APPENDIX A NOTATION

Table 5: Major notations occurred in the paper.

Notations	Dimension	Description
k, K	$\mathbb{N}$	number and total number of local update steps
c, N	$\mathbb{N}$	client index, total number of clients
n_c, n	$\mathbb{N}$	number of samples of client c and overall
m	$\mathbb{N}$	number of hidden neurons
η_l, η_g	$\mathbb{R}$	client-level learning rate, global learning rate
w_c, u	$\mathbb{R}^p$	local model of client c, global model
(x_i, y_i)	$\mathbb{R}^m$	sample pair of index i
$y_c(t), y(t)$	$\mathbb{R}$	model prediction at time t with local model, global model
S_c	set of size n_c	collection of samples of client c

APPENDIX B GLOBAL CONVERGENCE

B.1 FORMULATION

B.1.1 THE NEURAL NETWORK

For the following theoretical analysis, we follow Du et al. (2019)Song & Yang (2020)Huang et al. (2021) to consider an one-hidden-layer neural network with ReLU activation:

$$f(u, x) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \phi(u_r^T x)$$

where m is the total number of neurons in the hidden layer, a_r's are Rademacher random variables (take values $\{\pm 1\}$ with equal probability) and ϕ is the activation function. Each client aim to optimize its MSE loss:

$$L_c^{mse}(u, x) = \frac{1}{2} \sum_{i \in S_c} \left(f(u, x_i) - y_i\right)^2$$

where S_c represents the collection of data of client c. The global loss is taken as the average of the loss of each client:

$$L(u, x) = \frac{1}{N} \sum_{c \in [N]} L_c(u, x)$$

B.1.2 OUR ALGORITHM

We first formulate our proposed algorithm mathematically to establish consistent notations and facilitate analysis.

In FL, each client alternate between performing local updates on its local data and communicating with the central server for global aggregation. Each client takes K local updates between each communication round. The local update is performed using vanilla gradient descent with a local learning rate η_l, and $w_c(t, k)$ represents the weight parameters of client c at global round t and local step k:

$$w_c(t, k+1) \leftarrow w_c(t, k) - \eta_l \frac{\partial L_c(w_c(t, k))}{\partial w_c(t, k)}$$

After each communication, the global aggregation procedure is conducted by taking the average of local updates of all N clients, and a learning rate of η_g is added for the global update:

$$\Delta u(t) = \frac{\eta_g}{N} \sum_{c \in [N]} \Delta w_c(t)$$

where $\Delta w_c(t) = w_c(t, K) - w_c(t, 0) = -\sum_k \eta_l \frac{\partial L_c(w_c(t,k))}{\partial w_c(t,k)}$ is the cumulative local updates of client c at global round t. For the local step, a combination of local update and global update is taken, with $\lambda(t) \in [0, 1]$ being the combination factor:

$$w_c(t+1, 0) \leftarrow w_c(t, 0) + (1 - \lambda(t))\Delta u(t) + \lambda(t)\Delta w_c(t)$$

B.1.3 GRADIENT UPDATES

With the above setting, we can explicitly write out the gradient updates:

$$\Delta w_{c,r}(t) = -\sum_{k\in[K]} \eta_l \frac{\partial L_c^{mse}(w_{c,r}(t,k))}{\partial w_{c,r}(t,k)} = -\frac{\eta_l}{\sqrt{m}} \sum_{k\in[K]} \sum_{i\in S_c} [f(w_{c,r}, x_i) - y_i] a_r x_i \mathbb{1}_{w_{c,r}^T x_i \geq 0}$$

$$\Delta u_r(t) = -\frac{\eta_l \eta_g}{N\sqrt{m}} \sum_{c\in[N]} \sum_{k\in[K]} \sum_{i\in S_c} [f(w_{c,r}(t,k), x_i) - y_i] a_r x_i \mathbb{1}_{w_{c,r}^T x_i \geq 0}$$

B.2 CONVERGENCE ANALYSIS

We analyze the convergence behavior of all clients collectively. That is, we consider the dynamics of $\|y - y(t)\|^2 = \sum_c \|y_c - y_c(t)\|^2$, where

$$y(t) = (f(w_1(t), x_1), ..., f(w_1(t), x_{n_1}), f(w_2(t), x_1), ..., f(w_N(t), x_{n_N}))^T$$
$$y_c(t) = (0, ..., 0, \underbrace{f(w_c(t), x_1), ..., f(w_c(t), x_{n_c})}_{x_i \in S_c}, 0, ..., 0)^T$$

are the stacked vector of predictions and y, y_c are the corresponding ground truth.

Note first the following recurrence relation (†):

$$\begin{aligned}
\|y - y(t+1)\|^2 &= \|[y - y(t)] - [y(t+1) - y(t)]\|^2 \\
&= \|y - y(t)\|^2 - 2(y - y(t))^T (y(t+1) - y(t)) + \|y(t+1) - y(t)\|^2 \\
&= \|y - y(t)\|^2 - 2\underbrace{\sum_c (y_c - y_c(t))^T (y_c(t+1) - y_c(t))}_{\text{the cross term}} + \|y(t+1) - y(t)\|^2
\end{aligned}$$

We will express $\|y - y(t+1)\|^2$ in terms of $\|y - y(t)\|^2$ with a shrinking factor, by bounding each of these terms, and hence prove the convergence of the algorithm.

B.2.1 THE CROSS TERM

We first investigate the cross term. Note that the difficulty in the analysis mainly comes from the non-linear activation pattern. However, this is overcame by a key observation in classical NTK theoryDu et al. (2019)Huang et al. (2021) that the activation patterns stay the same for most of the neurons.

We follow their approaches to define

$$Q_i := \{r \in [m] : \forall m \in \mathbb{R}^d s.t. \|w - w_r(0)\|_2 \leq R, \mathbb{1}_{w_r(0)^T x_i \geq 0} = \mathbb{1}_{w^T x_i \geq 0}\}$$

which represent the set of neurons whose activation pattern does not change during training for sample x_i, and let $\bar{Q}_i$ denote its complement. Then for each sample $i \in S_c$,

$$\begin{aligned}
y_i(t+1) - y_i(t) &= \frac{1}{\sqrt{m}} \sum_{r\in[m]} a_r \Big[\phi(w_r^T(t+1)x_i) - \phi(w_r^T(t)x_i)\Big] \\
&= \underbrace{\frac{1}{\sqrt{m}} \sum_{r\in Q_i} a_r (1 - \lambda_c(t)) \Delta u_r^T(t) x_i \mathbb{1}_{w_r^T(t)x_i \geq 0}}_{v_{1,i}} \\
&\quad + \underbrace{\frac{1}{\sqrt{m}} \sum_{r\in Q_i} a_r \lambda_c(t) \Delta w_{c,r}^T(t) x_i \mathbb{1}_{w_r^T(t)x_i \geq 0}}_{v_{2,i}} + v_{3,i}
\end{aligned}$$

where

$$v_{1,i} = -\frac{(1-\lambda_c(t))\eta_l\eta_g}{Nm} \sum_{k\in[K],r\in Q_i} \sum_{j\in S'_c, c'\in[N]} (y(t,k)_j - y_j) x_i^T x_j \mathbb{1}_{w_{c',r}^T(t,k)x_j\geq 0, w_{c,r}^T(t)x_i\geq 0}$$

$$v_{2,i} = -\frac{\lambda_c(t)\eta_l}{m} \sum_{k\in[K],r\in Q_i} \sum_{j\in S_c} (y_c(t,k)_j - y_{c,j}) x_i^T x_j \mathbb{1}_{w_{c,r}^T(t,k)x_j\geq 0, w_r^T(t)x_i\geq 0}$$

$$v_{3,i} = \frac{1}{\sqrt{m}} \sum_{r\notin Q_i} a_r \Big[\phi(w_r^T(t+1)x_i) - \phi(w_r^T(t)x_i)\Big]$$

We can already notice the almost symmetric kernel factor in the terms above. We give the formal definitions here.

Definition 1 (Global Gram matrix). *For $t \in [T], k \in [K], c, c' \in [N], i \in S_c$ and $j \in S_{c'}$, we define the global gram matrix as:*

$$H(t,k)_{i,j} := \frac{1}{m} \sum_{r\in[m]} x_i^T x_j \mathbb{1}_{w_{c,r}^T(t)x_i\geq 0, w_{c',r}^T(t,k)x_j\geq 0} \qquad \in \mathbb{R}^{n\times n}$$

$$H(t,k)^{\perp}_{i,j} := \frac{1}{m} \sum_{r\notin Q_i} x_i^T x_j \mathbb{1}_{w_{c,r}^T(t)x_i\geq 0, w_{c',r}^T(t,k)x_j\geq 0} \qquad \in \mathbb{R}^{n\times n}$$

Note that this definition is similar to, but not exactly the same as, the definition in FL-NTK Huang et al. (2021). This is because they considered the vanilla FedAvg with no personalization of model parameters.

Definition 2 (Local Gram matrix). *For $t \in [T], k \in [K], c \in [N]$ and $i, j \in S_{c'}$, we define the local gram matrix as:*

$$H_c(t,k)_{i,j} = \frac{1}{m} \sum_{r} x_i^T x_j \mathbb{1}_{w_{c,r}^T(t)x_i\geq 0, w_{c,r}^T(t,k)x_j\geq 0} \qquad \in \mathbb{R}^{n_c\times n_c}$$

$$H_c(t,k)^{\perp}_{i,j} = \frac{1}{m} \sum_{r\notin Q_i} x_i^T x_j \mathbb{1}_{w_{c,r}^T(t)x_i\geq 0, w_{c,r}^T(t,k)x_j\geq 0} \qquad \in \mathbb{R}^{n_c\times n_c}$$

However, in order to maintain consistent dimensions and correspond to our definition of y_c, we can extend the dimension of H_c to $n \times n$ by adding zeros to the undefined entries, i.e.,

$$\begin{pmatrix} 0 & \cdots & 0 \\ \vdots & H_c & \vdots \\ 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{n\times n}$$

From now on, the symbol H_c will refer to this $n \times n$ matrix. Note that $(H_c)_{i,j} = H_{i,j}\mathbb{1}_{i,j\in S_c}$.

We will show that the convergence can be governed by the spectral property of these Gram matrices. Substitute them into the cross term, we get:

$$\begin{aligned}
&\sum_c (y_c - y_c(t))^T (y_c(t+1) - y_c(t)))\\
=&\sum_c \frac{(1-\lambda_c)\eta_l\eta_g}{N} \sum_{i\in S_c} (y_{c,i} - y_c(t)_i) \sum_{k\in[K],j\in[n]} (y(t,k)_j - y_j)(H(t,k)_{i,j} - H(t,k)^{\perp}_{i,j})\\
&\quad + \sum_c \lambda_c\eta_l \sum_{i\in S_c} (y_{c,i} - y_c(t)_i) \sum_{k\in[K],j\in S_c} (y_c(t,k)_j - y_{c,j})(H_c(t,k)_{i,j} - H_c(t,k)^{\perp}_{i,j})\\
&\quad - \sum_c \sum_{i\in S_c} (y_{c,i} - y_c(t)_i) v_{3,i}
\end{aligned}$$

Let

$$C_1 := -\sum_c \frac{(1-\lambda_c(t))\eta_l\eta_g}{N} \sum_{i\in S_c}(y_{c,i}-y_c(t)_i)\sum_{k,j}(y(t,k)_j - y_j)(H(t,k)_{i,j} - H(t,k)^{\perp}_{i,j})$$

$$C_2^{(c)} := -\lambda_c(t)\eta_l \sum_{i\in S_c}(y_{c,i}-y_c(t)_i)\sum_{k,j\in S_c}(y_c(t,k)_j - y_{c,j})(H_c(t,k)_{i,j} - H_c(t,k)^{\perp}_{i,j})$$

$$C_2 := \sum_c C_2^{(c)}$$

$$C_3 := -\sum_c \sum_{i\in S_c}(y_{c,i}-y_c(t)_i)v_{3,i}$$

Then by substituting them back into the recursive relation (†), we get:

$$\|y-y(t+1)\|^2 = \|y-y(t)\|^2 + 2(C_1+C_2+C_3) + \|y(t+1)-y(t)\|^2$$

We will bound each of these terms and hence prove the result.

B.3 Convergence analysis - Main Theorem

We first restate the main convergence theorems.

Theorem 1. *For uniform $\lambda_c(t) = \lambda(t), \forall c$, for $m = \Omega(\lambda^{-4}n^4\log(n/\delta))$, randomly initialized parameters (i.e. $w(0) \sim \mathcal{N}(0,I)$), and $\eta_l = \mathcal{O}(\lambda/\kappa K n^2)$, $\eta_g = \mathcal{O}(1)$, then with probability at least $1-\delta$ over the random initialization, we have for $\forall t$:*

$$\|y-y(t+1)\|^2 \le \|y-y(t)\|^2 - \zeta\eta_g(1-\lambda(t))s_{min}^{(H)}\|y-y(t)\|^2 - \zeta\sum_c \lambda(t)s_{min}^{(H_c)}\|y_c-y_c(t)\|^2$$

where $\zeta := \frac{\eta_l K}{2N}$.

Theorem 2. *For non-uniform $\lambda_c(t)$, let $\lambda_{min}(t) := \min_c \lambda_c(t)$ and $\lambda_{max}(t) := \max_c \lambda_c(t)$, For $m = \Omega(\lambda^{-4}n^4\log(n/\delta))$, randomly initialized parameters (i.e. $w(0) \sim \mathcal{N}(0,I)$), and $\eta_l = \mathcal{O}(\lambda/\kappa K n^2)$, $\eta_g = \mathcal{O}(1)$, then with probability at least $1-\delta$ over the random initialization, we have for $\forall t$:*

$$\|y-y(t+1)\|^2 \le \|y-y(t)\|^2 - \zeta\eta_g(1-\lambda_{min}(t))s_{min}^{(H)}K\|y-y(t)\|^2 - \zeta\sum_c \lambda_{max}(t)s_{min}^{(H_c)}\|y_c-y_c(t)\|^2$$

where $\zeta := \frac{\eta_l K}{2N}$.

We will give the proof of theorem 1 in the subsequent sections, and we note that theorem 2 is a natural extension of theorem 1 so the proof also naturally extends.

B.4 Useful Lemmas

Before giving the proof of the theorem, we state two useful lemmas.

The first lemma gives bounds on the norm of the local and global updates.

Lemma 1. *With $\|x_i\|_2 = 1$, we have*

$$\|\Delta u_r(t)\|_2 \le \frac{2\eta_l\eta_g K(1+2\eta_l nK)\sqrt{n}}{N\sqrt{m}}\|y-y(t)\|_2$$

$$\|\Delta w_r^{(c)}(t)\|_2 \le \frac{2\eta_l K(1+2\eta_l n_c K)\sqrt{n_c}}{\sqrt{m}}\|y_c-y_c(t)\|_2$$

Proof. The first inequality follows from FL-NTK. For the second inequality, consider:

$$\begin{aligned}\|\Delta w_r^{(c)}(t)\|_2 &= \eta_l \Big\| \frac{a_r}{\sqrt{m}} \sum_{k\in[K]} \sum_{i\in S_c} [y(t,k)_i - y_i] x_i \mathbb{1}_{w_{k,c}^T x_i \geq 0} \Big\| \\ &\leq \frac{\eta_l}{\sqrt{m}} \sum_{k\in[K]} \sum_{i\in S_c} |y_i - y(t,k)_i| \\ &\leq \frac{\eta_l \sqrt{n_c}}{\sqrt{m}} \sum_{k\in[K]} \|y_c - y_c(t,k)\| \\ &\leq \frac{\eta_l K(1+2\eta_l n_c K)\sqrt{n_c}}{\sqrt{m}} \|y_c - y_c(t)\|_2\end{aligned}$$

□

The second lemma bounds the sum of client prediction error by that of the global error.

Lemma 2.

$$\sum_c \|y_c - y_c(t)\| \leq \sqrt{N}\|y - y(t)\|$$

Proof. By Jensen's inequality, and since the square root function is concave,

$$\sqrt{\frac{1}{N}\sum_c \|y_c - y_c(t)\|^2} \geq \frac{1}{N}\sum_c \|y_c - y_c(t)\|$$

□

B.5 Proof of Theorem 1

We provide here a detailed proof of theorem 1, and we note that the same proof naturally extends to prove theorem 2. We will use λ to represent $\lambda(t)$ for ease of notation.

Firstly, here are two results that directly follow from FL-NTK. They provide bounds on the effect of global and local updates respectively.

Proposition 2. *With probability at least* $1 - n\exp(-mR)$ *over random initialization, we have*

$$\begin{aligned}C_1 \leq \frac{\eta_l \eta_g (1-\lambda)}{N} \|y - y(t)\|^2 (-K s_{min}^{(H)} + 40\sqrt{n}RK(1 + 2\eta_l K\sqrt{n}) + 2\eta_l s_{max}^{(H)} K^2 \sqrt{n})) \\ + \frac{8\eta_g \eta_l (1-\lambda)}{N} K(1 + 2\eta_l nK) nR \|y - y(t)\|^2\end{aligned}$$

Proposition 3. *With probability at least* $1 - n\exp(-mR)$ *over random initialization, we have*

$$\begin{aligned}C_2^{(c)} \leq \frac{\lambda \eta_l}{N} \|y_c - y_c(t)\|^2 (-K s_{min}^{(H_c)} + 40\sqrt{n}RK(1 + 2\eta_l K\sqrt{n} + 2\eta_l s_{max}^{(H_c)} K^2 \sqrt{n})) \\ + \frac{8\lambda \eta_l}{N} K(1 + 2\eta_l nK) nR \|y_c - y_c(t)\|^2\end{aligned}$$

For the following two propositions, we assume that all clients possess the same number of samples, i.e., $n_c = n/N, \forall c$. Additionally, let $\tilde{\eta}_g$ denote $\max\{1, \eta_g\}$.

The following proposition aims to bound the effect of updates on neurons whose activation pattern changed during the algorithm.

Proposition 4. *With probability at least* $1 - n\exp(-mR)$ *over random initialization, we have*

$$C_3 \leq \frac{8\eta_l \tilde{\eta}_g K}{N}(1 + 2\eta_l nK) nR \|y - y(t)\|^2$$

Proof. Consider

$$\|v_3\|_2^2 \leq \underbrace{\frac{1-\lambda}{m}\sum_{i\in[n]}\Big(\sum_{r\in\bar{Q}_i}|\Delta u_r(t)^T x_i|\Big)^2}_{A} + \underbrace{\frac{\lambda}{m}\sum_{i\in[n]}\Big(\sum_{r\in\bar{Q}_i}|\Delta w_r(t)^T x_i|\Big)^2}_{B}$$

and

$$A \leq \Big(\frac{8(1-\lambda)\eta_g\eta_l K}{N}(1+2\eta_l nK)nR\|y-y(t)\|\Big)^2$$

As for B,

$$\begin{aligned}
B &= \frac{\lambda}{m}\sum_c\sum_{i\in S_c}\Big(\sum_{r\in[m]}\mathbb{1}_{r\in\bar{Q}_i}|\Delta u_r(t)^T x_i|\Big)^2 \\
&\leq \frac{\lambda\eta_l^2}{m}\sum_c\frac{4K^2(1+2\eta_l n_c K)^2 n_c}{m}\|y_c-y_c(t)\|^2\cdot n_c(4mR)^2 \\
&\leq \Big(\frac{8\lambda\eta_l K}{N}(1+2\eta_l nK)nR\|y-y(t)\|\Big)^2
\end{aligned}$$

where for the second inequality we used the assumption made above. Then

$$\begin{aligned}
C_3 &:= -\sum_{i\in[n]}(y_i-y_i(t))v_{3,i} \\
&\leq \|y-y(t)\|_2\|v_3\|_2 \\
&\leq \frac{8\eta_l\tilde{\eta}_g K}{N}(1+2\eta_l nK)nR\|y-y(t)\|^2
\end{aligned}$$

□

Now we have bounded the cross term. For the last term, we have the following inequality:

Proposition 5. *We have*

$$\|y(t+1)-y(t)\|^2 \leq \frac{4\eta_l^2\tilde{\eta}_g^2 n^2K^2(1+2\eta_l nK)^2}{N^2}\|y-y(t)\|^2$$

Proof.

$$\begin{aligned}
\|y(t+1)-y(t)\|^2 &\leq \frac{1-\lambda}{m}\sum_{i\in[n]}\Big(\sum_{r\in[m]}|\Delta u_r(t)^T x_i|\Big)^2 + \frac{\lambda}{m}\sum_{i\in[n]}\Big(\sum_{r\in[m]}|\Delta w_r(t)^T x_i|\Big)^2 \\
&\leq \frac{(1-\lambda)\eta_g^2\eta_l^2}{m}\Big(\frac{2K(1+2\eta_l nK)\sqrt{n}}{N\sqrt{m}}\|y-y(t)\|\Big)^2\cdot nm^2 \\
&\qquad + \sum_c\frac{\lambda\eta_l^2}{m}\Big(\frac{2K(1+2\eta_l n_c K)\sqrt{n_c}}{\sqrt{m}}\|y_c-y_c(t)\|\Big)^2\cdot n_c m^2 \\
&\leq \frac{4\eta_l^2\tilde{\eta}_g^2 n^2K^2(1+2\eta_l nK)^2}{N^2}\|y-y(t)\|^2
\end{aligned}$$

where we have used the assumption that $n_c = n/N$. □

Now by substituting the above results to the recursion equation, we get:

$$\begin{aligned}\|y-y(t+1)\|^2 \leq \|y-y(t)\|^2 \\ &+\frac{2\eta_l\eta_g(1-\lambda)}{N}\|y-y(t)\|^2(-Ks_{min}^{(H)}+40\sqrt{n}RK(1+2\eta_lK\sqrt{n}) \\ &+2\eta_l s_{max}^{(H)}K^2\sqrt{n}))+\frac{16\eta_g\eta_l(1-\lambda)}{N}K(1+2\eta_lnK)nR\|y-y(t)\|^2 \\ &+\sum_c\frac{2\lambda\eta_l}{N}\|y_c-y_c(t)\|^2(-Ks_{min}^{(H_c)}+40\sqrt{n}RK(1+2\eta_lK\sqrt{n} \\ &+2\eta_l s_{max}^{(H_c)}K^2\sqrt{n}))+\frac{16\lambda\eta_l}{N}K(1+2\eta_lnK)nR\sum_c\|y_c-y_c(t)\|^2 \\ &+\frac{16\eta_l\tilde{\eta}_gK}{N}(1+2\eta_lnK)nR\|y-y(t)\|^2 \\ &+\frac{4\eta_l^2\tilde{\eta}_g^2n^2K^2(1+2\eta_lnK)^2}{N^2}\|y-y(t)\|^2\end{aligned}$$

Then by the choice of $\eta_l \leq \min\{\frac{s_{min}^{(H)}}{1000\kappa n^2K}, \min_c\{\frac{s_{min}^{(H_c)}}{1000\kappa_c n^2K}\}\}$ where $\kappa := s_{max}/s_{min}$ and $\eta_l\eta_g \leq \min\{\frac{s_{min}^{(H)}}{1000\kappa n^2K}, \min_c\{\frac{s_{min}^{(H_c)}}{1000\kappa_c n^2K}\}\}$ and $R \leq s_{min}^{(H)}/(1000n)$, we have

$$\begin{aligned}\|y-y(t+1)\|^2 \leq \|y-y(t)\|^2 \\ &-\frac{(1-\lambda)\eta_l\eta_g s_{min}^{(H)}K}{N}\|y-y(t)\|^2-\sum_c\frac{\lambda\eta_l s_{min}^{(H_c)}K}{N}\|y_c-y_c(t)\|^2 \\ &+40\frac{\eta_l\eta_gKnR}{N}\|y-y(t)\|^2\times 2 \\ &+\frac{\eta_l^2\tilde{\eta}_g^2n^2K^2}{N^2}\|y-y(t)\|^2 \\ \leq \|y-y(t)\|^2-\frac{(1-\lambda(t))\eta_l\eta_g s_{min}^{(H)}K}{2N}\|y-y(t)\|^2 \\ &-\sum_c\frac{\lambda(t)\eta_l s_{min}^{(H_c)}K}{2N}\|y_c-y_c(t)\|^2\end{aligned}$$

by substituting in the condition on η_l and R.

Quod erat demonstrandum.

APPENDIX C GENERALIZATION

In this section, we prove the generalization bounds. That is, we aim to find a bound on

$$\mathcal{L}_{\mathcal{D}}(f) := \mathbb{E}_{(x,y)\sim D}[l(f(x),y)]$$

where f refer to the prediction function we consider. Note that, in practice, this is approximated by the empirical loss $L_S(f) = \frac{1}{n}\sum_{i\in[n]} l(f(x_i),y_i)$. We also consider a more general initialization scheme $w_r \sim \mathcal{N}(0,\sigma^2 I)$.

C.1 SETUP

We follow Arora et al. (2019b); Huang et al. (2021) to consider a non-degenerate data distribution.

Definition 3 (Non-degenerate Data Distribution)**.** *A distribution $\mathcal{D}$ over $\mathbb{R}^b \times \mathbb{R}$ is (λ,δ,n)-non-degenerate, if with probability at least $1-\delta$, for n iid samples $\{(x_i,y_i)\}_{i=1}^n$ chosen from $\mathcal{D}$, $s_{\min}^{(H^\infty)} \geq s > 0$.*

We also state here the definition of the dynamic matrices which can be used to describe the evolution of the neural network:

Definition 4 (Global Trajectory Matrix)**.**

$$J(t,k) = \frac{1}{\sqrt{m}} \begin{pmatrix} a_1 x_1 \mathbb{1}_{w_{c_{1,1}}^T(t,k)x_1 \geq 0} & \cdots & a_1 x_n \mathbb{1}_{w_{c_{n,1}}^T(t,k)x_n \geq 0} \\ \vdots & \ddots & \vdots \\ a_m x_1 \mathbb{1}_{w_{c_{1,m}}^T(t,k)x_1 \geq 0} & \cdots & a_m x_n \mathbb{1}_{w_{c_{n,m}}^T(t,k)x_n \geq 0} \end{pmatrix} \in \mathbb{R}^{md \times n}$$

Definition 5 (Local Trajectory Matrix)**.**

$$J_c(t,k) = \frac{1}{\sqrt{m}} \begin{pmatrix} a_1 x_1 \mathbb{1}_{w_{c_{1,1}}^T(t,k)x_1 \geq 0} & \cdots & a_1 x_{n_c} \mathbb{1}_{w_{c_{n_c,1}}^T(t,k)x_{n_c} \geq 0} \\ \vdots & \ddots & \vdots \\ a_m x_1 \mathbb{1}_{w_{c_{1,m}}^T(t,k)x_1 \geq 0} & \cdots & a_m x_{n_c} \mathbb{1}_{w_{c_{n_c,m}}^T(t,k)x_{n_c} \geq 0} \end{pmatrix} \in \mathbb{R}^{md \times n_c}$$

for x_i's sample of client c, and where appropriate, we fill in the undefined entries with 0 to form a matrix of dimension $md \times n$.

Note that $H = J^T J$ and $H_c = J_c^T J_c$. We also give some useful notations following the above definitions.

Notation 1.

$$\tilde{J}(t,k) = (J_{c_1}(t,k), J_{c_2}(t,k), \cdots, J_{c_N}(t,k)) \in \mathbb{R}^{md \times n}$$

Notation 2.

$$\tilde{H} = \begin{pmatrix} H_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & H_N \end{pmatrix} \in \mathbb{R}^{n \times n}$$

We also use a notation $vec(A)$ to express the vectorization of a matrix A in column-first order. Then the gradient update rule can be expressed as:

$$\begin{aligned} vec(W_c(t,k+1)) &= vec(W_c(t,k)) - \eta_l J_c(t,k)(y_c(t,k) - y_c) \\ vec(U(t+1)) &= vec(U(t)) - \frac{\eta_l \eta_g}{N} \sum_k J(t,k)(y(t,k) - y) \\ vec(W_c(t+1)) &= vec(W_c(t)) - \lambda \eta_l \sum_k J_c(t,k)(y_c(t,k) - y_c) \\ &\quad - (1-\lambda)\frac{\eta_l \eta_g}{N} \sum_k J(t,k)(y(t,k) - y) \end{aligned} \tag{8}$$

C.2 Some Useful Results

We first quote a result from Huang et al. (2021) which will be used later.

Lemma 3. *For $J(t,k)$ as defined above, with probability at least $1 - n\exp(-m\exp(-m(R\sigma^{-1} + \delta)/10))$, we have*

$$\|J(t,k) - J(0,0)\|_F \leq 2n(R\sigma^{-1} + \delta)$$

The following lemma give an approximation on the dynamics of the global model.

Lemma 4. *For $A(\lambda) = (1-\lambda)\frac{\eta_l \eta_g K}{N} H^\infty + \lambda \eta_l K \tilde{H}_c^\infty$ and $\beta(\lambda) = (1-\lambda)\frac{\eta_l \eta_g K}{N} + \lambda \eta_l K$, we have*

$$y(t) - y = -(I - A(\lambda))^t y + e(t)$$

where

$$\|e(t)\|_2 \leq \mathcal{O}\left((1 - \beta(\lambda)s_{min})^t \left(\sqrt{n}\sigma + \frac{t\beta(\lambda)n^{7/2}}{s_{min}\sigma\sqrt{m}}\right) \textbf{poly}(\log(m/\delta)\right)$$

Proof. Recall that from Appendix B, we have $[y(t)-y]-[y(t-1)-y]=v_1+v_2+v_3$, and that

$$\begin{aligned}
v_{1,i} = &-\frac{(1-\lambda)\eta_l\eta_g K}{N}\sum_{j\in[n]}(y_j(t)-y_j)H^{\infty}_{i,j}\\
&-\frac{(1-\lambda)\eta_l\eta_g}{N}\sum_{j\in[n],k}(y_j(t,k)-y_j(t))H^{\infty}_{i,j}\\
&-\frac{(1-\lambda)\eta_l\eta_g}{N}\sum_{j\in[n],k}(y_j(t,k)-y_j)(H(t,k)_{i,j}-H^{\infty}_{i,j})\\
&-\frac{(1-\lambda)\eta_l\eta_g}{N}\sum_{j\in[n],k}(y_j(t,k)-y_j)(H^{\perp}(t,k)_{i,j})
\end{aligned}$$

and similar for $v_{2,i}$ except that $i,j\in S_c$ for some client c.

Let

$$\begin{aligned}
\xi_i(t) :=& v_{1,i}(t)+v_{2,i}(t)+v_{3,i}(t)\\
&+\frac{(1-\lambda)\eta_l\eta_g K}{N}\sum_{j\in[n]}(y_j(t)-y_j)H^{\infty}_{i,j}\\
&+\lambda\eta_l K\sum_{j\in S_c}(y_j(t)-y_j)(H^{\infty}_c)_{i,j}
\end{aligned}$$

Note that by Appendix B, $\|v_3(t)\|=\frac{16\eta_l\tilde{\eta}_g K}{N}(1+2\eta_l nK)nR\|y-y(t)\|$, $\|y_c(t)-y_c(t,k)\|\le 2\eta_l nK\|y_c(t)-y_c\|$, $\|y-y(t,k)\|_2\le 2(1+2\eta_l nK)\|y-y(t)\|_2$, $\|H(w,w)-H(w_1,w_2)\|_F\le 4nR$ and $\|H(t,k)^{\perp}\|_F\le 4nR$ etc. By taking the maximum order among the terms, we have that

$$\|\xi(t)\|_2\le\mathcal{O}\Big(\frac{\beta(\lambda)n^3 s_{max}\sqrt{\log(m\delta)\log^2(n/\delta)}}{\sigma\lambda\sqrt{m}}\|y-y(t)\|_2\Big)$$

where $\beta(\lambda):=\frac{(1-\lambda)\eta_l\eta_g K}{N}+\lambda\eta_l K$.

Then

$$\begin{aligned}
y(t)-y &= (I-A(\lambda))(y(t-1)-y)+\xi(t-1)\\
&= (I-A(\lambda))^t(y(0)-y)+\sum_{\tau\in[t-1]}(I-A(\lambda))^{\tau}\xi(t-1-\tau)\\
&= -(I-A(\lambda))^t y+e(t)
\end{aligned}$$

where

$$e(t)=(I-A(\lambda))^t y(0)+\sum_{\tau\in[t-1]}(I-A(\lambda))^{\tau}\xi(t-1-\tau)$$

and since

$$\|y(0)\|_2^2\le n\sigma^2\cdot 2\log(2mn/\delta)\cdot\log^2(4n/\delta)$$

we have

$$\begin{aligned}
&\|e(t)\|_2\\
&\le\mathcal{O}\bigg((1-\beta(\lambda)s_{min})^t\Big(\sqrt{n\sigma^2}\sqrt{2\log(2mn/\delta)}\log(8n/\delta)+t\frac{\beta(\lambda)n^{7/2}\log(m/\delta)\log^2(n/\delta)}{s_{min}\sigma\sqrt{m}}\Big)\bigg)\\
&\le\mathcal{O}\bigg((1-\beta(\lambda)s_{min})^t\Big(\sqrt{n}\sigma+\frac{t\beta(\lambda)n^{7/2}}{s_{min}\sigma\sqrt{m}}\Big)\mathrm{poly}(\log(m/\delta)\bigg)
\end{aligned}$$

□

C.3 AVERAGE GENERALIZATION

When examing a new OOD sample, we would use the average of all current parameters for prediction. Therefore, we first examine the generalization performance of the average of all paramters:

$$W(t) := \frac{1}{N}\sum_c W_c(t)$$

By (8), we have:

$$vec(W(t+1)) = vec(W(t)) - \frac{\lambda\eta_l}{N}\sum_{k,c} J_c(t,k)(y_c(t,k)-y_c) - (1-\lambda)\frac{\eta_l\eta_g}{N}\sum_k J(t,k)(y(t,k)-y)$$

Lemma 5. *For $A(\lambda) = (1-\lambda)\frac{\eta_l\eta_g K}{N}H^\infty + \lambda\eta_l K\tilde{H}_c^\infty$ and $\gamma(\lambda) = (1-\lambda)\frac{\eta_l\eta_g K}{N} + \lambda\frac{\eta_l K}{N}$, we have*

$$\|W(t) - W(0)\|_F \le (y^T A(\lambda)^{-T} H^\infty A(\lambda)^{-1} y)^{1/2} + \mathcal{O}\Big(\frac{n\sigma}{s_{min}}\cdot poly(\log(m/\delta)) + \frac{n^4}{\sigma^{1/2}m^{1/4}}\cdot poly(\log(m/\delta))\Big)$$

Proof.

$$\begin{aligned}
&vec(W(T)) - vec(W(0)) \\
=&\sum_{t\in[T-1]}\Big[-(1-\lambda)\frac{\eta_l\eta_g}{N}\sum_k J(t,k)(y(t,k)-y) - \frac{\lambda\eta_l}{N}\sum_c\sum_k J_c(t,k)(y_c(t,k)-y_c)\Big] \\
=&\sum_{t\in[T-1],k} -\frac{\gamma(\lambda)}{K}J(t,k)(y(t,k)-y) \\
=&\sum_{t\in[T-1],k}\frac{\gamma(\lambda)}{K}J(t,k)(I-A(\lambda))^t y - \sum_{t\in[T-1],k}\frac{\gamma(\lambda)}{K}J(t,k)(y(t,k)-y(t)+e(k)) \\
=&\sum_{t\in[T-1]}\gamma(\lambda)J(0,0)(I-A(\lambda))^t y \\
&\quad+\sum_{t\in[T-1],k}\frac{\gamma(\lambda)}{K}(J(t,k)-J(0,0))(I-A(\lambda))^t y \\
&\quad-\sum_{t,k}\frac{\gamma(\lambda)}{K}J(t,k)(y(t,k)-y(t)+e(k)) \\
=&B_1+B_2+B_3
\end{aligned}$$

where

$$\begin{aligned}
B_1 &= \sum_{t\in[T-1]}\gamma(\lambda)J(0,0)(I-A(\lambda))^t y \\
B_2 &= \sum_{t\in[T-1],k}\frac{\gamma(\lambda)}{K}(J(t,k)-J(0,0))(I-A(\lambda))^t y \\
B_3 &= \sum_{t,k}\frac{\gamma(\lambda)}{K}J(t,k)(y(t,k)-y(t)+e(k))
\end{aligned}$$

Then by substituing in the following claims, we have

$$\begin{aligned}
&\|W(T)-W(0)\|_F \\
\le&(y^T A(\lambda)^{-T}H^\infty A(\lambda)^{-1}y)^{1/2} + \mathcal{O}\Big(\frac{n\sigma}{s_{min}}\cdot\text{poly}(\log(m/\delta)) + \frac{n^4}{\sigma^{1/2}m^{1/4}}\cdot\text{poly}(\log(m/\delta))\Big)
\end{aligned}$$

□

Claim 1. *With probability at least* $1-\delta$ *over random initialization, as* $t \to \infty$*, we have*

$$\|B_1\|_2^2 \leq y^T(A(\lambda)^{-1})^T H^\infty A(\lambda)^{-1} y + \mathcal{O}(\frac{n^2\sqrt{\log(n/\delta)}}{s_{min}^2\sqrt{m}})$$

Claim 2. *With probability at least* $1-\delta$ *over random initialization, we have*

$$\|B_2\|_2 \leq \frac{n^{3/2} poly(\log(m/\delta))}{m^{1/4}\sigma^{1/2}s_{min}^{3/2}}$$

Claim 3.

$$\|B_3\|_2 \leq (\frac{n\sigma}{s_{min}} + \frac{n^4}{s_{min}^3\sigma\sqrt{m}}) \cdot poly(\log(m/\delta))$$

The above three claims are the slightly modified version of Claim C.8-10 in Huang et al. (2021), So the proofs from there naturally extend to the proofs of these three claims.

Theorem 3. *For* T *sufficiently large,* $\sigma = \mathcal{O}(\lambda poly(\log n, \log(1/\delta))/n))$, $m = \Omega(\sigma^{-2}(n^{16} poly(\log n, \log(1/\delta), \lambda^{-1})))$*, the loss function* l *being 1-Lipschitz in its first argument, then with probability at least* $1-\delta$ *over the random initialization, the population loss* $L_{\mathcal{D}}(f)$ *of the global model* $W := \frac{1}{N}\sum_c W_c$ *is upper bounded by*

$$L_{\mathcal{D}}(f) \leq \sqrt{2y^T A(\lambda)^{-T} H^\infty A(\lambda)^{-1} y/n} + \mathcal{O}\big(\sqrt{\log(n/s_{min}\delta)/2n}\big)$$

where $A(\lambda) = (1-\lambda)\frac{\eta_l\eta_g K}{N} H^\infty + \lambda\eta_l K \tilde{H}_c^\infty$.

Proof. This is an extension of the result in Huang et al. (2021), by substituting lemma 5 into the proof of Theorem C.11 in Huang et al. (2021). □

C.4 CLIENT LEVEL

For further inspection on the algorithm, we consider a client level generalization. That is, we consider the generalization bound on a client's parameter W_c. Note that:

$$vec(W_c(t+1)) = vec(W_c(t)) - \lambda\eta_l \sum_k J_c(t,k)(y_c(t,k) - y_c) - (1-\lambda)\frac{\eta_l\eta_g}{N}\sum_k J(t,k)(y(t,k)-y)$$

We first define a matrix that will be used later.

Definition 6 (Cross Gram matrix)**.**

$$H_c^\times(t,k)_{i,j} = \frac{1}{m}\sum_r x_i^T x_j \mathbb{1}_{w_{0,r}^T x_i \geq 0, w_{c,r}^T(t,k) x_j \geq 0} \qquad \in \mathbb{R}^{n_c \times n}$$

where $i \in S_c$ *and* j *spans all clients.*

This matrix describes the effect of the global update on the client's local model.

Theorem 4. *For* T *sufficiently large,* $\sigma = \mathcal{O}(\lambda poly(\log n, \log(1/\delta))/n))$, $m = \Omega(\sigma^{-2}(n^{16} poly(\log n, \log(1/\delta), \lambda^{-1})))$*, the loss function* l *being 1-Lipschitz in its first argument, then with probability at least* $1-\delta$ *over the random initialization, the population loss* $L_D(f)$ *of the client model* W_c *is upper bounded by*

$$L_{\mathcal{D}}(f) \leq \sqrt{2y^T A(\lambda)^{-T} G_c(\lambda) A(\lambda)^{-1} y/n} + \mathcal{O}\big(\sqrt{\log(n/s_{min}\delta)/2n}\big)$$

where $G_c(\lambda) = (1-\lambda)^2\frac{\eta_l^2\eta_g^2 K^2}{N^2} H + \lambda^2\eta_l^2 K^2 H_c + \lambda(1-\lambda)\eta_l^2\eta_g K^2 H_c^\times$.

This theorem is a natural extension of theorem 3, and the proof also naturally extends. Further discussion is left for future work.

APPENDIX D EXPERIMENTS

In this section, we show more experimental results and implementation details. Sec. D.1 gives more details for implementation, including dataset details, model architectures, training, and testing details. Sec. D.2 shows more experiment results, including the performance regarding various batch sizes, the complete evaluation metrics on the binary classification on the Camelyon17 dataset.

D.1 EXPERIMENTAL DETAILS

We here introduce the complete details of datasets, data splitting, and implementation details.

Digits5. *Digits-5* zhou2020learning,fedbn with digits images showing drastic differences in font style, color, and background. We take each data source/style as a client. We train a 6-layer convolutional neural network for the digit classification, specifically, the model has 3 convolutional layers and 3 fully-connected layers, we further add batch normalization layers after the first five layers to fit the requirements of FedBN. We use the SGD optimizer with a learning rate of 0.01 and batch size of 128. The loss function is the Cross Entropy loss. The total number of training rounds is 100 with a local update epoch of 1. All input images are resized to 28×28.

Office-Caltech10. *Office-Caltech10* gong2012geodesic contains images acquired in different cameras or environments, with four different data sources in total. We take ResNet-18 as the backbone and use the SGD optimizer with a learning rate of 0.01 and batch size of 32. The loss function is the Cross Entropy loss. The total number of training rounds is 200 with a local update epoch of 1. The input images are normalized using the mean and std of Imagenet in PyTorch, which is specifically mean = [0.485, 0.456, 0.406] and std = [0.229, 0.224, 0.225]. All input images are resized to 256×256

DomainNet. *DomainNet* peng2019moment has images with different image styles (clipart, infograph, painting, quickdraw, real, and sketch). Following FedBN, we choose the top-10 class based on data amount from DomainNet containing images over 345 categories for simplicity. The training settings are the same as the Office-Caltech10 dataset, and we change the training round from 200 to 100, since the model converges faster on the DomainNet dataset.

Camelyon17. *Camelyon17* bandi2018detection shows histology images with different stains from 5 hospitals. All histopathology images are stained with the H.E. staining and show various appearances. We use the DenseNet121 as the backbone, SGD optimizer with a learning rate of 0.01, and batch size of 32. We train the model for 40 rounds in total, and the local update epoch is 1. The image input size is 96×96. The loss function is the Cross Entropy loss. Note that this dataset is a very large dataset which contains over 450,000 histology images.

Retinal. *Retinal* fundus dataset contains retinal fundus images acquired from 6 different institutions (Fumero et al., 2011; Sivaswamy et al., 2015; Almazroa et al., 2018; Orlando et al., 2020). We use the U-Net for segmentation, the optimizer is Adam with a learning rate of $1e^{-3}$ and $\beta = (0.9, 0.99)$. We train the model for 100 communication rounds in total with a local update epoch of 1. The batch size is 8. We use the dice loss and report both the Dice score and HD distance. All images are resized to 256×256.

For all datasets, we take each data source as one client and split the data of each client into train, validation, and testing sets with a ratio of 0.6, 0.2, and 0.2. We choose the best model based on the validation data and report the test performance accordingly. Code will be released after acceptance.

D.2 MORE EXPERIMENTS

Here we present more experiment results, which mainly include two parts. The first part is the study on the effects of batch size on our method's performance, and the second part is the complete evaluation results on the Camelyon17 dataset.

Effects of batch sizes. As our implementation accumulates the feature matrix iteratively during local steps (Line 7 in the algorithm box), the batch size may slightly change the values of the feature matrix. In this case, we further explore the performance changes by using different batch sizes (4,8,16,32,64,128) on the Digit5 dataset. The results are shown in Table. 6. From the results it can be observed that changing batch size has very mild effects on the final performance, the overall accuracy

Table 6: Performance using different batch sizes on the Digits dataset.

Batchsize	MNIST	SVHN	USPS	Synth	MNISTM	Average
8	99.40	93.53	98.92	99.70	97.19	97.75
16	99.49	93.85	98.87	99.68	97.41	97.86
32	99.40	93.37	99.03	99.62	96.86	97.66
64	99.37	92.87	98.87	99.50	96.14	97.35
128	99.25	92.17	98.71	99.42	95.71	97.05

changes are less than 1%. This further supports our implementation of iteratively accumulating the feature matrix during local SGD, which takes less computational cost than re-calculate all samples again after 1 local epoch.

Complete evaluation on Camelyon17. As the classification task on the Camelyon17 dataset is a binary classification, so we further report the full evaluation metrics, including the Accuracy, AUC, sensitivity, specificity, and the F1-score. From the Table. 7, it can be observed that our proposed method consistently outperforms all compared methods regarding all metrics.

Table 7: Complete evaluation metrics on the Camelyon17 dataset.

	Accuracy	AUC	Sensitivity	Specificity	F1-score
FedAvg (McMahan et al., 2017)	95.30	98.90	94.91	95.69	95.30
APFL (Deng et al., 2020)	96.42	99.35	94.42	98.42	96.42
L2SGD (Hanzely et al., 2020)	95.93	99.22	95.12	96.73	95.92
FedAlt (Pillutla et al., 2022)	97.91	99.57	96.55	97.90	97.22
PerFedAvg (Fallah et al., 2020)	96.08	99.24	95.37	96.79	96.08
FedBN (Li et al., 2021c)	95.77	99.15	95.02	96.53	95.77
FedFOMO (Zhang et al., 2021)	95.62	99.15	95.35	95.90	95.62
FedRep (Collins et al., 2021)	96.78	99.39	96.78	96.79	96.78
FedBABU (Oh et al., 2022)	95.77	99.15	95.03	96.52	95.77
FedHKD (Chen et al., 2023)	95.89	98.43	93.89	94.67	94.28
LG-Mix (Ours)	**98.75**	**99.89**	**98.63**	**98.87**	**98.75**