A Value-Preserving and Efficiency-Aware Split Learning Framework for Recommender System: Rebuttal

To Reviewer RSHF

Thanks very much for your precious time and careful review. Here we provide answers for some of the questions mentioned hoping to clarify your confusion.

Q1 Can the proposed framework be generalized to other base models other than FM?

R1 Here, we first provide the conclusion: other base models might be adapted to the decentralized split RS setting, but require dedicated designs to assure security and efficiency during the training and inference process, which is a future direction of our work.

In this paper, we focus on adapting DeepFM into this decentralized split setting. Previous SOTA, FedAds [1], pioneered the usage of features from multiple parts in a vertically split scenario, whose model structure used is naive with only fully connected layers. We take a step forward and adapt DeepFM to such a multiple-party-involved, vertical federated setting. Compared to naive DNN, DeepFM is a classic but effective structure and is still widely used in real-world business. However, adapting DeepFM to such a vFL setting is not straightforward because the second-order feature (e.g. $x_1 \times x_2$) in the Factorized Machine (FM) requires direct interaction in the early stage, which leads to a serious threat in the above-mentioned decentralized paradigm. Therefore, we design a safe FM-based splitting with secure interaction modeling in section 3.2 to avoid direct input exposure from a follower(s) to the leader.

Under the split DeepFM overarching premise, the three parts 3.2 (FM-based Secure Modeling), 3.3 (Reducing Data Leakage), and 3.4 (Efficiency Optimization) do not have a sequential relationship; rather, they serve the overall framework aimed at leveraging features from more participants for a better recommendation system. However, the VIB (section 3.3) and efficiency optimization (section 3.4) techniques can be potentially generalized to other base models as long as the models are adapted to such a decentralized setting.

Q2-1 Eq. (3): missing square on v and x?

R2-1 No, the h_{p_1} and h_{p_2} in Eq.3 are the key to our secure interaction modeling of FM. Given the original target of the FM part in Eq.2, $\hat{y}_{spltFM}$ can be divided into the sum of $y_1 = \sum_{p=1}^{q} w_{p0} + \sum_{p=1}^{q}\sum_{i=1}^{n_p} w_{pi}x_{pi} - \frac{1}{2}[\sum_{p=1}^{q}\sum_{j=1}^{k}\sum_{i=1}^{n_p} v_{pi,j}^2 x_{pi}^2]$ and $y_2 = \frac{1}{2}\sum_{j=1}^{k}(\sum_{p=1}^{q}\sum_{i=1}^{n_p} v_{pi,j}x_{pi})^2$. Here y_1 can be separately calculated by local input features by participants. However, y_2 requires the early interaction of different parties. In a centralized setting, the input features are put together, here, each follower only sends its local sum $h_{p_2} = \sum_{i=1}^{n_p} v_{pi,j}x_{pi}$ to the leader. The leader aggregates all h_{p_2} and calculates the final y_2 at the leader's end. The leader cannot infer the concrete value of input features from the follower from h_{p_2}, which guarantees the secure split DeepFM modeling.

Q2-2 This is actually the value of the j-th value in the k-dimensional hp2, i.e., hp2[j]?

R2-2 You are correct. As explained in A2-1, the h_{p_2} is a k-dimensional vector and this is the value of the j-th value in the k-dimensional hp2. We will revise the expression of Eq. 2 to eliminate confusion.

Q3 Line 335: K-th: better use lower case. capital letters are used for constants.

R3 We will use another lowercase letter to represent the index to reduce ambiguity.

Q4-1 Eq. (5) Can you justify the rationale of using model performance to assess the level of information leakage? In my opinion, you assume that there's the correlation between performance and level of leakage, which may not be true.

R4-1 With the secure interaction above, which focuses solely on the FM part, such a decentralized RS paradigm may still have potential risks of privacy leakage (the followers' data value), which is not considered in [1]. However, the measurement of such data leakage is still unclear. So in section 3.3, we first define the data value leakage γ between the primary task A to the unknown attack task B, which measures how much unnecessary information is leaked from the follower to the leader. In other words, we allow the adequate data value transferred from the follower to the leader since additional information is needed for improving the recommendation performance on the primary task A. If there is no information transfer, such a multi-party paradigm will not bring any benefits. However, what we do want to mitigate, is the extra data value leakage from the follower's intermediate feature which is unnecessary for recommendation task A and can be potentially used for another task B (or C, D, E).

In our split DeepFM, γ measures the information leakage on h_{p3} (the output of the Deep part). Since h_{p_3} is also a vital part contributing to the final result. The theory of our proposed VIB to prevent data value leakage is to pass only the information necessary to the primary task ('Read' here) but reduce the extra information that can be potentially used by the leader for another secret task ('Like', 'Share', 'Favorite' here in Table 2). If the leader uses the followers' features and gets a relatively high performance gain on a secret attack task (measured by recommendation performance) than only using its own local features, this means unnecessary data value leakage.

Q4-2 What's the range of gamma? There's no guarantee gamma is positive?

R4-2 The theoretical range of gamma is from 0% to 100%. 0% means when the leader uses the intermediate feature received from a follower(s) for primary task A, the leader cannot get performance gain on other secret tasks (B, C, D). A 100% means when the leader uses the intermediate feature received from follower(s) for primary task A, the performance gain is more or less the same as the original data from followers are directly sent to the leader and trained in a centralized environment.

Q5-1 Table 2. It seems like there's a consistent absolute drop in model performance (0.1 0.2 drop). It is hard to convince me that KL divergence alleviates information leakage, rather than saying that KL divergence leads to consistent information drop. Besides, the drop on the main task Read is more significant than other tasks.

R5-1 The theory of our proposed VIB to prevent data value leakage is to pass only the information necessary to the primary task ('Read' here) but reduce the extra information that can be potentially used by the leader for another secret task.

Here, our aim is to reduce the performance gain on the attack tasks ('Like', 'Share', 'Favorite') while minimizing the impact on the performance of the main task 'Read'.

Results in Table 2 (VIB(ours)) and Figure 5 (blue) show that our VIB protection proposed can effectively prevent data value leakage (significant performance gain decrease in attack task 'Like' and 'Favorite') and the performance drop in main task 'Read' is minor (no drop when $\beta = 0$and -0.13% when $\beta = 0.001$).

Only the attack task 'Share' shares a similar performance drop rate with the primary task 'Read' (which means cutting the information does not work well on task 'Share'). This is because task 'Share' only uses a minimal amount of input features, or in other words, there is merely no extra abundant information in h_{p3}if we want to keep the recommendation performance of task 'Read' unaffected.

Q5-2 Overall impression on gamma: the value of gamma is not so straightforward in telling the level of leakage. It varies in a large range.

R5-2 According to Eq.5, we define data value leakage as the measurement of how much unnecessary information is transferred from the follower(s) to the leader, which is a major concern preventing enthusiasm of parties to participate concerning the uncontrollable information abuse at the leader's end. From our perspective, a higher γmeans the leader can once and for all make use of the data value and do not require any participants for future cooperation. While a close-to-zero γensures no extra information leakage other than the necessary part for the agreed task.

Q5-3 68.5% drop in Share scenario: Why the drop in leakage is not significant in the Share scenario?

R5-3 Following A5-1, only the attack task 'Share' shares a similar performance drop rate with the primary task 'Read' (which means cutting the information does not work well on task 'Share'). This is because task 'Share' only uses a minimal amount of input features, or in other words, there is merely no extra abundant information in h_{p3}if we want to keep the recommendation performance of task 'Read' unaffected.

Q6 Line 397: introducing, Line 402: are

R6 We would like to thank the reviewer once again for their careful reading. We will thoroughly review the entire text again and correct all the grammatical errors and typos in the edited version.

[1] FedAds: A Benchmark for Privacy-Preserving CVR Estimation with Vertical Federated Learning, https://arxiv.org/pdf/2305.08328