Neural Tangent Kernel Empowered Federated Learning

Kai Yue [1] Richeng Jin [1] Ryan Pilgrim [2] Chau-Wai Wong [1] Dror Baron [1] Huaiyu Dai [1]

Abstract

Federated learning (FL) is a privacy-preserving paradigm where multiple participants jointly solve a machine learning problem without sharing raw data. Unlike traditional distributed learning, a unique characteristic of FL is statistical heterogeneity, namely, data distributions across participants are different from each other. Meanwhile, recent advances in the interpretation of neural networks have seen a wide use of neural tangent kernels (NTKs) for convergence analyses. In this paper, we propose a novel FL paradigm empowered by the NTK framework. The paradigm addresses the challenge of statistical heterogeneity by transmitting update data that are more expressive than those of the conventional FL paradigms. Specifically, sample-wise Jacobian matrices, rather than model weights/gradients, are uploaded by participants. The server then constructs an empirical kernel matrix to update a global model without explicitly performing gradient descent. We further develop a variant with improved communication efficiency and enhanced privacy. Numerical results show that the proposed paradigm can achieve the same accuracy while reducing the number of communication rounds by an order of magnitude compared to federated averaging.

1. Introduction

Federated learning (FL) has emerged as a popular paradigm involving a large number of clients collaboratively solving a machine learning problem (Kairouz et al., 2021). In a typical FL framework, a server broadcasts a global model to selected clients and collects model updates without accessing the raw data. One popular algorithm is known as *federated averaging* (FedAvg) (McMahan et al., 2017), in which clients perform stochastic gradient descent (SGD) to update the local models and then upload the weight vectors to the server. A new global model is constructed on the server by averaging the received weight vectors.

Li et al. (2020) characterized some unique challenges for FL. First, client data are generated locally and remain decentralized, which implies that they may not be independent and identically distributed (IID). Prior work has shown that statistical heterogeneity can negatively affect the convergence of FedAvg (Zhao et al., 2018). This phenomenon may be explained by noting that local updating under data heterogeneity will cause cost-function inconsistency (Wang et al., 2020). A more challenging issue is the system heterogeneity, including the diversity of client hardware, battery power, and network connectivity. Local updating schemes often exacerbate the straggler issue caused by heterogeneous system characteristics (Li et al., 2020).

Recent studies have proposed various strategies to alleviate the statistical heterogeneity. One possible solution is to share a globally available dataset with participants to reduce the distance between client-data distributions and the population distribution (Zhao et al., 2018). In practice, though, such a dataset may be unavailable or too small to meaningfully compensate for the heterogeneity. Some researchers replaced the coordinate-wise weight averaging strategy in FedAvg with nonlinear aggregation schemes (Wang et al., 2020; Chen & Chao, 2021). The nonlinear aggregation relies on a separate optimization routine, which can be elusive, especially when the federated model does not perform well. Another direction is to modify the local objectives or local update schemes to cancel the effects of client drift (Li et al., 2020; Karimireddy et al., 2020). However, some studies reported that these methods are not consistently effective when evaluated in various settings (Reddi et al., 2021; Haddadpour et al., 2021; Chen & Chao, 2021).

In this work, we present a neural tangent kernel empowered federated learning (NTK-FL) paradigm. NTK-FL outperforms state-of-the-art methods by achieving the target accuracy with fewer communication rounds. We summarize our contributions as follows.

- We propose a novel FL paradigm without requiring clients to perform local gradient descent. To the best of our knowledge, this is the first work using the NTK method

[1]NC State University [2]Independent Scholar. Correspondence to: Chau-Wai Wong <chauwai.wong@ncsu.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

to replace gradient descent to diversify the design of FL algorithms.

- Our scheme inherently solves the non-IID data problem of FL. Compared to FedAvg, it is robust to different degrees of data heterogeneity and has a consistently fast convergence speed. We verify the effectiveness of the paradigm theoretically and experimentally.
- We add **c**ommunication-efficient and **p**rivacy-preserving features to the paradigm and develop CP-NTK-FL by combining strategies such as random projection and data subsampling. We show that some strategies can also be applied to traditional FL methods. Although such methods cause performance degradation when applied to FedAvg, they only slightly worsen the model accuracy when applied to the proposed CP-NTK-FL.

2. Related Work

Neural Tangent Kernel. Jacot et al. (2018) showed that training an infinitely wide neural network with gradient descent in the parameter space is equivalent to kernel regression in the function space. Lee et al. (2019) used a first-order Taylor expansion to approximate the neural network output and derived the training dynamics in a closed form. For the analyses, Chen et al. (2020) established the generalization bounds for a two-layer over-parameterized neural network with the NTK framework. The NTK computation has been extended to convolutional neural networks (CNNs) (Arora et al., 2019), recurrent neural networks (RNNs) (Alemohammad et al., 2021), and even to neural networks with arbitrary architectures (Yang & Littwin, 2021). Empirical studies have also provided a good understanding of the wide neural network training (Lee et al., 2020).

Federated Learning. FL aims to train a model with distributed clients without transmitting local data (McMahan et al., 2017; Kairouz et al., 2021). FedAvg has been proposed as a generic solution with theoretical analyses and implementation variants. Recent studies have shown a growing interest in improving its communication efficiency, privacy guarantees, and robustness to heterogeneity. To reduce communication cost, gradient quantization and sparsification were incorporated into FedAvg (Reisizadeh et al., 2020; Sattler et al., 2019). From the security perspective, Zhu et al. (2019) showed that sharing gradients may cause privacy leakage in the model inversion attack. To address this challenge, differentially private federated optimization and decentralized aggregation methods were developed (Girgis et al., 2021; Cheng et al., 2021). Other works put the focus on the statistical heterogeneity issue and designed methods such as adding regularization terms to the objective function (Li et al., 2020; Smith et al., 2017) or employing personalized models (Liang et al., 2019). In this work, we focus on a novel FL paradigm where the global model is derived based on the NTK evolution. We show that the proposed NTK-FL is robust to statistical heterogeneity inherently, and extend it to a variant with improved communication efficiency and enhanced privacy.

Kernel Methods in Federated Learning. The NTK framework has been mostly used for convergence analyses in FL. Seo et al. (2020) studied two knowledge distillation methods in FL and compared their convergence properties based on the neural network function evolution in the NTK regime. Li et al. (2021) incorporated batch normalization layers to local models and provided theoretical justification for its faster convergence by studying the minimum nonnegative eigenvalue of the tangent kernel matrix. To facilitate the understanding of the conventional FL process, Huang et al. (2021) directly used the NTK framework to analyze the convergence rate and generalization bound of two-layer ReLU neural networks trained with FedAvg. Likewise, Su et al. (2021) studied the convergence behavior of a set of FL algorithms in the kernel regression setting. In comparison, our work does not focus on pure convergence analyses of existing algorithms. We propose a novel FL framework by replacing the gradient descent with the NTK evolution.

3. Background and Preliminaries

Symbol conventions are as follows. We use $[N]$ to denote the set of the integers $\{1, 2, \dots, N\}$. Lowercase nonitalic boldface, nonitalic boldface capital, and italic boldface capital letters denote column vectors, matrices, and tensors, respectively. For example, for column vectors $\mathbf{a}_j \in \mathbb{R}^M$, $j \in [N]$, $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_N]$ is an $M \times N$ matrix. A third-order tensor $\boldsymbol{A} \in \mathbb{R}^{K \times M \times N}$ can be viewed as a concatenation of such matrices. We use a *slice* to denote a matrix in a third-order tensor by varying two indices (Kolda & Bader, 2009). Take tensor $\boldsymbol{A}$, for instance: $\boldsymbol{A}_{i::}$ is a matrix of the ith horizontal slice, and $\boldsymbol{A}_{:j:}$ is its jth lateral slice (Kolda & Bader, 2009). The indicator function of an event is denoted by $\mathbb{1}(\cdot)$.

3.1. Federated Learning Model

Consider an FL architecture where a server trains a global model by indirectly using datasets distributed among M workers. The local dataset of the mth worker is denoted by $\mathcal{D}_m = \{(\mathbf{x}_{m,i}, \mathbf{y}_{m,i})\}_{i=1}^{N_m}$, where $(\mathbf{x}_{m,i}, \mathbf{y}_{m,i})$ is an input-output pair. The local objective can be formulated as an empirical risk minimization over N_m training examples:

$$F_m(\mathbf{w}) = \frac{1}{N_m} \sum_{i=1}^{N_m} R(\mathbf{w}; \mathbf{x}_{m,j}, \mathbf{y}_{m,i}), \tag{1}$$

where R is a sample-wise risk function quantifying the error of model with a weight vector $\mathbf{w} \in \mathbb{R}^d$ estimating the label

$\mathbf{y}_{m,i}$ for an input $\mathbf{x}_{m,i}$. The global objective function is denoted by $F(\mathbf{w})$, and the optimization problem may be formulated as:

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) = \frac{1}{M}\sum_{m=1}^{M} F_m(\mathbf{w}). \tag{2}$$

3.2. Linearized Neural Network Model

Let $(\mathbf{x}_i, \mathbf{y}_i)$ denote a training pair, with $\mathbf{x}_i \in \mathbb{R}^{d_1}$ and $\mathbf{y}_i \in \mathbb{R}^{d_2}$, where d_1 is the input dimension and d_2 is the output dimension. $\mathbf{X} \triangleq [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top$ represents the input matrix and $\mathbf{Y} \triangleq [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\top$ represents the label matrix. Consider a neural network function $\mathbf{f} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ parameterized by a vector $\mathbf{w} \in \mathbb{R}^d$, which is the vectorization of all weights for the multilayer network. Given an input $\mathbf{x}_i$, the network outputs a prediction $\hat{\mathbf{y}}_i = \mathbf{f}(\mathbf{w}; \mathbf{x}_i)$. Let $\ell(\hat{\mathbf{y}}_i, \mathbf{y}_i)$ be the loss function measuring the dissimilarity between the predicted result $\hat{\mathbf{y}}_i$ and the true label $\mathbf{y}_i$. We are interested in finding an optimal weight vector $\mathbf{w}^\star$ that minimizes the empirical loss over N training examples:

$$\mathbf{w}^\star = \operatorname*{argmin}_{\mathbf{w}} L(\mathbf{w}; \mathbf{X}, \mathbf{Y}) \triangleq \frac{1}{N}\sum_{i=1}^{N} \ell(\hat{\mathbf{y}}_i, \mathbf{y}_i). \tag{3}$$

One common optimization method is the gradient descent training. Given the learning rate η, gradient descent updates the weights at each time step as: $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\nabla_{\mathbf{w}} L$. To simplify the notation, let $\mathbf{f}^{(t)}(\mathbf{x})$ be the output at time step t with an input $\mathbf{x}$, i.e., $\mathbf{f}^{(t)}(\mathbf{x}) \triangleq \mathbf{f}(\mathbf{w}^{(t)}; \mathbf{x})$. Following Lee et al. (2019), we use the first-order Taylor expansion around the initial weight vector $\mathbf{w}^{(0)}$ to approximate the neural network output given an input $\mathbf{x}$, i.e.,

$$\mathbf{f}^{(t)}(\mathbf{x}) \approx \mathbf{f}^{(0)}(\mathbf{x}) + \mathbf{J}^{(0)}(\mathbf{x})(\mathbf{w}^{(t)} - \mathbf{w}^{(0)}), \tag{4}$$

where $\mathbf{J}^{(0)}(\mathbf{x}) = [\nabla\mathbf{f}_1^{(0)}(\mathbf{x}), \ldots, \nabla\mathbf{f}_{d_2}^{(0)}(\mathbf{x})]^\top$, with $\nabla\mathbf{f}_j^{(t)}(\mathbf{x}) \triangleq [\partial\hat{y}_j^{(t)}/\partial w_1^{(t)}, \ldots, \partial\hat{y}_j^{(t)}/\partial w_d^{(t)}]^\top$ being the gradient of the jth component of the neural network output with respect to $\mathbf{w}^{(t)}$. Consider the halved mean-squared error (MSE) loss ℓ, namely, $\ell(\mathbf{a}, \mathbf{b}) = \frac{1}{d_2}\sum_{j=1}^{d_2}\frac{1}{2}(a_j - b_j)^2$. Based on the continuous-time limit, one can show that the dynamics of the gradient flow are governed by the following differential equation:

$$\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}t} = -\eta\,\mathbf{H}^{(0)}\big(\mathbf{f}^{(t)}(\mathbf{X}) - \mathbf{Y}\big), \tag{5}$$

where $\mathbf{f}^{(t)}(\mathbf{X}) \in \mathbb{R}^{N\times d_2}$ is a matrix of concatenated output for all training examples, and $\mathbf{H}^{(0)}$ is the neural tangent kernel at time step 0, with each entry $(\mathbf{H}^{(0)})_{ij}$ equal to the scaled Frobenius inner product of the Jacobian matrices:

$$(\mathbf{H}^{(0)})_{ij} = \frac{1}{d_2}\left\langle \mathbf{J}^{(0)}(\mathbf{x}_i),\, \mathbf{J}^{(0)}(\mathbf{x}_j)\right\rangle_{\mathrm{F}}. \tag{6}$$

The differential equation (5) has the closed-form solution:

$$\mathbf{f}^{(t)}(\mathbf{X}) = \left(\mathbf{I} - e^{-\frac{\eta t}{N}\mathbf{H}^{(0)}}\right)\mathbf{Y} + e^{-\frac{\eta t}{N}\mathbf{H}^{(0)}}\mathbf{f}^{(0)}(\mathbf{X}). \tag{7}$$

The neural network state $\mathbf{f}^{(t)}(\mathbf{X})$ can thus be directly obtained from (7) without running the gradient descent algorithm.

4. Proposed FL Paradigm via the NTK Framework

In this section, we present the NTK-FL paradigm (Figure 1) and then extend it to the variant CP-NTK-FL (Figure 2) with improved communication efficiency and enhanced privacy. The detailed algorithm descriptions are presented as follows.

4.1. NTK-FL Paradigm

NTK-FL follows four steps to update the global model in each round. **First**, the server will select a subset $\mathcal{C}_k$ of clients and broadcast to them a model weight vector $\mathbf{w}^{(k)}$ from the kth round. Here, the superscript k is the communication round index, and it should be distinguished from the gradient descent time step t in Section 3.2. **Second**, each client will use its local training data $\mathcal{D}_m$ to compute a Jacobian tensor $\boldsymbol{J}_m^{(k)} \in \mathbb{R}^{N_m\times d_2\times d}$, $\forall\ m \in \mathcal{C}_k$, which is a concatenation of N_m sample-wise Jacobian matrices $(\boldsymbol{J}_m^{(k)})_{i::} = [\nabla\mathbf{f}_1^{(k)}(\mathbf{x}_{m,i}), \ldots, \nabla\mathbf{f}_{d_2}^{(k)}(\mathbf{x}_{m,i})]^\top$, $i \in [N_m]$. The client will then upload the Jacobian tensor $\boldsymbol{J}_m^{(k)}$, labels $\mathbf{Y}_m$, and initial condition $\mathbf{f}^{(k)}(\mathbf{X}_m)$ to the server. The transmitted information corresponds to the variables in the solution for the state evolution in (7). **Third**, the server will construct a global Jacobian tensor $\boldsymbol{J}^{(k)} \in \mathbb{R}^{N_k\times d_2\times d}$ based on received $\boldsymbol{J}_m^{(k)}$'s, with each client contributing N_m horizontal slices to $\boldsymbol{J}^{(k)}$. Here, we use N_k to denote the number of training examples in communication round k when there is no ambiguity.

We use a toy example to explain the process as follows. Suppose the server selects client 1 and client 2 in a certain round. Clients 1 and 2 will compute the Jacobian tensors $\boldsymbol{J}_1^{(k)}$ and $\boldsymbol{J}_2^{(k)}$, respectively. The global Jacobian tensor is constructed as:

$$\boldsymbol{J}_{i::}^{(k)} = \begin{cases} \boldsymbol{J}_{1,i::}^{(k)} & \text{, if } i \leqslant N_1, \\ \boldsymbol{J}_{2,j::}^{(k)} & ,\, j = i - N_1,\ \text{if } i \geqslant N_1 + 1. \end{cases} \tag{8}$$

After obtaining the global Jacobian tensor $\boldsymbol{J}^{(k)}$, the (i, j)th entry of the global kernel $\mathbf{H}^{(k)}$ is calculated as the scaled Frobenius inner product of two horizontal slices of $\boldsymbol{J}^{(k)}$, i.e., $(\mathbf{H}^{(k)})_{ij} = \frac{1}{d_2}\langle\boldsymbol{J}_{i::}^{(k)}, \boldsymbol{J}_{j::}^{(k)}\rangle_{\mathrm{F}}$. **Fourth**, the server will perform the NTK evolution to obtain the updated neural network function $\mathbf{f}^{(k+1)}$ and weight vector $\mathbf{w}^{(k+1)}$. With a slight abuse of notation, let $\mathbf{f}^{(k,t)}$ denote the neural network

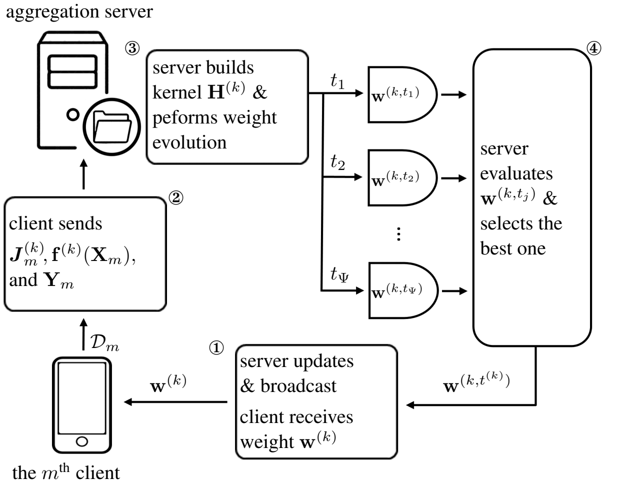

aggregation server
③
server builds kernel $\mathbf{H}^{(k)}$ & peforms weight evolution
t_1
$\mathbf{w}^{(k,t_1)}$
t_2
$\mathbf{w}^{(k,t_2)}$
⋮
t_Ψ
$\mathbf{w}^{(k,t_\Psi)}$
④
server evaluates $\mathbf{w}^{(k,t_j)}$ & selects the best one
②
client sends $\boldsymbol{J}_m^{(k)}, \mathbf{f}^{(k)}(\mathbf{X}_m)$, and $\mathbf{Y}_m$
$\mathcal{D}_m$
①
server updates & broadcast
client receives weight $\mathbf{w}^{(k)}$
$\mathbf{w}^{(k)}$
$\mathbf{w}^{(k,t^{(k)})}$
the m^{th} client

Figure 1: Schematic of NTK-FL. Each client first receives the weight $\mathbf{w}^{(k)}$, and then uploads the Jacobian tensor $\boldsymbol{J}_m^{(k)}$, label $\mathbf{Y}_m$, and initial condition $\mathbf{f}^{(k)}(\mathbf{X}_m)$. The server builds a global kernel $\mathbf{H}^{(k)}$ and performs the weight evolution with $\{t_1, \dots, t_\Psi\}$. We use (11a) to find the best t_j and update the weight accordingly.

output at gradient descent step t in communication round k. The neural network function evolution dynamics and weight evolution dynamics are given by:

$$\mathbf{f}^{(k,t)} = \left(\mathbf{I} - e^{-\frac{\eta t}{N_k}\mathbf{H}^{(k)}}\right)\mathbf{Y}^{(k)} + e^{-\frac{\eta t}{N_k}\mathbf{H}^{(k)}}\mathbf{f}^{(k)}, \tag{9a}$$

$$\mathbf{w}^{(k,t)} = \sum_{j=1}^{d_2} (\boldsymbol{J}_{:j:}^{(k)})^\top \mathbf{R}_{:j}^{(k,t)} + \mathbf{w}^{(k)}, \tag{9b}$$

where $\boldsymbol{J}_{:j:}^{(k)}$ is the jth lateral slice of $\boldsymbol{J}^{(k)}$, and $\mathbf{R}_{:j}^{(k,t)}$ is the jth column of the residual matrix $\mathbf{R}^{(k,t)}$ defined as follows:

$$\mathbf{R}^{(k,t)} \triangleq \frac{\eta}{N_k d_2} \sum_{u=0}^{t-1} \left[\mathbf{Y}^{(k)} - \mathbf{f}^{(k,u)}(\mathbf{X}^{(k)})\right]. \tag{10}$$

Here, $\mathbf{X}^{(k)}$ and $\mathbf{Y}^{(k)}$ denote a concatenation of client training examples and labels, respectively. The weight evolution in (9b) is derived by unfolding the gradient descent steps. To update the global weight, the server performs the evolution with various integer steps $\{t_1, \dots, t_\Psi\}$ and selects the best one with the smallest loss value. Our goal is to minimize the training loss with a small generalization gap (Nakkiran et al., 2020). The updated weight is decided by the following procedure:

$$t^{(k)} = \operatorname*{argmin}_{t_j} L(\mathbf{f}(\mathbf{w}^{(k,t_j)}; \mathbf{X}^{(k)}); \mathbf{Y}^{(k)}), \tag{11a}$$

$$\mathbf{w}^{(k+1)} \triangleq \mathbf{w}^{(k,t^{(k)})}. \tag{11b}$$

Alternatively, if the server has an available validation dataset, the optimal number of update steps can be selected based on the model validation performance. In practice, such a validation dataset can be obtained from held-out clients (Wang et al., 2021). Based on the closed-form solution in (9b), the search of $t^{(k)}$ over the grid $\{t_1, \dots, t_\Psi\}$ can be completed in $O(\Psi)$ time.

Robustness Against Statistical Heterogeneity. In essence, statistical heterogeneity comes from decentralized data with heterogeneous distributions owned by individual clients. If privacy is not an issue, the non-IID challenge can be readily resolved by mixing all clients' datasets and training a centralized model. In NTK-FL, instead of building a centralized dataset, we use Jacobian matrices to construct a global kernel $\mathbf{H}^{(k)}$, which is a concise representation of gathered data points from all selected clients. This representation is yet more *expressive*/less *compact* than that of a traditional FL algorithm. More precisely, the update being sent for NTK-FL regarding the ith training example of the mth client for NTK-FL is $\mathbf{J}_m = [\nabla \mathbf{f}_1(\mathbf{x}_{m,i}), \dots, \nabla \mathbf{f}_{d_2}(\mathbf{x}_{m,i})]^\top$, whereas the gradient update being sent for FedAvg is $\nabla L(\mathbf{w}; \mathbf{x}_{m,i}, \mathbf{y}_{m,i}) = \frac{1}{d_2} \sum_{j=1}^{d_2} (\hat{y}_{m,i,j} - y_{m,i,j}) \nabla \mathbf{f}_j(\mathbf{x}_{m,i})$, a weighted sum of row vectors in $\mathbf{J}_m$. The gradient will be further averaged over multiple training examples. By sending Jacobian matrices $\mathbf{J}_m$ and jointly processing them on the server, NTK-FL delays the more aggressive data aggregation step after the communication stage and therefore better approximates the centralized learning setting than FedAvg does.

Comparison of NTK-FL and Huang et al. (2021). Huang et al. (2021) presented the details of FedAvg by letting clients use local updates and upload gradients to train a two-layer neural network. In contrast, NTK-FL let each client transmit Jacobian matrices without performing local SGD steps. The model weight is updated via NTK evolution in (9b). The main differences include: (1) clients transmit more expressive Jacobian matrices to improve model performance in the non-IID FL setting; (2) more computation is shifted to the server.

4.2. CP-NTK-FL Variant

Compared to FedAvg, NTK-FL does not incur additional client computational overhead since calculating the Jacobian tensor enjoys the same computation efficiency as computing aggregated gradients. Without locally updating weight vectors, NTK-FL is faster than FedAvg on the client side. In this section, we focus on the perspectives of communication efficiency and security in terms of data confidentiality and membership privacy.

For communication, we follow the widely adopted analysis framework in wireless communication to examine only the client uplink overhead, assuming that the downlink bandwidth is much larger and the server will have enough trans-

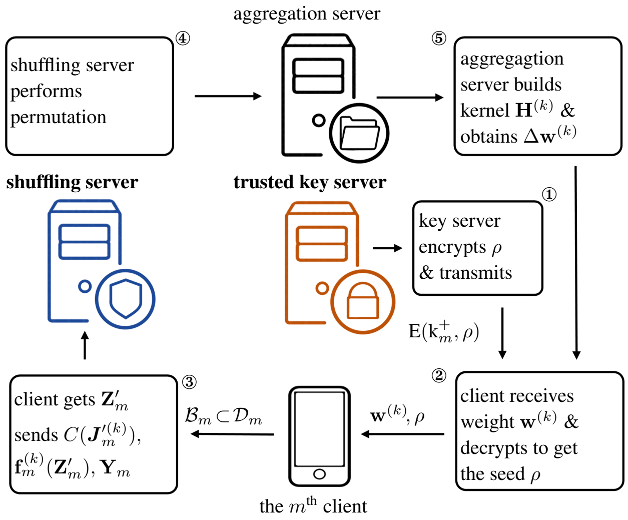

aggregation server
④
shuffling server performs permutation
⑤
aggregagtion server builds kernel $\mathbf{H}^{(k)}$ & obtains $\Delta\mathbf{w}^{(k)}$
shuffling server
trusted key server
①
key server encrypts ρ & transmits
$\mathrm{E}(\mathrm{k}_m^+, \rho)$
③
②
client gets $\mathbf{Z}'_m$ sends $C(\boldsymbol{J}'^{(k)}_m)$, $\mathbf{f}^{(k)}_m(\mathbf{Z}'_m), \mathbf{Y}_m$
$\mathcal{B}_m \subset \mathcal{D}_m$
$\mathbf{w}^{(k)}, \rho$
client receives weight $\mathbf{w}^{(k)}$ & decrypts to get the seed ρ
the m^{th} client

Figure 2: Schematic of CP-NTK-FL. A trusted key server (orange) sends an encrypted seed $\mathrm{E}(\mathrm{k}_m^+, \rho)$ with the public key k_m^+ for random projection. The client transmits the required message to the shuffling server (blue) to perform a permutation.

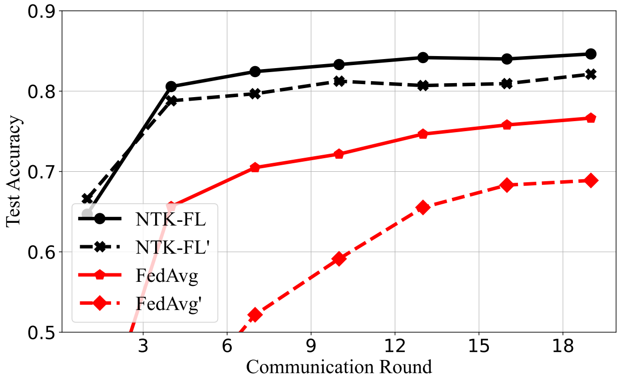


Figure 3: Training results of 300 clients via NTK-FL and FedAvg, along with variants with the local dataset subsampling and random projection, denoted as NTK-FL$'$ and FedAvg$'$, respectively. We train a two-layer multilayer perceptron on the Fashion-MNIST dataset. The joint effect causes more accuracy degradation in FedAvg (red) than in NTK-FL (black).

mission power (Tran et al., 2019). In NTK-FL, the client uplink communication cost and space complexity are dominated by a third-order tensor $\boldsymbol{J}^{(k)}_m$, i.e., an $O\left(N_m d_2 d\right)$ complexity compared to $O\left(d\right)$ in FedAvg. For security, we investigate a threat model where a curious server may perform membership inference attacks (Nasr et al., 2018) or model inversion/data reconstruction attacks (Zhu et al., 2019). Compared to the averaged gradient, sample-wise Jacobian matrices are more expressive, which may facilitate such attacks from the aggregation server. We extend NTK-FL by combining various tools to solve the aforementioned problems without jeopardizing the performance severely. These tools are optional building modules and can be adopted separately or jointly, depending on the available resources in practice. Although it is possible to incorporate these tools into FedAvg, we will show that overall it will lead to a more severe accuracy drop.

Jacobian Dimension Reduction. First, we let the mth client sample a subset $\mathcal{B}_m$ from its dataset $\mathcal{D}_m$ uniformly for the training. Let $\beta \in (0, 1)$ denote the sampling rate, $\mathcal{B}_m$ contains $N'_m = \beta N_m$ data points, with the training pairs denoted by $(\mathbf{X}'_m, \mathbf{Y}'_m)$. In general, using more data points will improve the model generalizability (Mohri et al., 2018). The sampling rate β controls the trade-off between efficiency and model performance. Next, we consider using a random projection to preprocess the input data via a seed shared by a *trusted key server*. Formally, the sampled training examples are projected into $\mathbf{Z}'_m$, i.e., $\mathbf{Z}'_m = \mathbf{X}'_m\mathbf{P}$, where $\mathbf{P} \in \mathbb{R}^{d_1 \times d'_1}$ is a projection matrix generated based on a seed ρ with IID standard Gaussian entries. In general, we have $d'_1 < d_1$ and an non-invertible projection operation. The concept of trusted key server follows the trusted third party in cryptography (Van Oorschot, 2020), and we assume it will not be compromised.

These two steps can already improve communication efficiency and confidentiality. We first examine the current Jacobian tensor $\boldsymbol{J}'^{(k)}_m \in \mathbb{R}^{N'_m \times d_2 \times d'}$. Compared with its original version $\boldsymbol{J}^{(k)}_m$, it has reduced dimensionality at the cost of certain information loss. Meanwhile, the random projection will defend against the data reconstruction attack, as the Jacobian tensor is now evaluated at the projected data $\mathbf{Z}'_m$. We empirically verify their impact on the test accuracy in Figure 3. We set $d'_1 = 100$ and sampling rate $\beta = 0.4$, and train a multilayer perceptron with 100 hidden nodes on the Fashion-MNIST dataset (Xiao et al., 2017). The joint effect of these strategies is a slight accuracy drop in NTK-FL and a nonnegligible accuracy degradation in FedAvg.

Jacobian Compression and Shuffling. We use a compression scheme to reduce the size of the Jacobian tensor by zeroing out the coordinates with small magnitudes (Alistarh et al., 2018). In addition to the communication efficiency, this compression scheme is empirically effective against the data reconstruction attack (Zhu et al., 2019). To further ensure the confidentiality and membership privacy, we introduce a *shuffling server*, inspired by some recent frameworks (Girgis et al., 2021; Cheng et al., 2021), to permute Jacobian tensors $\boldsymbol{J}^{(k)}_m$'s, neural network states $\mathbf{f}^{(k)}_m$'s, and labels $\mathbf{Y}_m$'s. Based on (9b), we denote the model update by $\Delta\mathbf{w}^{(k)} \triangleq \mathbf{w}^{(k+1)} - \mathbf{w}^{(k)} = \sum_{j=1}^{d_2}(\boldsymbol{J}^{(k)}_{:j:})^\top \mathbf{R}^{(k,t)}_{:j}$, which is a sum of matrix products. If rows and columns are permuted in synchronization, the weight update $\Delta \mathbf{w}^{(k)}$ will remain unchanged. Considering the high dimensionality of the neural network weight, the reconstruction attack becomes computationally infeasible. Since introducing a new differential privacy mechanism is not the main focus of this work and the privacy protection analysis is consistent with the existing framework (Girgis et al., 2021), we do not intend to go into details. Meanwhile, as a provable differential privacy guarantee does not explicitly protect against data reconstruction attacks (Zhang et al., 2020), we empirically study the privacy loss under the deep leakage from gradient algorithm (Zhu et al., 2019) in Appendix B. A thorough study of different attack schemes is out of the scope of this work and we leave it for future work.

5. Analysis of Algorithm

In this section, we analyze the loss decay rate between successive communication rounds in NTK-FL and make comparisons with FedAvg. Similar to prior work (Du et al., 2019; Dukler et al., 2020), we consider a two-layer neural network $f : \mathbb{R}^d \to \mathbb{R}$ of the following form to facilitate our analysis:

$$f(\mathbf{w}; \mathbf{x}) = \frac{1}{\sqrt{n}} \sum_{r=1}^{n} c_r \sigma(\mathbf{v}_r^\top \mathbf{x}), \tag{12}$$

where $\mathbf{x} \in \mathbb{R}^{d_1}$ is an input, $\mathbf{v}_r \in \mathbb{R}^{d_1}$ is the weight vector in the first layer, c_r is the weight in the second layer, and $\sigma(\cdot)$ is the rectified linear unit (ReLU) function, namely $\sigma(z) = \max(z, 0)$, applied coordinatewise. Without loss of generality, we assume the selected client set $\mathcal{C}_k$ is $\{1, 2, \dots, M_k\}$ in communication round k. Let $\mathbf{X}^{(k)} = [\mathbf{X}_1^\top, \dots, \mathbf{X}_{M_k}^\top]^\top \in \mathbb{R}^{N_k \times d_1}$ denote a concatenation of client inputs and $\mathbf{y}^{(k)} = [\mathbf{y}_1^\top, \dots, \mathbf{y}_{M_k}^\top]^\top \in \mathbb{R}^{N_k \times d_2}$ denote a concatenation of client labels. In the following analysis, we assume $d_2 = 1$ for simplicity. We state two assumptions as prerequisites.

Assumption 1 *(Weight Distribution). When broadcast in communication round k, the first layer $\mathbf{v}_r^{(k)}$'s follow the normal distribution $\mathcal{N}(0, \Sigma_k)$. The minimum eigenvalue of the covariance matrix is bounded by $\lambda_{\min}(\Sigma_k) \geqslant \alpha_k^2$, where α_k is a positive constant. The second layer c_r's are sampled from $\{-1, 1\}$ with equal probability and are kept fixed during training.*

Assumption 2 *(Normalized input). The input data are normalized, i.e., $\|\mathbf{x}_i\|_2 = 1, \forall\, i \in [N_k]$.*

For this neural network model, the (i, j)th entry of the empirical kernel matrix $\mathbf{H}^{(k)}$ given in (6) can be calculated as: $(\mathbf{H}^{(k)})_{ij} = \frac{1}{n} \mathbf{x}_i^\top \mathbf{x}_j \sum_{r=1}^{n} \mathbb{1}_{ir}^{(k)} \mathbb{1}_{jr}^{(k)}$, where $\mathbb{1}_{ir}^{(k)} \triangleq \mathbb{1}\{\langle \mathbf{v}_r^{(k)}, \mathbf{x}_i \rangle \geqslant 0\}$, and the term c_r^2 is omitted according to Assumption 1. Define $\mathbf{H}_\infty^{(k)}$, whose (i, j)th entry is given by:

$$(\mathbf{H}_\infty^{(k)})_{ij} \triangleq \mathbb{E}_{\mathbf{v}^{(k)}} \left[\mathbf{x}_i^\top \mathbf{x}_j \mathbb{1}_i^{(k)} \mathbb{1}_j^{(k)} \right], \tag{13}$$

where $\mathbb{1}_i^{(k)} \triangleq \mathbb{1}(\langle \mathbf{v}^{(k)}, \mathbf{x}_i \rangle \geqslant 0)$ and $\mathbf{v}^{(k)}$ follows the normal distribution $\mathcal{N}(0, \Sigma_k)$. Let λ_k denote the minimum eigenvalue of $\mathbf{H}_\infty^{(k)}$, which is restricted as follows.

Assumption 3 *The kernel matrix $\mathbf{H}_\infty^{(k)}$ is positive definite, namely, $\lambda_k > 0$.*

In fact, the positive-definite property of $\mathbf{H}_\infty^{(k)}$ can be shown under certain conditions (Dukler et al., 2020). For simplicity, we omit the proof details and directly assume the positive definiteness of $\mathbf{H}_\infty^{(k)}$ in Assumption 3. Next, we study the residual term $\|f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\|_2^2$. We give the convergence result by analyzing how the residual term decays given training examples $\mathbf{X}^{(k)}$.

Theorem 1 *For the NTK-FL scheme under Assumptions 1 to 3, let the learning rate $\eta = O\left(\frac{\lambda_k}{N_k}\right)$ and the neural network width $n = \Omega\left(\frac{N_k^2}{\lambda_k^2} \ln \frac{N_k^2}{\delta}\right)$, then with probability at least $1 - \delta$,*

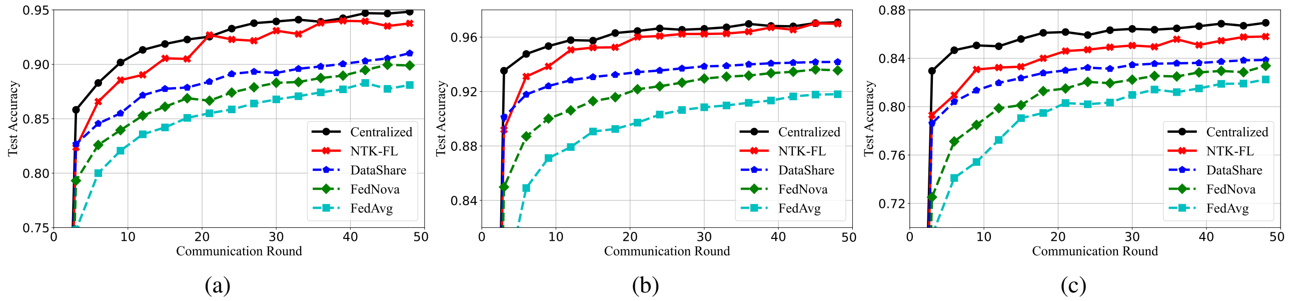
$$\left\|f^{(k+1)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\right\|_2^2 \leqslant \left(1 - \frac{\eta \lambda_k}{2N_k}\right)^{t^{(k)}} \left\|f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\right\|_2^2, \tag{14}$$

where $t^{(k)}$ is the number of update steps defined in (11a).

The proof of Theorem 1 can be found in Appendix D. We discuss the choice of the optimal number of update steps $t^{(k)}$ below.

Remark 1 *Based on the solution given by (9a), the neural network prediction error is a decreasing function of the update steps t. However, one could not pick an arbitrarily large t as the final optimal number of update steps $t^{(k)}$. When t increases, the neural network evolution in the function space does not consistently match with the evolution in the weight space. Empirical studies in the centralized training have confirmed the nonnegligible gap between the NTK weight and gradient descent weight when t is greater than the order of 10^2 to 10^3 (Lee et al., 2019). In Lemma 2 of Appendix C, we provide detailed explanations and show that the difference between the NTK weight in (9b) and the gradient descent weight increases with t.*

Next, we compare the proposed NTK-FL with FedAvg. By studying the asymmetric kernel matrix caused by local update (Huang et al., 2021), we have the following theorem for FedAvg, where the proof can be found in Appendix E.



(a) (b) (c)

Figure 4: Test accuracy versus communication round of different methods evaluated on: (a) FEMNIST dataset, where the heterogeneity comes from feature skewness. (b) non-IID MNIST dataset with label skewness ($\alpha = 0.5$). (c) non-IID Fashion-MNIST dataset with label skewness ($\alpha = 0.5$). NTK-FL outperforms all baseline FL algorithms in different scenarios and achieves similar test performance compared with the ideal centralized training case.

Theorem 2 *For FedAvg under Assumptions 1 to 3, let the learning rate* $\eta = O\left(\frac{\lambda_k}{\tau N_k M_k}\right)$ *and the neural network width* $n = \Omega\left(\frac{N_k^2}{\lambda_k^2} \ln \frac{N_k^2}{\delta}\right)$*, then with probability at least* $1-\delta$*,*

$$\left\| f^{(k+1)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2 \leqslant \left(1 - \frac{\eta \tau \lambda_k}{2 N_k M_k}\right) \left\| f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2, \quad (15)$$

where τ *is the number of local iterations, and* M_k *is the number of clients in communication round* k*.*

Remark 2 *(Fast Convergence of NTK-FL). The convergence rate of NTK-FL is faster than FedAvg. To see this, we compare the Binomial approximation of the decay coefficient in Theorem 1 with the decay coefficient in Theorem 2,*

$$1 - \frac{\eta_1 t^{(k)} \lambda_k}{2N_k} + O\left(\eta_1^2\right) < 1 - \frac{\eta_2 \tau \lambda_k}{2 N_k M_k}, \quad (16)$$

where $\eta_1 \ll 1$ *for a large* N_k*.* [1] *The number of NTK update steps* $t^{(k)}$ *is chosen dynamically in (11a), which is on the order of* 10^2 *to* 10^3*, whereas* τ *is often on the order of magnitude of* 10 *in the literature (Reisizadeh et al., 2020; Haddadpour et al., 2021). One can verify that* $\eta_1 t^{(k)} \lambda_k$ *is larger than* $\eta_2 \tau \lambda_k / M_k$ *and draw the conclusion.*

6. Experimental Results

Federated Settings. We use three datasets, namely, MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), and FEMNIST (Caldas et al., 2018) digits. All of them contain $C = 10$ categories. For MNIST and Fashion-MNIST, we follow Hsu et al. (2019) to simulate non-IID data with the symmetric Dirichlet distribution (Good, 1976). Specifically, for the mth client, we draw a random vector $\boldsymbol{q}_m \sim \mathrm{Dir}(\alpha)$, where $\boldsymbol{q}_m = [q_{m,1}, \dots, q_{m,C}]^\top$ belongs to the $(C-1)$-standard simplex. Images with category k are assigned to the mth client in proportional to $(100 \cdot q_{m,k})\%$. The heterogeneity in this setting mainly comes from label skewness. FEMNIST splits the dataset into shards indexed by the original writer of the digits. The heterogeneity mainly comes from feature skewness. A multilayer perceptron model with 100 hidden nodes is chosen as the target neural network model. We consider a total of 300 clients and select 20 of them with equal probability in each round.

Convergence. We empirically verify the convergence rate of the proposed method. For FedAvg, we use the number of local iterations from $\{1, 3, \dots, 9, 10, 20, \dots, 50\}$ and report the best results. For NTK-FL, we choose $t^{(k)}$ over the set $\{100, 200, \dots, 2000\}$. We use the following methods that are robust to the non-IID setting as the baselines: (i) Data sharing scheme suggested by Zhao et al. (2018), where a global dataset is broadcast to clients for local training; the size of the global dataset is set to be 10% of the total number of local data points. (ii) Federated normalized averaging (FedNova) (Wang et al., 2020), where the clients transmit normalized gradient vectors to the server. (iii) Centralized training simulation, where the server collects the data points from subset $\mathcal{C}_k$ of clients and performs gradient descent to directly train the global model. Scheme (iii) achieves the performance that can be considered as an upper bound of all other algorithms. The training curves over three repetitions are shown in Figure 4. More implementation details and the results on CIFAR-10 can be found in Appendix A. Our proposed NTK-FL method shows consistent advantages over other methods in different non-IID scenarios.

Degree of Heterogeneity. In this experiment, we select the Dirichlet distribution parameter α from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ and simulate different degrees of heterogeneity on Fashion-MNIST dataset. A smaller α

[1] For example, if we have 100 clients, each of which has 100 data points, then N_k is on the order of 10^4. Considering the choice of the learning rate η_1, the Binomial approximation holds.

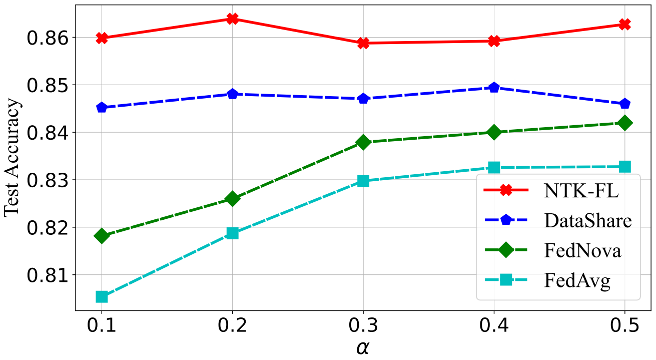


Figure 5: Test accuracy versus the Dirichlet distribution parameter α for different methods evaluated on the non-IID Fashion-MNIST dataset. Reducing the value of α will increase the degree of heterogeneity in the data distribution. NTK-FL is robust to different heterogeneous data distributions, and shows more advantages over FedAvg and FedNova when the degree of heterogeneity is larger.

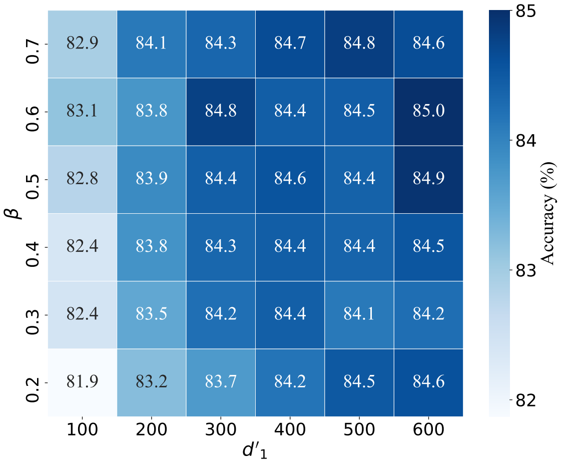


Figure 6: CP-NTK-FL test accuracy for different hyperparams. A larger data sampling rate β and a larger dimension d'_1 are expected to give a higher test accuracy. In general, the scheme is robust to different combinations of hyperparameters.

will increase the degree of heterogeneity in the data distribution. We evaluate NTK-FL, DataShare, FedNova, and FedAvg model test accuracy after training for 50 rounds. The mean values over three repetitions are shown in Figure 5, where each point is obtained over five repetitions with a standard deviation less than 1%. It can be observed that NTK-FL achieves stable test accuracy in different heterogeneous settings. In comparison, FedAvg and FedNova show a performance drop in the small α region. NTK-FL has more advantages over baselines methods when the degree of heterogeneity is larger.

Effect of Hyperparameters. We study the effect of the tunable parameters in CP-NTK-FL. We change the local data sampling rate β and dimension d'_1, and evaluate the model test accuracy on the non-IID Fashion-MNIST dataset ($\alpha = 0.1$) after 10 communication rounds. The results are shown in Figure 6. A larger data sampling rate β or a larger dimension d'_1 will cause less information loss and are expected to achieve a higher test accuracy. The results also show that the scheme is robust to different combinations of hyperparameters.

Uplink Communication. In federated learning, uplink communication overhead can be one of the bottlenecks in the training stage. We evaluate the uplink communication efficiency of CP-NTK-FL ($d'_1 = 200, \beta = 0.3$) by measuring the number of rounds and cumulative uplink communication cost to reach a test accuracy of 85% on non-IID Fashion-MNIST dataset ($\alpha = 0.1$). The results over three repetitions are shown in Table 1. Compared with federated learning with compression (FedCOM) (Haddadpour et al., 2021), quantized SGD (QSGD) (Alistarh et al., 2017), and FedAvg, CP-NTK-FL achieves the goal within an order of magnitude fewer iterations, which is particularly advantageous for applications with nonnegligible encoding/decoding delays or network latency.

Table 1: Uplink communication cost to reach an accuracy of 85% on non-IID Fashion-MNIST dataset ($\alpha = 0.1$) for different methods. CP-NTK-FL can achieve the target within the fewest communication rounds without incurring communication cost significantly.

optimization algorithms	comm. rounds	comm. cost (MB)
CP-NTK-FL	26	386
FedCOM	250	379
QSGD (4 bit)	614	465
FedAvg	284	1720

7. Conclusion and Future Work

In this paper, we have proposed an NTK empowered FL paradigm. It inherently solves the statistical heterogeneity challenge. By constructing a global kernel based on the local sample-wise Jacobian matrices, the global model weights can be updated via NTK evolution in the parameter space. Compared with traditional algorithms such as FedAvg, NTK-FL has a more centralized training flavor by transmitting more expressive updates. The effectiveness of the proposed paradigm has been verified theoretically and experimentally.

In future work, it will be interesting to extend the paradigm for other neural network architectures, such as CNNs, residual networks (ResNets) (He et al., 2016), and RNNs. It is also worthwhile to further improve the efficiency of NTK-FL and explore its savings in wall-clock time. We believe the proposed paradigm will provide a new perspective to solve federated learning challenges.

References

Aji, A. F. and Heafield, K. Sparse communication for distributed gradient descent. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

Alemohammad, S., Wang, Z., Balestriero, R., and Baraniuk, R. The recurrent neural tangent kernel. In *International Conference on Learning Representations*, 2021.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. QSGD: Communication-efficient SGD via gradient quantization and encoding. 2017.

Alistarh, D., Hoefler, T., Johansson, M., Konstantinov, N., Khirirat, S., and Renggli, C. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems*, 2018.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, 2019.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Chen, H.-Y. and Chao, W.-L. FedBE: Making Bayesian model ensemble applicable to federated learning. In *International Conference on Learning Representations*, 2021.

Chen, Z., Cao, Y., Gu, Q., and Zhang, T. A generalized neural tangent kernel analysis for two-layer neural networks. In *Advances in Neural Information Processing Systems*, 2020.

Cheng, P.-C., Eykholt, K., Gu, Z., Jamjoom, H., Jayaram, K., Valdez, E., and Verma, A. Separation of powers in federated learning. *arXiv preprint arXiv:2105.09400*, 2021.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

Dukler, Y., Montufar, G., and Gu, Q. Optimization theory for ReLU neural networks trained with normalization layers. In *International Conference on Machine Learning*, 2020.

Girgis, A. M., Data, D., Diggavi, S. N., Kairouz, P., and Suresh, A. T. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Gonzalez, R. C. and Woods, R. E. *Digital Image Processing, 3rd Edition*. 2014.

Good, I. J. On the application of symmetric Dirichlet distributions and their mixtures to contingency tables. *The Annals of Statistics*, 4(6):1159–1189, 1976.

Haddadpour, F., Kamani, M. M., Mokhtari, A., and Mahdavi, M. Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *International Conference on Computer Vision and Pattern Recognition*, 2016.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Huang, B., Li, X., Song, Z., and Yang, X. FL-NTK: A neural tangent kernel-based framework for federated learning convergence analysis. *arXiv preprint arXiv:2105.05001*, 2021.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 2021.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 2020.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Krizhevsky, A. Learning multiple layers of features from tiny images. *Master thesis, Dept. of Comput. Sci., Univ. of Toronto, Toronto, Canada*, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, 2019.

Lee, J., Schoenholz, S. S., Pennington, J., Adlam, B., Xiao, L., Novak, R., and Sohl-Dickstein, J. Finite versus infinite neural networks: An empirical study. In *Advances in Neural Information Processing Systems*, 2020.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. 2020.

Li, X., Jiang, M., Zhang, X., Kamp, M., and Dou, Q. FedBN: Federated learning on non-iid features via local batch normalization. In *International Conference on Learning Representations*, 2021.

Liang, P. P., Liu, T., Ziyin, L., Allen, N. B., Auerbach, R. P., Brent, D., Salakhutdinov, R., and Morency, L.-P. Think locally, act globally: Federated learning with local and global representations. *NeurIPS Workshop on Federated Learning*, 2019.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 2017.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020.

Nasr, M., Shokri, R., and Houmansadr, A. Machine learning with membership privacy using adversarial regularization. In *ACM SIGSAC Conference on Computer and Communications Security*, pp. 634–646, 2018.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., and Pedarsani, R. FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Sattler, F., Wiedemann, S., Müller, K.-R., and Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3400–3413, 2019.

Seo, H., Park, J., Oh, S., Bennis, M., and Kim, S.-L. Federated knowledge distillation. *arXiv preprint arXiv:2011.02367*, 2020.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, 2017.

Su, L., Xu, J., and Yang, P. Achieving statistical optimality of federated learning: Beyond stationary points. *arXiv preprint arXiv:2106.15216*, 2021.

Tran, N. H., Bao, W., Zomaya, A., Nguyen, M. N., and Hong, C. S. Federated learning over wireless networks: Optimization model design and analysis. In *IEEE Conference on Computer Communications*, pp. 1387–1395. IEEE, 2019.

Van Oorschot, P. C. *Computer Security and the Internet: Tools and Jewels*. Springer Nature, 2020.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., and Khazaeni, Y. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, 2020.

Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yang, G. and Littwin, E. Tensor programs IIb: Architectural universality of neural tangent kernel training dynamics. In *International Conference on Machine Learning*, 2021.

Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., and Song, D. The secret revealer: Generative model-inversion attacks against deep neural networks. In *IEEE/CVF Conference* *on Computer Vision and Pattern Recognition*, pp. 253–261, 2020.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Zhu, L., Liu, Z., and Han, S. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, 2019.

A. Implementation and Additional Results

We give the detailed setting of the learning rate and batch size. For the learning rate η, we search over the set $\{10^{-3}, 3 \times 10^{-3}, 10^{-2}, 3 \times 10^{-2}, 10^{-1}\}$. The learning rate is fixed during the training. For the client batch size, we set it to 200 for all datasets. We use the same setup as in Section 6. We evaluate different methods, including the centralized training simulation, data sharing method (Zhao et al., 2018), FedNova (Wang et al., 2020), FedAvg (McMahan et al., 2017), and the proposed NTK-FL on the non-IID CIFAR-10 dataset (Krizhevsky, 2009) and present the results in Figure 7. NTK-FL outperforms other FL algorithms and shows test accuracy close to the centralized simulation. The observation is consistent with the results in Figure 4. The implementation is available at `https://github.com/KAI-YUE/ntk-fed`.

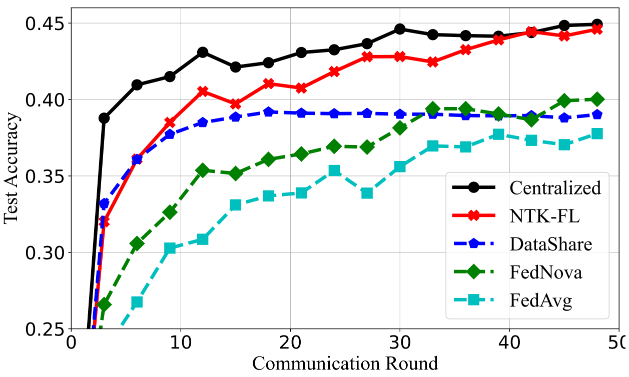


Figure 7: Test accuracy versus communication round of different methods evaluated on the non-IID CIFAR-10 dataset, where the Dirichlet distribution parameter α is set to 0.1.

B. Reconstruction Attack

In the following experiment, we measure the privacy loss when using Jacobian sparsification under the data reconstruction attack. We compress the Jacobian tensor and perform the deep leakage attack (Zhu et al., 2019). The sparsity levels and image structural similarity index measure (SSIM) (Gonzalez & Woods, 2014) are shown in Figure 8. When the sparsity decreases, the reconstructions are closer to original images, which is consistent with Zhu et al. (2019). The sparsity is set above 80%–90% when using the Top-k sparsification approach (Aji & Heafield, 2017), where the privacy loss becomes difficult to quantify. A solid privacy-protection study is nontrivial and we leave it for future work.

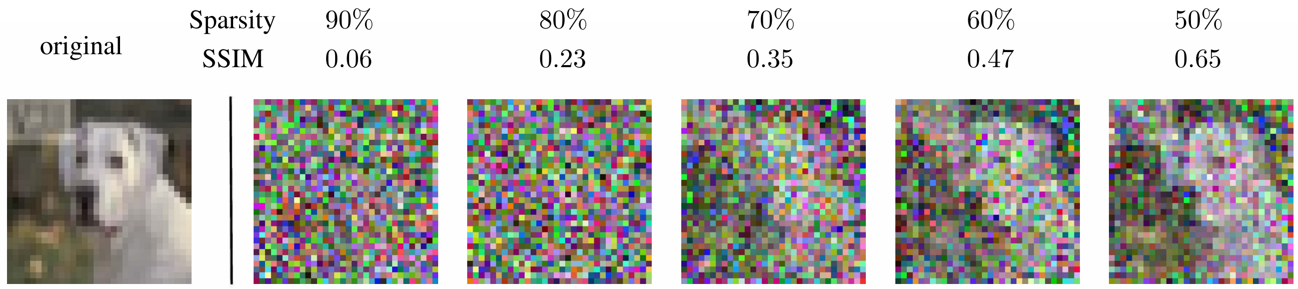


Figure 8: Attack results under different sparsity levels.

C. The Number of Update Steps

We explain below the reason that the optimal number of update steps, $t^{(k)}$, must not be too large in (11a). The weight $\mathbf{w}^{(k,t)}$ is generated by the NTK method in (9b). Let $\widetilde{\mathbf{w}}^{(k,t)}$ denote the weight generated by the gradient descent method. For the two-layer neural network model given in (12), we will show the upper bound on the gap between $\mathbf{w}^{(k,t)}$ and $\widetilde{\mathbf{w}}^{(k,t)}$. To this end, we first give Lemma 1 as a prerequisite. Define $\mathcal{S}_i^{(k)}$ as the set of indices corresponding to neurons whose activation patterns are different from the broadcast version $\mathbf{v}^{(k)}$ for an input $\mathbf{x}_i$:

$$\mathcal{S}_i^{(k)} \triangleq \left\{ r \in [n] \mid \exists\, \mathbf{v}^{(k)}, \|\mathbf{v}^{(k)} - \mathbf{v}_r^{(k)}\|_2 \leqslant R,\, \mathbb{1}_{ir}^{(k)} \neq \mathbb{1}\left(\langle \mathbf{v}^{(k)}, \mathbf{x}_i \rangle \geqslant 0\right) \right\}. \tag{17}$$

We upper bound the cardinality of $\mathcal{S}_i^{(k)}$ in Lemma 1.

Lemma 1 *Under Assumptions 1 to 2, with probability at least* $1 - \delta$*, we have*

$$|\mathcal{S}_i^{(k)}| \leqslant \sqrt{\frac{2}{\pi}} \frac{nR}{\delta \alpha_k}, \quad \forall\, i \in [N_k]. \tag{18}$$

Proof. To bound $|\mathcal{S}_i^{(k)}| = \sum_{r=1}^{n} \mathbb{1}\left(r \in \mathcal{S}_i^{(k)}\right)$, consider an event $A_{ir}^{(k)}$ defined as follows:

$$A_{ir}^{(k)} \triangleq \left\{\exists\, \mathbf{v}^{(k)}, \|\mathbf{v}^{(k)} - \mathbf{v}_r^{(k)}\|_2 \leqslant R,\, \mathbb{1}_{ir}^{(k)} \neq \mathbb{1}\left(\langle \mathbf{v}^{(k)}, \mathbf{x}_i\rangle \geqslant 0\right)\right\}. \tag{19}$$

Clearly, $\mathbb{1}\left(r \in \mathcal{S}_i^{(k)}\right) = \mathbb{1}\left(A_{ir}^{(k)}\right)$. According to Assumption 2, it can be shown that the event $A_{ir}^{(k)}$ happens if and only if $|(\mathbf{v}_r^{(k)})^\top \mathbf{x}_i| \leqslant R$ based on a geometric argument. From Assumption 1, we have $(\mathbf{v}_r^{(k)})^\top \mathbf{x}_i \sim \mathcal{N}(0, \mathbf{x}_i^\top \Sigma_k \mathbf{x}_i)$. The probability of event $A_{ir}^{(k)}$ is

$$\mathbb{P}[A_{ir}^{(k)}] = \mathbb{P}\left[|(\mathbf{v}_r^{(k)})^\top \mathbf{x}_i| \leqslant R\right] \tag{20a}$$

$$= \text{erf}\left(\frac{R}{\sqrt{2\mathbf{x}_i^\top \Sigma_k \mathbf{x}_i}}\right) \leqslant \sqrt{\frac{2}{\pi}}\frac{R}{\alpha_k}, \tag{20b}$$

where the error function is given by $\text{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}\,\mathrm{d}t$. By Markov's inequality, we have with probability at least $1-\delta$,

$$\sum_{r=1}^{n} \mathbb{1}\left(r \in \mathcal{S}_i^{(k)}\right) \leqslant \sqrt{\frac{2}{\pi}}\frac{nR}{\delta\alpha_k}. \tag{21}$$

The proof is complete. □

Lemma 2 *Consider the residual term* $\mathbf{r}^{(k,u)} = \mathbf{y}^{(k)} - f^{(k,u)}(\mathbf{X}^{(k)})$. *Suppose* $\forall i \in [N_k]$, $|r_i^{(k,u)}| < \gamma^{(k,u)}$, *where* $\gamma^{(k,u)}$ *is a positive constant and* $\lim_{u\to\infty} \gamma^{(k,u)} = 0$. *With probability at least* $1-\delta$, *we have*

$$\|\mathbf{w}^{(k,t)} - \widetilde{\mathbf{w}}^{(k,t)}\|_1 \leqslant \frac{\sqrt{2nd_1}\eta}{\sqrt{\pi}\delta\alpha_k}\sum_{u=1}^{t-1}\gamma^{(k,u)}. \tag{22}$$

Proof. The weights $\mathbf{w}^{(k,t)}$ and $\widetilde{\mathbf{w}}^{(k,t)}$ can be written as

$$\mathbf{w}^{(k,t)} = \frac{\eta}{N_k}\sum_{u=0}^{t-1}\nabla f^{(k,0)}(\mathbf{X}^{(k)})\left[\mathbf{y}^{(k)} - f^{(k,u)}(\mathbf{X}^{(k)})\right] + \mathbf{w}^{(k,0)}, \tag{23}$$

$$\widetilde{\mathbf{w}}^{(k,t)} = \frac{\eta}{N_k}\sum_{u=0}^{t-1}\nabla f^{(k,u)}(\mathbf{X}^{(k)})\left[\mathbf{y}^{(k)} - f^{(k,u)}(\mathbf{X}^{(k)})\right] + \mathbf{w}^{(k,0)}. \tag{24}$$

The ℓ_1 norm of the difference between $\mathbf{w}^{(k,t)}$ and $\widetilde{\mathbf{w}}^{(k,t)}$ is

$$\|\mathbf{w}^{(k,t)} - \widetilde{\mathbf{w}}^{(k,t)}\|_1 = \left\|\frac{\eta}{N_k}\sum_{u=1}^{t-1}\left[\nabla f^{(k,u)}(\mathbf{X}^{(k)}) - \nabla f^{(k,0)}(\mathbf{X}^{(k)})\right]\left[\mathbf{y}^{(k)} - f^{(k,u)}(\mathbf{X}^{(k)})\right]\right\|_1 \tag{25a}$$

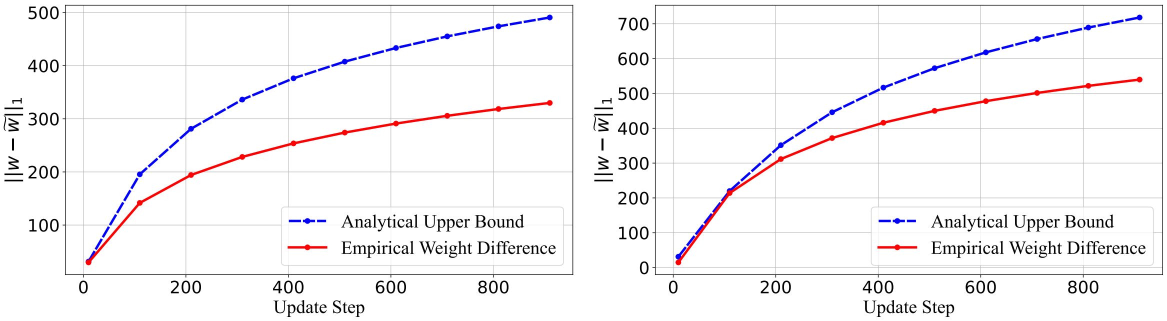
$$\leqslant \frac{\eta}{N_k}\sum_{u=1}^{t-1}\gamma^{(k,u)}\sum_{i=1}^{N_k}\left\|\frac{1}{\sqrt{n}}\sum_{r=1}^{n}c_r\mathbf{x}_i(\mathbb{1}_{ir}^{(k,t)} - \mathbb{1}_{ir}^{(k,0)})\right\|_1 \tag{25b}$$

$$\leqslant \frac{\eta}{N_k}\sum_{u=1}^{t-1}\gamma^{(k,u)}\sum_{i=1}^{N_k}\sum_{r=1}^{n}\frac{\sqrt{d_1}}{\sqrt{n}}|\mathbb{1}_{ir}^{(k,t)} - \mathbb{1}_{ir}^{(k,0)}| \tag{25c}$$

$$= \frac{\eta}{N_k}\sum_{u=1}^{t-1}\gamma^{(k,u)}\sum_{i=1}^{N_k}\sum_{r=1}^{n}\frac{\sqrt{d_1}}{\sqrt{n}}\mathbb{1}(r \in \mathcal{S}_i^{(k)}) \tag{25d}$$

$$\leqslant \frac{\sqrt{2nd_1}\eta}{\sqrt{\pi}\delta\alpha_k}\sum_{u=1}^{t-1}\gamma^{(k,u)}. \tag{25e}$$

□



(a)

(b)

Figure 9: The ℓ_1 norm of the difference between the NTK weight $\mathbf{w}^{(k,t)}$ and gradient descent weight $\widetilde{\mathbf{w}}^{(k,t)}$ on the (a) Fashion-MNIST dataset and (b) FEMNIST dataset. For the analytical upper bound, we calculate the term $\sum_u \gamma^{(k,u)}$ in (25e) and set the coefficient to 2. Both the theoretical result and real experiments show that the weight difference increases when more update steps are used.

The theoretical result indicates that the difference between the NTK weight $\mathbf{w}^{(k,t)}$ and gradient descent weight $\widetilde{\mathbf{w}}^{(k,t)}$ increases with the number of update steps t. We validate our theoretical result using real experiments and report the results in Figure 9. The consistent trend between the empirical weight difference and the analytical upper bound confirms our analysis that increasing the number of update steps enlarges the gap between $\mathbf{w}^{(k,t)}$ and $\widetilde{\mathbf{w}}^{(k,t)}$. In the NTK evolution scheme, one cannot choose an arbitrarily large t as the final number of update step $t^{(k)}$.

D. Proof of Theorem 1

We first present some lemmas to facilitate the convergence analysis. We bound the perturbation of the kernel matrix $\mathbf{H}^{(k,t)}$ in Lemma 3.

Lemma 3 *Under Assumptions 1 to 2, if* $\forall\, r \in [n]$, $\|\mathbf{v}_r^{(k,t)} - \mathbf{v}_r^{(k)}\|_2 \leqslant R$, *then*

$$\|\mathbf{H}^{(k,t)} - \mathbf{H}^{(k)}\|_2 \leqslant \frac{2\sqrt{2}N_k R}{\sqrt{\pi}\delta\alpha_k}. \tag{26}$$

Proof. We have

$$\begin{aligned}
\|\mathbf{H}^{(k,t)} - \mathbf{H}^{(k)}\|_2^2 &\leqslant \|\mathbf{H}^{(k,t)} - \mathbf{H}^{(k)}\|_{\mathrm{F}}^2 & \text{(27a)}\\
&= \sum_{i=1}^{N_k}\sum_{j=1}^{N_k}\Big[(\mathbf{H}^{(k,t)})_{ij} - (\mathbf{H}^{(k)})_{ij}\Big]^2 & \text{(27b)}\\
&= \frac{1}{n^2}\sum_{i=1}^{N_k}\sum_{j=1}^{N_k}(\mathbf{x}_i^\top \mathbf{x}_j)^2\left(\sum_{r=1}^{n}\mathbb{1}_{ir}^{(k,t)}\mathbb{1}_{jr}^{(k,t)} - \mathbb{1}_{ir}^{(k)}\mathbb{1}_{jr}^{(k)}\right)^2. & \text{(27c)}
\end{aligned}$$

Consider the event A_{ir} defined in (19). Let $\phi_{ijr}^{(k,t)} \triangleq \mathbb{1}_{ir}^{(k,t)}\mathbb{1}_{jr}^{(k,t)} - \mathbb{1}_{ir}^{(k)}\mathbb{1}_{jr}^{(k)}$. If $\neg A_{ir}$ and $\neg A_{jr}$ happen, clearly we have $|\phi_{ijr}^{(k,t)}| = 0$. Therefore, the expectation of $|\phi_{ijr}^{(k,t)}|$ can be bounded as

$$\begin{aligned}
\mathbb{E}\left[\left|\phi_{ijr}^{(k,t)}\right|\right] &\leqslant \mathbb{P}(A_{ir} \cup A_{jr}) \cdot 1 & \text{(28a)}\\
&\leqslant \mathbb{P}(A_{ir}) + \mathbb{P}(A_{jr}) & \text{(28b)}\\
&\overset{\textcircled{1}}{\leqslant} 2\sqrt{\frac{2}{\pi}}\frac{R}{\alpha_k}, & \text{(28c)}
\end{aligned}$$

where ① comes from (20b). By Markov's inequality, we have with probability at least $1-\delta$,

$$|\phi_{ijr}^{(k,t)}| \leqslant 2\sqrt{\frac{2}{\pi}}\frac{R}{\delta\alpha_k}. \tag{29}$$

Plugging (29) into (27c) yields

$$\|\mathbf{H}^{(k,t)} - \mathbf{H}^{(k)}\|_2^2 \leqslant \frac{N_k^2}{n^2} \frac{8n^2R^2}{\pi\delta^2\alpha_k^2} = \frac{8N_k^2R^2}{\pi\delta^2\alpha_k^2}. \tag{30}$$

Taking the square root on both sides completes the proof. □

Lemma 4 *With probability at least* $1-\delta$*,*

$$\|\mathbf{H}^{(k)} - \mathbf{H}_\infty^{(k)}\|_2 \leqslant N_k\sqrt{\frac{\ln\left(2N_k^2/\delta\right)}{2n}}. \tag{31}$$

Proof. We have

$$\|\mathbf{H}^{(k)} - \mathbf{H}_\infty^{(k)}\|_2^2 \leqslant \|\mathbf{H}^{(k)} - \mathbf{H}_\infty^{(k)}\|_\mathrm{F}^2 = \sum_{i=1}^{N_k}\sum_{j=1}^{N_k}\left[(\mathbf{H}^{(k)})_{ij} - (\mathbf{H}_\infty^{(k)})_{ij}\right]^2. \tag{32}$$

Note that $(\mathbf{H}^{(k)})_{ij} = \frac{1}{n}\mathbf{x}_i^\top\mathbf{x}_j\sum\limits_{r=1}^{n}\mathbb{1}_{ir}^{(k)}\mathbb{1}_{jr}^{(k)}$, $(\mathbf{H}^{(k)})_{ij} \in [-1,1]$. By Hoeffding's inequality, we have with probability at least $1-\delta/N_k^2$,

$$\left|(\mathbf{H}^{(k)})_{ij} - (\mathbf{H}_\infty^{(k)})_{ij}\right| \leqslant \sqrt{\frac{\ln\left(2N_k^2/\delta\right)}{2n}}. \tag{33}$$

Applying the union bound over $i,j \in [N_k]$ yields

$$\|\mathbf{H}^{(k)} - \mathbf{H}_\infty^{(k)}\|_2 \leqslant N_k\sqrt{\frac{\ln\left(2N_k^2/\delta\right)}{2n}}. \tag{34}$$

The proof is complete. □

Now we are going to prove Theorem 1.

Theorem 1 *For the NTK-FL scheme under Assumptions 1 to 3, let the learning rate* $\eta = O\left(\frac{\lambda_k}{N_k}\right)$ *and the neural network width* $n = \Omega\left(\frac{N_k^2}{\lambda_k^2}\ln\frac{N_k^2}{\delta}\right)$*, then with probability at least* $1-\delta$*,*

$$\left\|f^{(k+1)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\right\|_2^2 \leqslant \left(1 - \frac{\eta\lambda_k}{2N_k}\right)^{t^{(k)}}\left\|f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\right\|_2^2, \tag{35}$$

where $t^{(k)}$ *is the number of update steps defined in* (11a)*.*

Proof. Taking the difference between successive neural network predictions yields

$$f^{(k,t+1)}(\mathbf{x}_i) - f^{(k,t)}(\mathbf{x}_i) = \frac{1}{\sqrt{n}}\sum_{r=1}^{n}\left[c_r\sigma\left((\mathbf{v}_r^{(k,t+1)})^\top\mathbf{x}_i\right) - c_r\sigma\left((\mathbf{v}_r^{(k,t)})^\top\mathbf{x}_i\right)\right]. \tag{36}$$

We decompose the difference term to the sum of d_i^I and d_i^II, based on the set $\mathcal{S}_i^{(k)}$:

$$d_i^\mathrm{I} \triangleq \frac{1}{\sqrt{n}}\sum_{r\notin\mathcal{S}_i^{(k)}}\left[c_r\sigma\left((\mathbf{v}_r^{(k,t+1)})^\top\mathbf{x}_i\right) - c_r\sigma\left((\mathbf{v}_r^{(k,t)})^\top\mathbf{x}_i\right)\right], \tag{37a}$$

$$d_i^\mathrm{II} \triangleq \frac{1}{\sqrt{n}}\sum_{r\in\mathcal{S}_i^{(k)}}\left[c_r\sigma\left((\mathbf{v}_r^{(k,t+1)})^\top\mathbf{x}_i\right) - c_r\sigma\left((\mathbf{v}_r^{(k,t)})^\top\mathbf{x}_i\right)\right]. \tag{37b}$$

Consider the residual term

$$\left\|f^{(k,t+1)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\right\|_2^2 \tag{38a}$$

$$= \left\|f^{(k,t+1)}(\mathbf{X}^{(k)}) - f^{(k,t)}(\mathbf{X}^{(k)}) + f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\right\|_2^2 \tag{38b}$$

$$= \left\|f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\right\|_2^2 + 2\left\langle\mathbf{d}^\mathrm{I} + \mathbf{d}^\mathrm{II}, f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\right\rangle + \left\|f^{(k,t+1)}(\mathbf{X}^{(k)}) - f^{(k,t)}(\mathbf{X}^{(k)})\right\|_2^2. \tag{38c}$$

We will give upper bounds for the inner product terms $\left\langle \mathbf{d}^{\mathrm{I}}, f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\rangle$, $\left\langle \mathbf{d}^{\mathrm{II}}, f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\rangle$, and the difference term $\left\| f^{(k,t+1)}(\mathbf{X}^{(k)}) - f^{(k,t)}(\mathbf{X}^{(k)}) \right\|_2^2$, separately. Based on the property of the set $\mathcal{S}_i^{(k)}$, we have

$$d_i^{\mathrm{I}} = -\frac{\eta}{\sqrt{n}} \sum_{r \notin S_i} c_r \left\langle \nabla_{\mathbf{v}_r} L, \mathbf{x}_i \right\rangle \mathbb{1}_{ir}^{(k,t)} \tag{39a}$$

$$= -\frac{\eta}{n N_k} \sum_{j=1}^{N_k} \left(f^{(k,t)}(\mathbf{x}_j) - y_j \right) \mathbf{x}_j^\top \mathbf{x}_i \sum_{r \notin S_i} c_r^2 \, \mathbb{1}_{ir}^{(k,t)} \mathbb{1}_{jr}^{(k,t)} \tag{39b}$$

$$= -\frac{\eta}{N_k} \sum_{j=1}^{N_k} \left(f^{(k,t)}(\mathbf{x}_j) - y_j \right) \left((\mathbf{H}^{(k,t)})_{ij} - (\mathbf{H}^{\perp(k,t)})_{ij} \right), \tag{39c}$$

where $(\mathbf{H}^{\perp(k,t)})_{ij}$ is defined as

$$(\mathbf{H}^{\perp(k,t)})_{ij} \triangleq \frac{1}{n} \mathbf{x}_i^\top \mathbf{x}_j \sum_{r \in \mathcal{S}_i^{(k)}}^{n} \mathbb{1}_{ir}^{(k,t)} \mathbb{1}_{jr}^{(k,t)}. \tag{40}$$

For the inner product term $\left\langle \mathbf{d}^{\mathrm{I}}, f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\rangle$, we have

$$\left\langle \mathbf{d}^{\mathrm{I}}, f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\rangle = -\frac{\eta}{N_k} (f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top (\mathbf{H}^{(k,t)} - \mathbf{H}^{\perp(k,t)}) (f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}). \tag{41}$$

Let T_1 and T_2 denote the following terms

$$T_1 \triangleq -(f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top \mathbf{H}^{(k,t)} (f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}), \tag{42a}$$

$$T_2 \triangleq (f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top \mathbf{H}^{\perp(k,t)} (f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}). \tag{42b}$$

With probability at least $1 - \delta$, T_1 can be bounded as:

$$T_1 = -(f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top (\mathbf{H}^{(k,t)} - \mathbf{H}^{(k)} + \mathbf{H}^{(k)} - \mathbf{H}_\infty^{(k)} + \mathbf{H}_\infty^{(k)}) (f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}) \tag{43a}$$

$$\begin{aligned} &\leqslant -(f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top (\mathbf{H}^{(k,t)} - \mathbf{H}^{(k)}) (f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}) \\ &\quad - (f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top (\mathbf{H}^{(k)} - \mathbf{H}_\infty^{(k)}) (f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}) - \lambda_k \left\| f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2 \end{aligned} \tag{43b}$$

$$\overset{①}{\leqslant} \left(\frac{2\sqrt{2} N_k R}{\sqrt{\pi} \delta \alpha_k} + N_k \sqrt{\frac{\ln{(2N_k^2/\delta)}}{2n}} - \lambda_k \right) \left\| f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2, \tag{43c}$$

where ① comes from Lemma 3 and Lemma 4. To bound the term T_2, consider the ℓ_2 norm of the matrix $\mathbf{H}^{\perp(k,t)}$. With probability at least $1 - \delta$, we have:

$$\|\mathbf{H}^{\perp(k,t)}\|_2 \leqslant \|\mathbf{H}^{\perp(k,t)}\|_{\mathrm{F}} \tag{44a}$$

$$= \left(\sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \left(\frac{1}{n} \sum_{r \in \mathcal{S}_i^{(k)}} \mathbf{x}_i^\top \mathbf{x}_j \mathbb{1}_{ir}^{(k,t)} \mathbb{1}_{jr}^{(k,t)} \right)^2 \right)^{\frac{1}{2}} \tag{44b}$$

$$\leqslant \frac{N_k}{n} |\mathcal{S}_i^{(k)}| \overset{①}{\leqslant} \sqrt{\frac{2}{\pi}} \frac{N_k R}{\delta \alpha_k}, \tag{44c}$$

where ① comes from Lemma 1. Therefore, with probability at least $1 - \delta$, we have

$$T_2 \leqslant \sqrt{\frac{2}{\pi}} \frac{N_k R}{\delta \alpha_k} \left\| f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. \tag{45}$$

Combine the results of (43c) and (45):

$$\left\langle \mathbf{d}^{\mathrm{I}}, f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\rangle \leqslant \eta \left(\frac{3\sqrt{2} R}{\sqrt{\pi} \delta \alpha_k} + \sqrt{\frac{\ln{(2N_k^2/\delta)}}{2n}} - \frac{\lambda_k}{N_k} \right) \left\| f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. \tag{46}$$

For the inner product term $\left\langle \mathbf{d}^{\mathrm{II}}, f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\rangle$, we first bound $\|\mathbf{d}^{\mathrm{II}}\|_2^2$ as follows:

$$
\begin{aligned}
\|\mathbf{d}^{\mathrm{II}}\|_2^2 &= \sum_{i=1}^{N_k} \left(\frac{1}{\sqrt{n}} \sum_{r \in \mathcal{S}_i^{(k)}} \left[c_r \sigma \left((\mathbf{v}_r^{(k,t+1)})^\top \mathbf{x}_i \right) - c_r \sigma \left((\mathbf{v}_r^{(k,t)})^\top \mathbf{x}_i \right) \right] \right)^2 && \text{(47a)} \\
&\overset{①}{\leqslant} \frac{\eta^2}{n} \sum_{i=1}^{N_k} |\mathcal{S}_i^{(k)}| \sum_{r \in \mathcal{S}_i^{(k)}} \left(c_r \langle \nabla_{\mathbf{v}_r} L, \mathbf{x}_i \rangle \right)^2 && \text{(47b)} \\
&\overset{②}{\leqslant} \frac{\eta^2}{n} \sum_{i=1}^{N_k} |\mathcal{S}_i^{(k)}| \sum_{r \in \mathcal{S}_i^{(k)}} \|\nabla_{\mathbf{v}_r} L\|_2^2 \, \|\mathbf{x}_i\|_2^2 && \text{(47c)} \\
&\leqslant \frac{\eta^2 N_k}{n} |\mathcal{S}_i^{(k)}|^2 \max_{r \in [n]} \left\| \nabla_{\mathbf{v}_r} L \right\|_2^2 && \text{(47d)} \\
&\leqslant \frac{\eta^2 |\mathcal{S}_i^{(k)}|^2}{n^2} \left\| f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2, && \text{(47e)}
\end{aligned}
$$

where ① comes from the Lipschitz continuity of the ReLU function $\sigma(\cdot)$, ② holds due to Cauchy–Schwartz inequality. Plug (21) into (47e), we have with probability at least $1 - \delta$:

$$
\|\mathbf{d}^{\mathrm{II}}\|_2^2 \leqslant \frac{2\eta^2 R^2}{\pi \delta^2 \alpha_k^2} \left\| f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. \tag{48}
$$

The inner product term $\left\langle \mathbf{d}^{\mathrm{II}}, f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\rangle$ can be bounded as

$$
\left\langle \mathbf{d}^{\mathrm{II}}, f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\rangle \leqslant \frac{\sqrt{2}\eta R}{\sqrt{\pi} \delta \alpha_k} \left\| f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. \tag{49}
$$

Finally, the bound for the difference term is derived as

$$
\left\| f^{(k,t+1)}(\mathbf{X}^{(k)}) - f^{(k,t)}(\mathbf{X}^{(k)}) \right\|_2^2 \leqslant \sum_{i=1}^{N_k} \left(\frac{\eta}{\sqrt{n}} \sum_{r=1}^{n} c_r \langle \nabla_{\mathbf{v}_r} L, \mathbf{x}_i \rangle \right)^2 \leqslant \eta^2 \left\| f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. \tag{50}
$$

Combine the results of (46), (49) and (50):

$$
\left\| f^{(k,t+1)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2 \leqslant \left[1 + \frac{8\sqrt{2}\eta R}{\sqrt{\pi}\delta\alpha_k} + 2\eta \sqrt{\frac{\ln(2N_k^2/\delta)}{2n}} - \frac{2\eta\lambda_k}{N_k} + \eta^2 \right] \left\| f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. \tag{51}
$$

Let $R = O\left(\frac{\delta\alpha_k\lambda_k}{N_k}\right)$, $n = \Omega\left(\frac{N_k^2}{\lambda_k^2} \ln \frac{N_k^2}{\delta}\right)$, and $\eta = O(\frac{\lambda_k}{N_k})$, we have

$$
\left\| f^{(k,t+1)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2 \leqslant \left(1 - \frac{\eta\lambda_k}{2N_k} \right) \left\| f^{(k,t)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. \tag{52}
$$

Invoking the inequality (52) recursively completes the proof. □

E. Proof of Theorem 2

Theorem 2 *For FedAvg under Assumptions 1 to 3, let the learning rate* $\eta = O\left(\frac{\lambda_k}{\tau N_k M_k}\right)$ *and the neural network width* $n = \Omega\left(\frac{N_k^2}{\lambda_k^2} \ln \frac{N_k^2}{\delta}\right)$*, then with probability at least* $1 - \delta$*,*

$$
\left\| f^{(k+1)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2 \leqslant \left(1 - \frac{\eta\tau\lambda_k}{2N_k M_k} \right) \left\| f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2, \tag{53}
$$

where τ *is the number of local iterations, and* M_k *is the number of clients in communication round* k*.*

Proof. We first construct a different set of kernel matrices $\{\mathbf{\Lambda}^{(k)}, \mathbf{\Lambda}_m^{(k,\tau)}\}$ similar to Huang et al. (2021). Let $\mathbb{1}_{imr}^{(k,u)} \triangleq \mathbb{1}\{\langle \mathbf{v}_{m,r}^{(k,u)}, \mathbf{x}_i \rangle \geqslant 0\}$, the (i,j)th entry of $\mathbf{\Lambda}_m^{(k,u)}$ and $\mathbf{\Lambda}^{(k,u)}$ is defined as

$$(\mathbf{\Lambda}_m^{(k,u)})_{ij} \triangleq \frac{1}{n} \mathbf{x}_i^\top \mathbf{x}_j \sum_{r=1}^{n} \mathbb{1}_{imr}^{(k,0)} \mathbb{1}_{jmr}^{(k,u)}, \tag{54a}$$

$$(\mathbf{\Lambda}^{(k,u)})_{ij} \triangleq (\mathbf{\Lambda}_m^{(k,u)})_{ij}, \quad \text{if } (\mathbf{x}_j, y_j) \in \mathcal{D}_m. \tag{54b}$$

Taking the difference between successive terms yields

$$f^{(k+1)}(\mathbf{x}_i) - f^{(k)}(\mathbf{x}_i) = \frac{1}{\sqrt{n}} \sum_{r=1}^{n} \left[c_r \sigma \left((\mathbf{v}_r^{(k+1)})^\top \mathbf{x}_i \right) - c_r \sigma \left((\mathbf{v}_r^{(k)})^\top \mathbf{x}_i \right) \right]. \tag{55}$$

We decompose the difference term to the sum of d_i^{I} and d_i^{II}, based on the set $\mathcal{S}_i^{(k)}$ and its complement:

$$d_i^{\mathrm{I}} \triangleq \frac{1}{\sqrt{n}} \sum_{r \notin \mathcal{S}_i^{(k)}} \left[c_r \sigma \left((\mathbf{v}_r^{(k+1)})^\top \mathbf{x}_i \right) - c_r \sigma \left((\mathbf{v}_r^{(k)})^\top \mathbf{x}_i \right) \right], \tag{56a}$$

$$d_i^{\mathrm{II}} \triangleq \frac{1}{\sqrt{n}} \sum_{r \in \mathcal{S}_i^{(k)}} \left[c_r \sigma \left((\mathbf{v}_r^{(k+1)})^\top \mathbf{x}_i \right) - c_r \sigma \left((\mathbf{v}_r^{(k)})^\top \mathbf{x}_i \right) \right]. \tag{56b}$$

Consider the residual term

$$\left\| f^{(k+1)}(\mathbf{X}^{(k)}) - \mathbf{y} \right\|_2^2 \tag{57a}$$

$$= \left\| f^{(k+1)}(\mathbf{X}^{(k)}) - f^{(k)}(\mathbf{X}^{(k)}) + f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y} \right\|_2^2 \tag{57b}$$

$$= \left\| f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2 + 2 \left\langle \mathbf{d}^{\mathrm{I}} + \mathbf{d}^{\mathrm{II}}, f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y} \right\rangle + \left\| f^{(k+1)}(\mathbf{X}^{(k)}) - f^{(k)}(\mathbf{X}^{(k)}) \right\|_2^2. \tag{57c}$$

We will give upper bounds for the inner product terms $\langle \mathbf{d}^{\mathrm{I}}, f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y} \rangle$, $\langle \mathbf{d}^{\mathrm{II}}, f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y} \rangle$, and the difference term $\left\| f^{(k+1)}(\mathbf{X}^{(k)}) - f^{(k)}(\mathbf{X}^{(k)}) \right\|_2^2$, separately. For an input $\mathbf{x} \in \mathbb{R}^{d_1}$, let $f_m^{(k,u)}(\mathbf{x}) \triangleq \frac{1}{\sqrt{n}} \sum_{r=1}^{n} c_r \sigma(\langle \mathbf{v}_{m,r}^{(k,u)}), \mathbf{x} \rangle)$. By the update rule of FedAvg, the relation between the weight vector $\mathbf{v}_r^{(k)}$ in successive communication rounds is:

$$\mathbf{v}_r^{(k+1)} = \mathbf{v}_r^{(k)} - \frac{\eta}{M_k} \sum_{m \in \mathcal{C}_k} \sum_{u=0}^{\tau-1} \nabla L_{\mathbf{v}_r^{(k,u)}} \tag{58a}$$

$$= \mathbf{v}_r^{(k)} - \frac{\eta c_r}{N_k \sqrt{n} M_k} \sum_{m \in \mathcal{C}_k} \sum_{u=0}^{\tau-1} \sum_{j \in \mathcal{I}_m} (f_m^{(k,u)}(\mathbf{x}_j) - y_j) \mathbf{x}_j \mathbb{1}_{jmr}^{(k,u)}. \tag{58b}$$

Based on the property of the set $\mathcal{S}_i^{(k)}$, we have

$$d_i^{\mathrm{I}} = -\frac{1}{\sqrt{n}} \sum_{m \in \mathcal{C}_k} \sum_{u=0}^{\tau-1} \sum_{r \notin \mathcal{S}_i^{(k)}} c_r \left\langle \mathbf{v}_r^{(k+1)} - \mathbf{v}_r^{(k)}, \mathbf{x}_i \right\rangle \mathbb{1}_{ir}^{(k)} \tag{59a}$$

$$= -\frac{\eta}{N_k n M_k} \sum_{m \in \mathcal{C}_k} \sum_{u=0}^{\tau-1} \sum_{r \notin \mathcal{S}_i^{(k)}} \sum_{j \in \mathcal{I}_m} (f_m^{(k,u)}(\mathbf{x}_j) - y_j) \mathbf{x}_i^\top \mathbf{x}_j \mathbb{1}_{ir}^{(k)} \mathbb{1}_{jmr}^{(k,u)} \tag{59b}$$

$$= -\frac{\eta}{N_k M_k} \sum_{m \in \mathcal{C}_k} \sum_{u=0}^{\tau-1} \sum_{j \in \mathcal{I}_m} (f_m^{(k,u)}(\mathbf{x}_j) - y_j) \left[(\mathbf{\Lambda}_m^{(k,u)})_{ij} - (\mathbf{\Lambda}_m^{\perp(k,u)})_{ij} \right]. \tag{59c}$$

For the inner product term $\langle \mathbf{d}^{\mathrm{I}}, f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y} \rangle$, we have

$$\left\langle \mathbf{d}^{\mathrm{I}}, f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\rangle = -\frac{\eta}{N_k M_k} \sum_{u=0}^{\tau-1} (f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top (\mathbf{\Lambda}^{(k,u)} - \mathbf{\Lambda}^{\perp(k,u)}) (f_m^{(k,u)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}). \tag{60}$$

Let T_1 and T_2 denote the following terms

$$T_1 \triangleq -(f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top \mathbf{\Lambda}^{(k,u)}(f_g^{(k,u)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}), \tag{61a}$$

$$T_2 \triangleq (f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top \mathbf{\Lambda}^{\perp(k,u)}(f_g^{(k,u)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}), \tag{61b}$$

where $f_g^{(k,u)}(\mathbf{X}^{(k)}) \triangleq [f_1^{(k,u)}(\mathbf{X}_1)^\top, \cdots, f_{M_k}^{(k,u)}(\mathbf{X}_{M_k})^\top]^\top$. We are going to bound T_1 and T_2 separately. T_1 can be written as:

$$\begin{aligned} T_1 &= -(f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top (\mathbf{\Lambda}^{(k,u)} - \mathbf{H}^{(k)} + \mathbf{H}^{(k)} - \mathbf{H}_\infty^{(k)} + \mathbf{H}_\infty^{(k)})(f_g^{(k,u)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}) && (62a)\\ &= -(f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top (\mathbf{\Lambda}^{(k,u)} - \mathbf{H}^{(k)})(f_g^{(k,u)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}) \\ &\quad - (f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top (\mathbf{H}^{(k)} - \mathbf{H}_\infty^{(k)})(f_g^{(k,u)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}) \\ &\quad - (f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top \mathbf{H}_\infty^{(k)}(f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}) \\ &\quad - (f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)})^\top \mathbf{H}_\infty^{(k)}(f_g^{(k,u)}(\mathbf{X}^{(k)}) - f^{(k)}(\mathbf{X}^{(k)})). && (62b) \end{aligned}$$

First, we bound the norm of $f_g^{(k,u)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}$. It can be shown that

$$\begin{aligned} \|f_m^{(k,u)}(\mathbf{X}_m) - \mathbf{y}_m\|_2 &= \|f_m^{(k,u)}(\mathbf{X}_m) - f_m^{(k,u-1)}(\mathbf{X}_m) + f_m^{(k,u-1)}(\mathbf{X}_m) - \mathbf{y}_m\|_2 && (63a)\\ &\leqslant \|f_m^{(k,u)}(\mathbf{X}_m) - f_m^{(k,u-1)}(\mathbf{X}_m)\|_2 + \|f_m^{(k,u-1)}(\mathbf{X}_m) - \mathbf{y}_m\|_2 && (63b)\\ &\overset{①}{\leqslant} (1+\eta)\|f_m^{(k,u-1)}(\mathbf{X}_m) - \mathbf{y}_m\|_2, && (63c) \end{aligned}$$

where ① holds based on the derivation of (50). Applying (63c) recursively yields

$$\|f_m^{(k,u)}(\mathbf{X}_m) - \mathbf{y}_m\|_2 \leqslant (1+\eta)^u \|f^{(k)}(\mathbf{X}_m) - \mathbf{y}_m\|_2. \tag{64}$$

The bound for $\|f_g^{(k,u)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\|_2^2$ can thus be derived as

$$\begin{aligned} \|f_g^{(k,u)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\|_2^2 &= \sum_{i=1}^{N_k} \left[f_g^{(k,u)}(\mathbf{x}_i) - y_i\right]^2 && (65a)\\ &= \sum_{m\in\mathcal{C}_k} \left\|f_m^{(k,u)}(\mathbf{X}_m) - \mathbf{y}_m\right\|_2^2 && (65b)\\ &\leqslant (1+\eta)^{2u}\left\|f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\right\|_2^2. && (65c) \end{aligned}$$

Second, following the steps in Lemma 3, it can be shown that with probability at least $1-\delta$,

$$\|\mathbf{\Lambda}^{(k,t)} - \mathbf{H}^{(k)}\|_2 \leqslant \frac{2\sqrt{2}N_k R}{\sqrt{\pi}\delta\alpha}. \tag{66}$$

We also bound the difference between $f_g^{(k,u)}(\mathbf{X}^{(k)})$ and $f^{(k)}(\mathbf{X}^{(k)})$ as follows:

$$\begin{aligned} \|f_g^{(k,u)}(\mathbf{X}^{(k)}) - f^{(k)}(\mathbf{X}^{(k)})\|_2 &\overset{①}{\leqslant} \sum_{v=1}^{u} \|f_g^{(k,v)}(\mathbf{X}^{(k)}) - f_g^{(k,v-1)}(\mathbf{X}^{(k)})\|_2 && (67a)\\ &\overset{②}{\leqslant} \sum_{v=1}^{u} \eta \|f_g^{(k,v-1)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\|_2 && (67b)\\ &\overset{③}{\leqslant} \sum_{v=1}^{u} \eta(1+\eta)^{v-1} \|f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\|_2 && (67c)\\ &= [(1+\eta)^u - 1]\,\|f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\|_2, && (67d) \end{aligned}$$

where ① holds due to triangle inequality, ② comes from (50), ③ comes from (65c). Plugging the results from (65c), (66), and (67d) into (62b), we have with probability at least $1-\delta$,

$$T_1 \leqslant \left[(1+\eta)^u \left(\frac{2\sqrt{2}N_k R}{\sqrt{\pi}\delta\alpha} + N_k\sqrt{\frac{\ln(2N_k^2/\delta)}{2n}} + \kappa\lambda_k\right) - (1+\kappa)\lambda_k\right] \|f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)}\|_2^2, \tag{68}$$

where κ is the condition number of the matrix $\mathbf{H}_{\infty}^{(k)}$. Next, consider the bound for T_2. The ℓ_2 norm of $\mathbf{\Lambda}^{\perp(k,u)}$ can be bounded as

$$
\begin{aligned}
\|\mathbf{\Lambda}^{\perp(k,u)}\|_2 &\leqslant \|\mathbf{\Lambda}^{\perp(k,u)}\|_{\mathrm{F}} && \text{(69a)}\\
&= \left(\sum_{i=1}^{N_k} \sum_{m\in\mathcal{C}_k} \sum_{j\in\mathcal{I}_m} \left(\frac{1}{n} \sum_{r\in\mathcal{S}_i^{(k)}} \mathbf{x}_i^\top \mathbf{x}_j \mathbb{1}_{ir}^{(k)} \mathbb{1}_{jmr}^{(k,u)} \right)^2 \right)^{\frac{1}{2}} && \text{(69b)}\\
&\leqslant \frac{N_k}{n}|\mathcal{S}_i^{(k)}| \overset{\textcircled{1}}{\leqslant} \sqrt{\frac{2}{\pi}}\frac{N_k R}{\delta\alpha}, && \text{(69c)}
\end{aligned}
$$

where ① comes from Lemma 1. Therefore, we have with probability at least $1-\delta$,

$$
T_2 \leqslant (1+\eta)^u \sqrt{\frac{2}{\pi}}\frac{N_k R}{\delta\alpha}\left\| f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. \tag{70}
$$

Combine the results of (68) and (70):

$$
\begin{aligned}
\left\langle \mathbf{d}^{\mathrm{I}}, f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\rangle \leqslant \frac{\tau}{M_k}\Bigg[&\left(\eta + \frac{(\tau-1)}{2}\eta^2 + o(\eta^2) \right) \left(\frac{3\sqrt{2}R}{\sqrt{\pi}\delta\alpha} + \sqrt{\frac{\ln\left(\frac{2N_k^2}{\delta}\right)}{2n}} + \frac{\kappa\lambda_k}{N_k} \right) \\
&- \frac{(1+\kappa)\eta\lambda_k}{N_k} \Bigg] \left\| f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2.
\end{aligned} \tag{71}
$$

For the inner product term $\left\langle \mathbf{d}^{\mathrm{II}}, f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y} \right\rangle$, we first bound $\|\mathbf{d}^{\mathrm{II}}\|_2^2$ with probability at least $1-\delta$:

$$
\begin{aligned}
\|\mathbf{d}^{\mathrm{II}}\|_2^2 &= \sum_{i=1}^{N_k} \left(\frac{1}{\sqrt{n}} \sum_{r\in\mathcal{S}_i^{(k)}} \left[c_r \sigma\left((\mathbf{v}_r^{(k+1)})^\top \mathbf{x}_i \right) - c_r \sigma\left((\mathbf{v}_r^{(k)})^\top \mathbf{x}_i \right) \right] \right)^2 && \text{(72a)}\\
&\leqslant \frac{1}{n} \sum_{i=1}^{N_k} |\mathcal{S}_i^{(k)}| \sum_{r\in\mathcal{S}_i^{(k)}} \left(c_r \langle \mathbf{v}_r^{(k+1)} - \mathbf{v}_r^{(k)}, \mathbf{x}_i \rangle \right)^2 && \text{(72b)}\\
&\leqslant \frac{1}{n} \sum_{i=1}^{N_k} |\mathcal{S}_i^{(k)}| \sum_{r\in\mathcal{S}_i^{(k)}} \left(\frac{\eta c_r}{N_k\sqrt{n}M_k} \sum_{m\in\mathcal{C}_k} \sum_{u=0}^{\tau-1} \sum_{j\in\mathcal{I}_m} (f_m^{(k,u)}(\mathbf{x}_j) - y_j) \mathbb{1}_{jmr}^{(k,u)} \right)^2 && \text{(72c)}\\
&\leqslant \frac{\eta^2}{N_k^2 n^2 M_k^2} \sum_{i=1}^{N_k} |\mathcal{S}_i^{(k)}| \sum_{r\in\mathcal{S}_i^{(k)}} \left(\sum_{m\in\mathcal{C}_k} \sum_{u=0}^{\tau-1} \sum_{j\in\mathcal{I}_m} \left| f_m^{(k,u)}(\mathbf{x}_j) - y_j \right| \right)^2 && \text{(72d)}\\
&\leqslant \frac{\eta^2}{N_k^2 n^2 M_k^2} \sum_{i=1}^{N_k} |\mathcal{S}_i^{(k)}| \sum_{r\in\mathcal{S}_i^{(k)}} \left(\sum_{m\in\mathcal{C}_k} \sum_{u=0}^{\tau-1} |\mathcal{I}_m| \left\| f_m^{(k,u)}(\mathbf{X}_m) - \mathbf{y}_m \right\|_2 \right)^2. && \text{(72e)}
\end{aligned}
$$

By apply the results from previous steps, we have

$$
\begin{aligned}
\|\mathbf{d}^{\mathrm{II}}\|_2^2 &\overset{\text{①}}{\leqslant} \frac{\eta^2}{N_k^2 n^2 M_k^2} \sum_{i=1}^{N_k} |\mathcal{S}_i^{(k)}| \sum_{r \in \mathcal{S}_i^{(k)}} \left(\sum_{m \in \mathcal{C}_k} \sum_{u=0}^{\tau-1} (1+\eta)^u |\mathcal{I}_m| \left\| f^{(k)}(\mathbf{X}_m) - \mathbf{y}_m \right\|_2 \right)^2 && \text{(73a)} \\
&\overset{\text{②}}{\leqslant} \frac{1}{N_k^2 n^2 M_k^2} \sum_{i=1}^{N_k} |\mathcal{S}_i^{(k)}| \sum_{r \in \mathcal{S}_i^{(k)}} \left(\sum_{m \in \mathcal{C}_k} \left((1+\eta)^\tau - 1\right) |\mathcal{I}_m| \left\| f^{(k)}(\mathbf{X}_m) - \mathbf{y}_m \right\|_1 \right)^2 && \text{(73b)} \\
&\overset{\text{③}}{\leqslant} \frac{1}{N_k n^2 M_k^2} \sum_{i=1}^{N_k} |\mathcal{S}_i^{(k)}| \sum_{r \in \mathcal{S}_i^{(k)}} \left(\left((1+\eta)^\tau - 1\right) \left\| f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2 \right)^2 && \text{(73c)} \\
&\overset{\text{④}}{\leqslant} \frac{2R^2}{\pi \delta^2 \alpha^2 M_k^2} \left(\tau\eta + \frac{\tau(\tau-1)}{2}\eta^2 + o(\eta^2) \right)^2 \left\| f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. && \text{(73d)}
\end{aligned}
$$

where ① comes from (64), ② holds due to $\|\mathbf{a}\|_1 \leqslant \|\mathbf{a}\|_2$, ③ holds due to $\|\mathbf{a}\|_1 \leqslant \sqrt{\dim(\mathbf{a})}\|\mathbf{a}\|_2$, ④ is from Lemma 1. With probability at least $1-\delta$, the inner product term can thus be bounded as

$$
\left\langle \mathbf{d}^{\mathrm{II}}, f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y} \right\rangle \leqslant \frac{\sqrt{2}\tau R}{\sqrt{\pi}\delta\alpha M_k} \left(\eta + \frac{(\tau-1)}{2}\eta^2 + o(\eta^2) \right) \left\| f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. \tag{74}
$$

The bound for the difference term is derived as

$$
\begin{aligned}
\left\| f^{(k+1)}(\mathbf{X}^{(k)}) - f^{(k)}(\mathbf{X}^{(k)}) \right\|_2^2 &\leqslant \sum_{i=1}^{N_k} \left(\frac{\eta}{\sqrt{n}} \sum_{r=1}^{n} c_r \langle \mathbf{v}_r^{(k+1)} - \mathbf{v}_r^{(k)}, \mathbf{x}_i \rangle \right)^2 && \text{(75a)} \\
&\leqslant \frac{1}{M_k^2} \left(\tau\eta + \frac{\tau(\tau-1)}{2}\eta^2 + o(\eta^2) \right)^2 \left\| f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. && \text{(75b)}
\end{aligned}
$$

Combine the results of (71), (74) and (75b):

$$
\begin{aligned}
\left\| f^{(k+1)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2 \leqslant \Bigg\{ 1 + \frac{2\eta\tau}{M_k} \Bigg[\left(\frac{4\sqrt{2}R}{\sqrt{\pi}\delta\alpha} + \sqrt{\frac{\ln\left(\frac{2N_k^2}{\delta}\right)}{2n}} + \frac{\kappa\lambda_k}{N_k} \right) \\
- \frac{(1+\kappa)\lambda_k}{N_k} \Bigg] + \frac{\eta^2\tau^2}{M_k^2} + o(\eta^2) \Bigg\} \left\| f^{(k)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2.
\end{aligned} \tag{76}
$$

Let $R = O\left(\frac{\delta\alpha\lambda_k}{N_k}\right)$, $n = \Omega\left(\frac{N_k^2}{\lambda_k^2} \ln \frac{N_k^2}{\delta}\right)$, and $\eta = O\left(\frac{\lambda_k}{\tau N_k M_k}\right)$, we have

$$
\left\| f^{(k+1)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2 \leqslant \left(1 - \frac{\eta\tau\lambda_k}{2N_k M_k} \right) \left\| f^{(k+1)}(\mathbf{X}^{(k)}) - \mathbf{y}^{(k)} \right\|_2^2. \tag{77a}
$$

□