SUPPLEMENTARY MATERIAL TO THE PAPER ENTITLED "PREVENTING THE POPULAR ITEM EMBEDDING BASED ATTACK IN FEDERATED RECOMMENDATIONS"

A. Details of the Experiment Settings

1) Datasets: We use three datasets for evaluation, namely MovieLens-100K (ML-100K) [11], MovieLens-1M (ML-1M) [11], and Amazon Digital Music (AZ) [12]. The statistics are shown in Table IX, where AZ features a lower interaction rate compared to the other two while all datasets from real-world scenarios exhibit high sparsity. Such sparsity typically makes it challenging for FRS to accurately learn users' preferences and achieve effective recommendations. Following a previous study [16], for each user, we adopt the leave-one-out method to create the training and test sets.

TABLE IX: Dataset statistics ('Rate' equals #(Interactions)/ #(Users) and 'Sparsity' equals $1 - \#(\text{Interactions})/(\#(\text{Users}) \times \#(\text{Items}))$).

Dataset	#(Users)	#(Items)	#(Interactions)	Rate	Sparsity
ML-100K	943	1,682	100,000	106	93.70%
ML-1M	6,040	3,706	1,000,209	166	95.53%
AZ	16,566	11,797	169,781	10	99.91%

2) System Settings: To confirm the model-agnostic nature of PIECK, we adopt Matrix Factorization [29] for MF-FRS and Neural Collaborative Filtering (NCF) [15] for DL-FRS as the underlying base models. Despite the existence of various dedicated recommender models with intricate components, the above two models have gained widespread adoption in practice and have been extended to the federated setting [30], [31], [41]. Their appeal lies in their ability to ensure high generalization and low communication overhead, making them highly representative choices for our evaluation. In our setting, each user in the dataset is regarded as a 'client' in the federation. The batch size of randomly selected users per round for MF-FRS is 256 on ML-100K and ML-1M datasets and 1024 on AZ. For DL-FRS, the batch size remains 256 for all datasets. We train MF-FRS and DL-FRS with learning rates $\eta = 1.0$ and $\eta = 0.005$, where corresponding model parameters are set the same as previous studies [30], [31]. Other parameter settings, related to the popular item mining algorithm and the defense loss can be found in our instruction [1].

3) Attack Baselines: We compare PIECK with the following state-of-the-art model poisoning attacks:

- FEDRECA [31] accesses a *portion* of benign users' historical interactions to approximate their embeddings.
- PIPA [41] involves explicit promotion and popularity enhancement for target items.
- A-RA [30] randomly initializes users' embeddings at malicious users, specifically designed for DL-FRS.
- A-HUM [30] extends A-RA by enhancing the attack based on mining hard users for increased effectiveness.

For a fair comparison without utilizing prior knowledge, we mask the historical interactions for FEDRECA and popularity levels of items for PIPA in the implementation. Moreover, we set null parameters for A-RA when applying it to MF-FRS.

4) Defense Methods: To evaluate PIECK and compare our proposed defense method, we apply different defense methods to the aggregation function $\text{Agg}(\cdot)$ on the server and tune them *optimal*. They process the uploaded gradients as follows.

- NORMBOUND [32]: A thresholding approach is employed to bound the L_2 Norm of all gradients uploaded by users.
- MEDIAN [39]: The median of received gradients for each dimension is computed.
- TRIMMEDMEAN [39]: The $\tilde{p}$ largest and smallest values for each dimension are removed, and the rest are averaged.
- KRUM [4]: The most similar gradient from received gradients in the squared Euclidean norm space is selected.
- MULTIKRUM [4]: The $(2\tilde{p})$ least similar gradients produced by KRUM are removed iteratively with the rest averaged.
- BULYAN [24]: Gradients are selected by MULTIKRUM and then averaged using TRIMMEDMEAN.

5) Evaluation Metrics: In attacking and defending an FRS, both the **effectiveness of attack** and the **recommendation performance** are crucial considerations. An effective attack should successfully promote target items without significantly degrading the performance of the system to avoid detection.

To assess the attack effectiveness in promoting target items, we utilize the *Exposure Ratio at rank K* (ER@K) as defined in Eq. (3). Attacks like PIPA and A-HUM specifically focus on the least popular items, which allows for more significant promotion opportunities since the server receives fewer benign gradients. However, for fairness, we follow FEDRECA [31] and randomly select target items to T from the set of uninteracted items. To measure the recommendation performance, we employ the *Hit Ratio at rank K* (HR@K) following the NCF approach [15]. A higher HR@K indicates more accurate recommendations of the FRS.

B. Discussion of High Sampling Ratio q

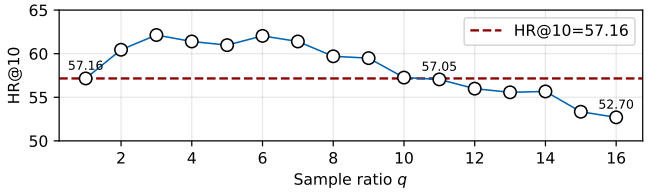

Fig. 7: Effect of sample ratio q of MF-FRS on ML-100K.

Here, we discuss the effect of sampling rate q in FRS on recommendation performance without attack and defense. It can be seen from Figure 7 that as q increases, HR @10 first increases and then decreases. When $q \geq 11$, the RS performance gradually drops to unreliable, which is worse than $q = 1$ (HR@10=57.16), which makes it unattractive to attackers.

C. Discussion of Target Item Number

We evaluate the performance of attack and defense under $\tilde{p} = 5\%$ and a larger # of target items $|T| = 2, 3, 4, 5$. In both cases, malicious users ($P = 5$ for PIECKIPE and $P =$

TABLE X: Effect of T on proposed attacks and defense.

Attacks	Defenses	$\|T\|=2$		$\|T\|=3$		$\|T\|=4$		$\|T\|=5$	
		ER@10	HR@10	ER@10	HR@10	ER@10	HR@10	ER@10	HR@10
NoAttack	NoDefense	0.00	57.26	0.23	57.16	0.23	57.16	0.23	57.16
PieckIpe	NoDefense	75.61	**57.58**	57.09	57.48	39.55	57.16	32.70	56.84
PieckIpe	**ours**	0.80	57.48	0.46	**58.01**	0.05	57.90	0.02	58.22
PieckUea	NoDefense	**92.72**	57.05	**87.33**	57.05	**85.24**	56.73	**81.84**	56.31
PieckUea	**ours**	0.00	56.2	0.00	57.69	0.00	58.11	0.13	58.22

TABLE XI: Effect of inconsistent learning rates on our attacks.

η_i	NoAttack		PieckIpe		PieckUea	
	ER@10	HR@10	ER@10	HR@10	ER@10	HR@10
1e-0	0.23	57.16	87.47	57.69	93.39	57.69
1e-2	0.00	49.31	98.06	48.99	95.56	48.36
1e-2∼1e-0	0.00	21.74	55.13	22.16	55.69	21.95

TABLE XII: Effect of loss function on proposed attacks and defense.

Attacks	Defenses	BCE		BPR	
		ER@10	HR@10	ER@10	HR@10
NoAttack	NoDefense	0.23	57.16	0.00	57.26
PieckIpe	NoDefense	87.47	**57.69**	83.14	**57.48**
PieckIpe	**ours**	1.25	56.31	0.00	54.29
PieckUea	NoDefense	**93.39**	57.69	**90.21**	56.95
PieckUea	**ours**	0.00	55.89	3.64	53.76

50 for PieckUea) manipulated an equal number of poisoned gradients for target items. Table X shows the effectiveness of our attacks and defenses on a larger set of target items. Notably, attackers may employ more sophisticated strategies to achieve superior results, such as directing 1/3 of malicious users to poison one of the three target items, which will better fit the larger $|T|$.

D. Discussion of Consistency of Learning Rate

we explore the impact of inconsistent learning rates on MF-FRS and ML-100K. First, we maintain uniform learning rates within the client ($\eta_i = 1e-2$) but not with the server ($\eta = 1e-0$). Second, we introduce non-uniform learning rates within the client, with dynamic changes ranging from $1e-2$ to $1e-0$. Table XI indicates that two inconsistent learning rate settings will significantly reduce the recommendation performance, but PieckIpe is still effective. It's noteworthy that our attack remains effective across all settings.

E. Discussion of Universality of Loss Function

In this scenario, we consider the client utilizing the BPR loss to train FRS. As shown in Table XII, our attacks and defenses remain effective under the BPR loss, demonstrating the universality of our attack and defense across different loss functions.