τ is set as 10 in *FREIB*. For standardized comparisons, we adopt NCF as the backbone in FedProx and FedProto, while the hyper-parameters for regularization and prototype learning weights in FedProx and FedProto are also set to 10. We conduct prototype learning according to labels and the averaging in FedProto. We implement the federated learning methods on different platforms by applying the Dirichlet sampling with common parameter $\beta = \{1.0, 0.5\}$. We fix the seed to ensure reproduction and conduct experiments on the NVIDIA 3090.

Results

Performance Comparison. We compare the performance of *FREIB* on three datasets, and the results are reported in Tab. 3. From the results, we have the following observations. *FREIB* outperforms all the baseline methods in

Methods	β=1						β=0.5					
	MovieLens-100K		MovieLens-1M		Beauty		MovieLens-100K		MovieLens-1M		Beauty	
	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓
FedProx	0.9425	1.1245	0.9491	1.1193	1.1865	1.3785	0.9811	1.1742	0.9482	1.1184	1.4337	1.5347
FedMF	0.9434	1.1253	0.9450	1.1173	1.1766	1.3756	0.9716	1.1507	0.9421	1.1159	1.4305	1.5319
FedProto	0.9426	1.1249	0.9479	1.1189	1.1849	1.3782	0.9795	1.1671	0.9474	1.1186	1.4499	1.5496
FedNCF	0.9429	1.1245	0.9498	1.1200	1.1848	1.3781	0.9809	1.1735	0.9492	1.1190	1.3977	1.5036
FedPerGNN	0.9371	1.1243	0.9477	1.1182	1.1837	1.3738	0.9677	1.1441	0.9157	1.1113	1.3007	1.4333
PFedRec	1.1708	1.4694	0.8650	1.0949	1.0969	1.3427	0.9191	1.1626	0.8431	1.0680	1.2962	1.4272
FREIB	**0.7369**	**0.9395**	**0.7275**	**0.9241**	**0.9579**	**1.3180**	**0.7926**	**0.9912**	**0.7454**	**0.9406**	**1.1635**	**1.4233**

Table 3: **Comparison with the state-of-the-art methods** on MovieLens-100K, MovieLens-1M and Beauty datasets. The best results in federated setting are bolded and the suboptimal results are underlined.

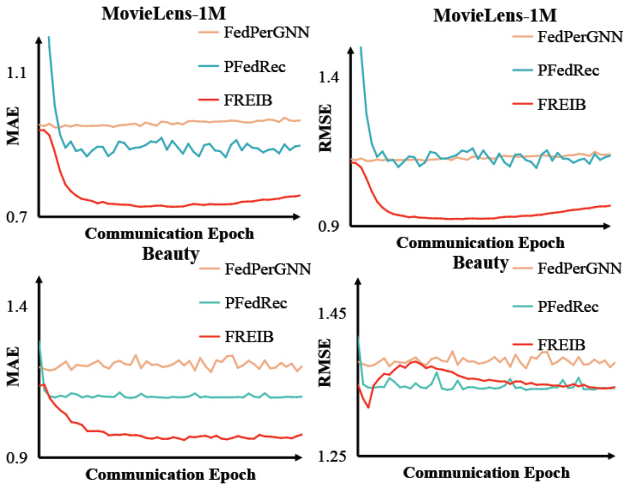

Figure 3: **MAE and RMSE** of MovieLens-1M and Beauty in the training

.

the platform-level federated setting, indicating the effectiveness of our framework. Due to the label skew existing in platform-level federated scenarios, methods simply utilizing the matrix factorization, NCF framework, and graph networks perform badly in the experiments, showing the challenge of learning consistent item embedding. PFedRec obtains sub-optimal results in most datasets because it isolates the learning of item embedding and sore function, highlighting the need for differentiated learning item embedding. It suffers severe performance degradation in MovieLens-100K, and this may be due to the fact that it over-amplifies the learning rate of the item, resulting in the failure to learn the accurate and consistent item embedding in the federated recommendation scenario of label skew. Besides, applying the prototype learning paradigm directly to the NCF framework, FedProto does not work well either due to ignoring the importance of item bias embedding. This also proves that simply applying prototype learning does not lead to significant performance improvements.

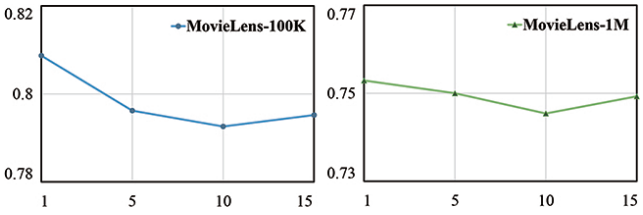

Figure 4: Results of different τ in MovieLens-100K and MovieLens-1M ($\beta = 0.5$).

Furthermore, we count the values of the MAE and RMSE during the training process for methods on the MovieLens-1M and Beauty dataset ($\beta = 1$), as shown in Fig. 3. The curves of the four graphs show a similar trend. The results indicate that methods like FedPerGNN that do not model items separately struggle to converge in the platform-level federated recommendation scenario, while PFedRec and *FREIB* converge effectively. Meanwhile, the trend of a similar curve of PFedRec further proves the effectiveness of the learning of separated item embedding. Due to the synergistic effect of our introduced modules, *FREIB* converges fast and steadily.

Hyper-Parameter Setting. As for the hyper-parameter τ, we conduct experiments in MovieLens-100K and MovieLens-1M with Dirichlet sampling parameter $\beta = 0.5$. We search the best hyper-parameter in the range of $[1, 5, 10, 15]$. From Fig. 4, we can observe that MAE decreases as τ increases from 1 to 10, and it will increase when τ exceeds 10 in both datasets. These two similar curve trends are because when τ is relatively small, $\mathcal{L}^m_{proto}$ occupies a relatively small proportion, and *FREIB* cannot align the prototype well. When τ is too large, it will focus too much on the prototype alignment and ignore the learning of other components, affecting the model's generalization ability and leading to performance degradation. Therefore, we choose 10 as the best hyper-parameter in *FREIB*.

Ablation Study. To better understand the performance of the modules of *FREIB*, we conduct a series of ablation ex-

Bias	Distill	Proto	MovieLens-100K MAE	MovieLens-100K RMSE	MovieLens-1M MAE	MovieLens-1M RMSE
✘	✔	✘	0.9449	1.1253	0.9272	1.1129
✔	✘	✔	0.7767	0.9904	0.7788	0.9999
✔	✔	✘	0.7597	0.9530	0.7419	0.9491
✔	✔	✔	0.7369	0.9395	0.7275	0.9241

Table 4: **Ablation Study** on MovieLens-100K and MovieLens-1M datasets($\beta = 1$). ✘ denotes removing the corresponding module while ✔ means keeping it.

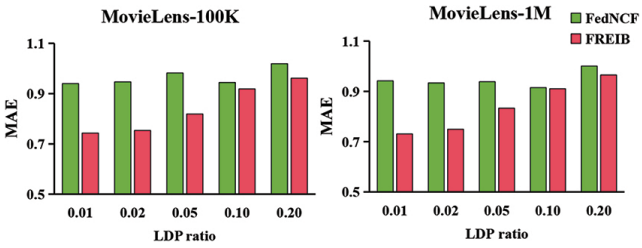

Figure 5: **Influence of the LDP ratio** in the training of MovieLens-100K and MovieLens-1M. The higher LDP ratio indicates stronger noises and the lower MAE indicates better performance.

periments and the results are demonstrated in Tab. 4. We remove the item bias encoder and prototype alignment module to completely isolate the influence of item bias. This leads to the most significant performance degradation, highlighting the importance of introducing explicit item bias. The explicit item bias encoder avoids the effect of label distribution of different platforms and ensures consistent bias representations for items. This also shows that the direct application of distillation to the ordinary NCF model still cannot solve the problem of learning item bias. Besides, without the global knowledge guidance, *FREIB* fails to learn the common behavior patterns in rating items and the consistent item bias, resulting in worse performance. Additionally, removing the feature prototype alignment reveals that rating bias prevents both clients and the server from learning similar item features.

Protection with Local Differential Privacy. To improve the protection of user privacy, we apply Local Differential Privacy(LDP) (Choi et al. 2018) in our framework. Concretely, we combine the parameters of clients with the Laplacian noise before uploading to the server. The Laplacian noise can be formulated as Laplace(0,λ), and λ is the LDP ratio, which represents the noise strength. We set $\lambda = [0.01, 0.02, 0.05, 0.10, 0.20]$ to test our framework in MovieLens-100K and MovieLens-1M. As shown in Fig. 5, *FREIB* outperforms the baseline even with added noise. Although performance degrades as the LDP ratio increases, it remains acceptable, striking a balance between privacy protection and recommendation accuracy.

Discussion

Relationship with relative federated prototype learning. The core idea of prototype learning is to store a set of representative samples (prototypes) and use these prototypes to perform tasks such as classification, regression, or clustering. The main advantage is its ability to effectively handle complex data distributions, especially when there is overlap or imbalance between data categories. Different from previous works like FedProto(Tan et al. 2022), which learn prototypes according to categories, we specifically consider rating bias and item bias in the platform-level federated learning scenario and aggregate the features of item bias as prototypes regarding their labels. We apply prototype learning to enhance feature learning of item bias, learn potential similarities between items with the same label, and achieve remarkable results in experiments.

Conception Differences. Previous federated recommendation works have mostly focused on providing federated recommendation models for individual-level users, with less attention paid to platform-level federated recommendation issues. These works utilize various federated training methods and graph network technologies (Liang, Pan, and Ming 2021; Wu et al. 2022; Zhang et al. 2023) to bring improvements to personalized federated recommendation. However, our focus is on the platform-level federated recommendation scenario of label skew, where the phenomena of item bias and rating bias are more prominent. However, previous works have not been able to provide specialized solutions to these two issues. We explicitly model item bias for these two phenomena and further solve the label skew problem using global knowledge guidance and feature prototype alignment.

Limitations. The components we introduced bring extra time cost, but they result in stable performance improvements on multiple datasets, demonstrating the promising potential of our approach. Besides, we explicitly encode the item bias, which can effectively alleviate label skew in platform-level federated scenarios. In addition, there is a lack of semantic interpretation of the embedding of item bias and its prototype, which can be further explored in future work.

Conclusions and Future Work

In this paper, we propose a federated recommendation method with explicit item bias, namely *FREIB*, focusing on the scenario of platform-level federated learning. *FREIB* is capable of handling the item bias and rating bias existing in the platform-level federated recommendation. We conduct various experiments and the results show that our method outperforms state-of-the-art methods.

While our introduced modules bring significant performance gains, they also introduce additional time and space overheads that can be further optimized in future work. In addition, the semantic interpretability of item bias and prototype in federated recommendations needs to be further explored.

Acknowledgments

We thank the support of National Natural Science Foundation of China under Grant Nos. 62032016 and 623B2080.

References

Acar, D. A. E.; Zhao, Y.; Navarro, R. M.; Mattina, M.; Whatmough, P. N.; and Saligrama, V. 2021. Federated Learning Based on Dynamic Regularization. In *9th International Conference on Learning Representations*.

Agrawal, N.; Sirohi, A. K.; Kumar, S.; and , J. 2024. No Prejudice! Fair Federated Graph Neural Networks for Personalized Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Ammad-ud-din, M.; Ivannikova, E.; Khan, S. A.; Oyomno, W.; Fu, Q.; Tan, K. E.; and Flanagan, A. 2019. Federated Collaborative Filtering for Privacy-Preserving Personalized Recommendation System. *CoRR*, abs/1901.09888.

Balakrishnan, N.; Kotz, S.; and Johnson, N. L. 2019. Continuous Multivariate Distributions, Volume 1: Models and Applications.

Chai, D.; Wang, L.; Chen, K.; and Yang, Q. 2021. Secure Federated Matrix Factorization. *IEEE Intelligent Systems*.

Chen, G.; Zhang, X.; Su, Y.; Lai, Y.; Xiang, J.; Zhang, J.; and Zheng, Y. 2023. Win-Win: A Privacy-Preserving Federated Framework for Dual-Target Cross-Domain Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Choi, W.-S.; Tomei, M.; Vicarte, J. R. S.; Hanumolu, P. K.; and Kumar, R. 2018. Guaranteeing Local Differential Privacy on Ultra-Low-Power Systems. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture*.

Dai, Y.; Chen, Z.; Li, J.; Heinecke, S.; Sun, L.; and Xu, R. 2023a. Tackling Data Heterogeneity in Federated Learning with Class Prototypes. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Dai, Y.; Chen, Z.; Li, J.; Heinecke, S.; Sun, L.; and Xu, R. 2023b. Tackling Data Heterogeneity in Federated Learning with Class Prototypes. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Diao, Y.; Li, Q.; and He, B. 2024. Exploiting Label Skews in Federated Learning with Model Concatenation. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hao, M.; Li, H.; Chen, H.; Xing, P.; Xu, G.; and Zhang, T. 2022. Iron: Private inference on transformers. *Advances in neural information processing systems*.

Harper, F. M.; and Konstan, J. A. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*.

Hong, J.; Wang, H.; Wang, Z.; and Zhou, J. 2023. Federated Robustness Propagation: Sharing Adversarial Robustness in Heterogeneous Federated Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hu, M.; Cao, Y.; Li, A.; Li, Z.; Liu, C.; Li, T.; Chen, M.; and Liu, Y. 2024a. FedMut: Generalized Federated Learning via Stochastic Mutation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hu, M.; Xia, Z.; Yan, D.; Yue, Z.; Xia, J.; Huang, Y.; Liu, Y.; and Chen, M. 2023a. GitFL: Uncertainty-Aware Real-Time Asynchronous Federated Learning Using Version Control. In *In Proceedings of IEEE Real-Time Systems Symposium*.

Hu, M.; Yue, Z.; Xie, X.; Chen, C.; Huang, Y.; Wei, X.; Lian, X.; Liu, Y.; and Chen, M. 2024b. Is aggregation the only choice? federated learning via layer-wise model recombination. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Hu, M.; Zhou, P.; Yue, Z.; Ling, Z.; Huang, Y.; Li, A.; Liu, Y.; Lian, X.; and Chen, M. 2024c. FedCross: Towards Accurate Federated Learning via Multi-Model Cross-Aggregation. In *IEEE International Conference on Data Engineering*.

Hu, P.; Lin, Z.; Pan, W.; Yang, Q.; Peng, X.; and Ming, Z. 2023b. Privacy-preserving graph convolution network for federated item recommendation. *Artificial Intelligence*.

Hu, P.; Yang, E.; Pan, W.; Peng, X.; and Ming, Z. 2022. Federated one-class collaborative filtering via privacy-aware non-sampling matrix factorization. *Knowledge-Based Systems*.

Huang, W.; Ye, M.; and Du, B. 2022. Learn from Others and Be Yourself in Heterogeneous Federated Learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2019. Advances and Open Problems in Federated Learning. *arXiv preprint arXiv:1912.04977*.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2020. SCAFFOLD: stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*.

Konečný, J.; McMahan, H. B.; Ramage, D.; and Richtárik, P. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *CoRR*, abs/1610.02527.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2022. Federated Learning on Non-IID Data Silos: An Experimental Study. In *2022 IEEE 38th International Conference on Data Engineering*.

Li, Q.; He, B.; and Song, D. 2021. Model-Contrastive Federated Learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020a. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020b. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems*.

Liang, F.; Pan, W.; and Ming, Z. 2021. FedRec++: Lossless Federated Recommendation with Explicit Feedback. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Liang, K.; Liu, Y.; Zhou, S.; Tu, W.; Wen, Y.; Yang, X.; Dong, X.; and Liu, X. 2023. Knowledge graph contrastive learning based on relation-symmetrical structure. *IEEE Transactions on Knowledge and Data Engineering*.

Liang, K.; Meng, L.; Liu, M.; Liu, Y.; Tu, W.; Wang, S.; Zhou, S.; Liu, X.; Sun, F.; and He, K. 2024. A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lin, G.; Liang, F.; Pan, W.; and Ming, Z. 2021. FedRec: Federated Recommendation With Explicit Feedback. *IEEE Intelligent Systems*.

Liu, W.; Chen, C.; Liao, X.; Hu, M.; Yin, J.; Tan, Y.; and Zheng, L. 2023. Federated probabilistic preference distribution modelling with compactness co-clustering for privacy-preserving multi-domain recommendation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*.

Long, G.; Tan, Y.; Jiang, J.; and Zhang, C. 2020. *Federated Learning for Open Banking*, 240–254. Springer International Publishing.

Luo, M.; Chen, F.; Hu, D.; Zhang, Y.; Liang, J.; and Feng, J. 2021. No Fear of Heterogeneity: Classifier Calibration for Federated Learning with Non-IID Data. In *Advances in Neural Information Processing Systems*.

Luo, S.; Xiao, Y.; and Song, L. 2022. Personalized Federated Recommendation via Joint Representation Learning, User Clustering, and Model Adaptation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*.

Ma, X.; Zhu, J.; Lin, Z.; Chen, S.; and Qin, Y. 2022. A state-of-the-art survey on solving non-IID data in Federated Learning. *Future Generation Computer Systems*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and Arcas, B. A. y. 2017a. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and Arcas, B. A. y. 2017b. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*.

Mu, X.; Shen, Y.; Cheng, K.; Geng, X.; Fu, J.; Zhang, T.; and Zhang, Z. 2023. FedProc: Prototypical contrastive federated learning on non-IID data. *Future Generation Computer Systems*.

Ni, J.; Li, J.; and McAuley, J. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Conference on Empirical Methods in Natural Language Processing*.

Perifanis, V.; and Efraimidis, P. S. 2022. Federated Neural Collaborative Filtering. *Knowledge-Based Systems*.

Robbins, H.; and Monro, S. 1951. A stochastic approximation method. *AoMS*.

Sun, Z.; Xu, Y.; Liu, Y.; He, W.; Kong, L.; Wu, F.; Jiang, Y.; and Cui, L. 2024. A Survey on Federated Recommendation Systems. *IEEE Transactions on Neural Networks and Learning Systems*.

T. Dinh, C.; Tran, N.; and Nguyen, J. 2020. Personalized Federated Learning with Moreau Envelopes. In *Advances in Neural Information Processing Systems*.

Tan, Y.; Liu, Y.; Long, G.; Jiang, J.; Lu, Q.; and Zhang, C. 2023. Federated Learning on Non-IID Graphs via Structural Knowledge Sharing. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022. FedProto: Federated Prototype Learning across Heterogeneous Clients. In *AAAI Conference on Artificial Intelligence*.

Wang, H.; Jia, Y.; Zhang, M.; Hu, Q.; Ren, H.; Sun, P.; Wen, Y.; and Zhang, T. 2024a. FedDSE: Distribution-aware Submodel Extraction for Federated Learning over Resource-constrained Devices. In *Proceedings of the ACM Web Conference 2024*.

Wang, H.; Zheng, P.; Han, X.; Xu, W.; Li, R.; and Zhang, T. 2024b. FedNLR: Federated Learning with Neuron-wise Learning Rates. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Wu, C.; Wu, F.; Lyu, L.; Qi, T.; Huang, Y.; and Xie, X. 2022. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications*.

Wu, X.; Huang, H.; Ding, Y.; Wang, H.; Wang, Y.; and Xu, Q. 2023. FedNP: Towards Non-IID Federated Learning via Federated Neural Propagation. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhang, C.; Long, G.; Guo, H.; Fang, X.; Song, Y.; Liu, Z.; Zhou, G.; Zhang, Z.; Liu, Y.; and Yang, B. 2024a. Federated Adaptation for Foundation Model-based Recommendations. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*.

Zhang, C.; Long, G.; Zhou, T.; Yan, P.; Zhang, Z.; Zhang, C.; and Yang, B. 2023. Dual Personalization on Federated Recommendation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*.

Zhang, C.; Long, G.; Zhou, T.; Zhang, Z.; Yan, P.; and Yang, B. 2024b. GPFedRec: Graph-Guided Personalization for Federated Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Zhang, C.; Long, G.; Zhou, T.; Zhang, Z.; Yan, P.; and Yang, B. 2024c. When Federated Recommendation Meets Cold-Start Problem: Separating Item Attributes and User Interactions. In *Proceedings of the ACM on Web Conference 2024*.

Zhang, L.; Luo, Y.; Bai, Y.; Du, B.; and Duan, L.-Y. 2021. Federated Learning for Non-IID Data via Unified Feature Learning and Optimization Objective Alignment. In *2021 IEEE/CVF International Conference on Computer Vision*.