- *1. **Privacy preserving**. R4/R3: the features maps might leak privacy; R1: privacy property has not been described.*

We will discuss the privacy concerns in our revision. **1)** our work does not focus on privacy-preserving techniques, but we believe the hidden vector reconstruction attack can be defended by privacy-preserving techniques such as differential privacy (DP) and multi-party computation (MPC). **2)** it is not necessarily straightforward to conclude that a hidden feature map is less safe than the model or gradient. As pointed out by R4, the model or gradient exchange may also leak privacy. At least from the current literature, no paper has investigated the comparison from a theoretical or experimental perspective. **3)** please note that the hidden feature exchange happens at the training phase. This makes the attack harder because what the attacker access is the evolving and untrained feature map rather than a fully trained feature map that represents the raw data. Analyzing the degree of privacy leakage under our framework will be our future works.

- *“2. **Communication Cost**. R4: may not always save the communication cost because it depends on the sizes of features and logits; R3: comparison has not been shown explicitly; R1: a dependence of bandwidth on the dataset size”*

Please note that the sizes of features and logits are fixed by the CNN architecture design. Compared to the entire model weight or gradient, the hidden vector is definitely much smaller (e.g., the hidden vector size of ResNet-110 is around 64KB while the entire gradient/model size is 4.6MB for 32x32 images). The hidden vector for each data point can be transmitted independently, thus GKT has less bandwidth requirement than gradient or model exchange, although the communication cost in total depends on the number of data points. Moreover, our experimental result shown in **Figure 5** even demonstrates that our method has smaller communication costs for the entire training than split learning (a method that also exchanges hidden vectors during training) because of less communication round for convergence. We will explain communication more obviously in our revision.

- *“3. **Scalability with subsampling strategy**. R3: does the proposed method support cross-device setting? Can we use subsampling?How to maintain the optimizer state among clients? R2: subsampling can save communication cost."*

Our method definitely can support the cross-device setting. First, we believe the user selection strategy is still an open problem. It is better to tailor the strategy for different models and optimization methods. The random subsampling method mentioned by R3 and R2 may not fit for our large DNN setting. The random subsampling may cause many users’ data to be touched only once. This is not a big issue for shallow NN because shallow NN requires much fewer data to converge than large DNN. However, it is a problem to large DNN because it typically requires all samples to be trained many epochs. Therefore, under GKT framework, it is more reasonable to use a pre-defined client selection strategy. All clients are divided into many groups. We then train group by group to make sure each group is trained multiple rounds (epochs). The optimizer state of each group can be maintained by uploading it to the server of GKT. Once the group ID is changed, the server then synchronizes the optimizer state to the clients in the new group. This training process is essentially the same as our GKT algorithms: from the perspective of optimization, both “viewing 10 users’ dataset as an epoch (as done in our experiments)“ and “viewing each user’ dataset as an epoch and training user by user“ can converge. We will demonstrate this in our revision. Besides sampling, client-edge-cloud hierarchical FL is also a potential solution. We can use the edge server in the hierarchical topology to improve load balance. Also, from the perspective of alternating optimization, GKT does not have scalability issues since GKT does not do synchronous aggregation on the server-side: the server can immediately start training once it receives updates from any client.

- *“4: **Ablation Study**. R3: the diverge result is unconvincing; mention the takeaway earlier in the method section; Please provide IID and non-IID experiments for ablation study”*

Good suggestions. The word “diverge” is somewhat misleading. Actually, we find it is hard to tune the parameters if we do not use KD, sometimes it diverges, and sometimes it gets a low accuracy. We will clarify this in our revision. Although we get this takeaway in ResNet, “both" KD loss may be more effective to other models. So it is better to leave it as a hyper-parameter to tune in practice rather simply concluding which direction is useful. Both IID and non-IID experiments will be provided in our revision.

- *“5: **Benchmark datasets**. R3: it is better to use the benchmark dataset by [1] adaptive federated optimization.”*

We admit that using benchmark dataset is a good practice for FL research, but please note that our scenario is totally different from [1]: 1) our research focuses on large CNN training, but [1] is shallow neural network research (see the appendix of this paper, only 2 Conv layers are used). Its CV dataset only includes FMNIST and CIFAR-100, which are inadequate to evaluate a large CNN model. In fact, we use datasets that are more difficult than [1], including CIFAR-10, CIFAR-100, and CINIC-10. 2) our method focuses on a new training framework rather than proposing a better FL optimizer. Thus it is unnecessary to align the CIFAR-100 partition the same as [1].

- *“6. additional comments from R3 such as 1) Algorithm 1 notation issues; 2) TensorFlow Federated”*

We sincerely thank R3 for so many useful suggestions. 1) we will modify Algorithm 1 as your suggestion; 2) Compared to TTF, we develop a more flexible send/recv framework (Fig. 6). We will try TTF in our future work.

- *“7. R1: the paper’s significance could be strengthened by discussing on the scale to high resolution images”*

Thanks for your suggestion. We will evaluate the efficiency for high resolution images in our future works.

- *“8. R2: comparing with FedMD; extend GKT to language models like LSTM.”*

We already discussed different knowledge distillation methods including FedMD in our related works. As for the language model, we extend it as a future work because careful consideration is needed to tailor for the characteristics of LSTM and Transformer (the back-propagation through time in LSTM and attention mechanisms in Transformer).