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

To Reviewer CkT9

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 I am confused about the "value protection". Imagine that you are in a company operating the machines (including the leader and follower). Now you run the code in these machines, why would there be a concern that the leader might use the follower's data for unauthorized tasks? Both machines are under the same control. I don't know where there is such a need to protect the follower representation from being abused by the leader.

R1 The initial and first purpose of our paper is to propose a cohesive split learning framework to enable the usage of more user or item-related features (not available or have serious safety concerns in a centralized setting) from different participants for a potentially better overall recommendation performance (both training & inference). Under this consumption, the participants (the leader and the followers) are all separated and do not belong to the same entity. This is also why we pay such close attention to the security of information dissemination (section 3.2,3.3) among different parties and the efficiency of training in real-life network environments (section 3.4).

Q2 In eq(5), does pstealB use the leader's data for task B?

R2 Yes, p^B_{steal} use the leader's data for secret task B. We 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 followers' 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 $h_{p3}$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.

Q3 In section 4.1, what method is used to divide the training set, validation set, and test set? Global timeline? Random?

R3 We randomly split the entire dataset into a training set, validation set, and test set with the proportions of 90%, 5%, and 5%, respectively. We use a fixed global random seed for the split, and this division remains unchanged for each experiment. The results of each experiment are averaged over five trials to ensure the validity and reliability of the conclusions.

Q4 Code should be released for reproducibility.

R4 We are more than willing to release our code; however, certain parts of our code, specifically, those related to improving training efficiency and experiments with different bandwidths in real-life applications utilize open-source libraries that may expose our identity. Therefore, we guarantee to release our code if published.