

A Historical Trajectory Assisted Optimization Method for Zeroth-Order Federated Learning

Chenlin Wu*, Xiaoyu He*, Zike Li*, Zibin Zheng*

*School of Software Engineering, Sun Yat-sen University, Zhuhai, China

Abstract—Federated learning heavily relies on distributed gradient descent techniques. In the situation where gradient information is not available, the gradients need to be estimated from zeroth-order information, which typically involves computing finite-differences along isotropic random directions. This method suffers from high estimation errors, as the geometric features of the objective landscape may be overlooked during the isotropic sampling. In this work, we propose a non-isotropic sampling method to improve the gradient estimation procedure. Gradients in our method are estimated in a subspace spanned by historical trajectories of solutions, aiming to encourage the exploration of promising regions and hence improve the convergence. The proposed method uses a covariance matrix for sampling which is a convex combination of two parts. The first part is a thin projection matrix containing the basis of the subspace which is designed to improve the exploitation ability. The second part is the historical trajectories. We implement this method in zeroth-order federated settings, and show that the convergence rate aligns with existing ones while introducing no significant overheads in communication or local computation. The effectiveness of our proposal is verified on several numerical experiments in comparison to several commonly-used zeroth-order federated optimization algorithms.

Index Terms—Federated Learning, Zeroth-Order Optimization, Historical Trajectories.

I. INTRODUCTION

THIS work considers solving the unconstrained federated optimization problem [1] defined as

$$\min_{w \in \mathbb{R}^n} f(w) = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{\xi \sim \mathcal{D}_i} [F(w; \xi)] \quad (1)$$

where N is the number of clients, ξ is a data sample, \mathcal{D}_i is the local data distribution associated with client i , and $w \in \mathbb{R}^n$ is the decision vector. The task is to optimize the global objective f with clients only having access to their local data samples. The data distributions \mathcal{D}_i and \mathcal{D}_j may differ for $i \neq j$, which is also known as the non independent and identically distributed (non-IID) setting.

Efficiently solving problem (1) is the key to implementing federated learning systems. One representative of federated optimization methods is FedAvg [2]. The clients in FedAvg fetch the global parameters from the server, update them using the stochastic gradient descent (SGD) method with their local data, and upload the parameters back to the server for aggregation. The server aggregates the obtained parameters without accessing client data or performing local model updates, which

ensures both data privacy and communication efficiency. Various FedAvg variants have been proposed, improving FedAvg in local update [3], client selection [4], and the aggregation mechanism [5], [6]. One of the most common features shared by these methods is that they rely on gradient information for guiding the clients' local updates, measuring the learning progress, or encoding the model to be transferred between the server and clients. The clients thus must support automatic differentiation or have manually coded gradients.

We focus on solving the problem (1) in zeroth-order settings. That is, the individual objective F is a black box, and for a given vector x and a data sample ξ , one can obtain no information other than the objective value $F(x; \xi)$. This simulates the real-world scenario where the structure of the objective is unknown or evaluating the gradients is intractable, typically due to security or commercial considerations. An example that has emerged recently is the federated learning system based on closed-source large language models where the neural architectures are not released publicly. Client-side users in this case cannot evaluate the gradients, making it intractable to deploy downstream tasks such as prompt tuning [7], [8], [9], [10]. Zeroth-order optimization also reduces memory usage as no backward pass is required [11]; this is particularly useful in certain federated systems where clients involve low-performing devices.

Gaussian smoothing [12] is one of the most popular techniques for zeroth-order optimization, which samples a random vector from an isotropic Gaussian distribution, and outputs the finite-difference along this vector as a gradient estimation. It is easy to implement, enjoys strong theoretical guarantees, and can be used in combination with established gradient-based algorithms. However, the estimation output by Gaussian smoothing is biased and noisy, and how accurate the estimation is depends heavily on the objective landscape features [13]. This may prevent the convergence of Gaussian smoothing based zeroth-order optimization, as the used isotropic Gaussian cannot fully exploit the objective landscape.

In this paper, we improve Gaussian smoothing and extend it to the federated setting. The key idea is to use a non-isotropic Gaussian in gradient estimation, aiming to exploit promising subspaces that are revealed from historical training trajectories. Precisely, when sampling the vector along which the finite-difference is performed, our algorithm utilizes a covariance matrix that is the convex combination of two parts. The first part is a thin projection matrix containing the basis of the subspace spanned by recent training trajectories, which is designed to improve the exploitation ability. The second part

is an identity matrix; it acts as a safeguard for preventing the Gaussian model from degradation, and therefore enhances global exploration. By updating the covariance matrix at certain periodic intervals, the communication overheads can be made negligible compared to existing methods. The simulation results suggest that the new method is robust and efficient.

In the remainder of this paper, we first provide preliminaries on Gaussian smoothing in Section II. The new algorithm is then detailed in Section III and verified with numerical simulations in Section IV. We discuss related work in Section V and conclude the paper in section VI.

a) *Notations:* $\mathcal{N}(\mu, C)$ represents a Gaussian distribution with expectation μ and covariance matrix C . I_n denotes an $n \times n$ identity matrix. $\|x\|_2$ denotes the ℓ_2 norm of a vector x . $\mathbb{E}[x]$ is the expectation of a random variable x . $\text{Tr}[M]$ is the trace of a matrix M .

II. NON-ISOTROPIC GAUSSIAN SMOOTHING

In this section, we assume the decision space \mathbb{R}^n is equipped with some generic vector norm $\|\cdot\|$. Its dual norm is denoted by $\|\cdot\|_*$.

A. Preliminaries on Gaussian smoothing

We firstly define the Gaussian smoothing based gradient estimation method that involves a non-isotropic covariance matrix. The method relies on the smoothness property of a function, which we detail as below:

Definition 1 (Smoothness). *We say a function h is L -smooth if it is differentiable and its gradient satisfies:*

$$\|\nabla h(x) - \nabla h(y)\|_* \leq L\|x - y\|, \forall x, y \quad (2)$$

for some constant $L \in \mathbb{R}_+$.

The smoothness of a function h implies that it has a quadratic bound:

$$|h(y) - h(x) - \langle \nabla h(x), y - x \rangle| \leq \frac{L}{2}\|y - x\|^2, \forall x, y. \quad (3)$$

Gaussian smoothing is a method to generate a smooth surrogate for a function h :

Definition 2 (Non-isotropic Gaussian Smoothing). *Given a function $h : \mathbb{R}^n \rightarrow \mathbb{R}$, its Gaussian smoothing is defined as:*

$$h_{\mu, C}(x) = \mathbb{E}_{v \sim \mathcal{N}(0, C)}[h(x + \mu v)] \quad (4)$$

where $\mu \in \mathbb{R}_+$ is called the smoothing radius, and $C \in \mathbb{R}^{n \times n}$ is a positive-definitive covariance matrix. When $C = I_n$, the function $h_{\mu, C}(x)$ reduces to the standard (isotropic) Gaussian smoothing as in [14]. For generic settings of the covariance matrix C , we call $h_{\mu, C}(x)$ the non-isotropic Gaussian smoothing.

The Gaussian smoothing $h_{\mu, C}$ is differentiable, and its gradient has a closed-form expression that only depends on zeroth-order information [13]:

Lemma 1 (Differentiability). *For a function $h : \mathbb{R}^n \rightarrow \mathbb{R}$, its Gaussian smoothing is differentiable and has the gradient defined as:*

$$\nabla h_{\mu, C}(x) = \mathbb{E}_{v \sim \mathcal{N}(0, C)} \left[\frac{h(x + \mu v) - h(x - \mu v)}{2\mu} C^{-1} v \right]. \quad (5)$$

The above shows that we can compute an ascent direction of $h_{\mu, C}$ as

$$g = \frac{h(x + \mu v) - h(x - \mu v)}{2\mu} v \quad (6)$$

where v is drawn from $\mathcal{N}(0, C)$. It is easy to see

$$\mathbb{E}[\langle g, \nabla h_{\mu, C}(x) \rangle] = \|\sqrt{C} \nabla h_{\mu, C}(x)\|_2^2 \geq 0.$$

Thus, g is indeed a stochastic ascent direction of $h_{\mu, C}$. Hereinafter we will call g the stochastic gradient of $\nabla h_{\mu, C}(x)$.

The importance of Gaussian smoothing is that it acts as a surrogate to the original function provided the latter is smooth:

Lemma 2 (Properties of non-isotropic Gaussian smoothing). *Assume a function $h : \mathbb{R}^n \rightarrow \mathbb{R}$ is L -smooth. Then, the Gaussian smoothing $h_{\mu, C}$ is also L -smooth. In addition, its difference to the original function can be bounded as*

$$\|\nabla h_{\mu, C}(x) - \nabla h(x)\|_* \leq \frac{L\mu}{2} \mathbb{E}[\|v\|^2 \|C^{-1} v\|_*], \quad (7)$$

and

$$|h_{\mu, C}(x) - h(x)| \leq \frac{L\mu^2}{2} \mathbb{E}[\|v\|^2], \quad (8)$$

where the expectation is taken over $v \sim \mathcal{N}(0, C)$.

B. Covariance matrix specification

Lemmas 1 and 2 state that the Gaussian smoothing $h_{\mu, C}$ serves as a good surrogate to the original objective function h , as it is smooth and has closed-form gradients. Optimizing $h_{\mu, C}$ instead of h would nevertheless introduce an approximation error due to the difference between these two functions. On the other hand, (8) shows that this error is bounded, and the bound depends on the covariance matrix C . It implies that we can control the covariance matrix to reduce the error, which is exactly why we need the non-isotropic Gaussian smoothing. A trivial choice to minimize the bound in (8) is letting $C \rightarrow 0$. This is however meaningless, as it would bring rounding errors in practice.

In this work we suggest constraining the covariance matrix as

$$C = (1 - \alpha)I_n + \alpha Q Q^\top, \quad (9)$$

where $\alpha \in [0, 1]$ is a constant and $Q \in \mathbb{R}^{n \times m}$ is a projection matrix satisfying $Q^\top Q = I_m$. The term m here is a user specified hyperparameter that should be smaller than n . The matrix Q is kept orthogonal, as its magnitude is not important: the magnitude of C can be absorbed into the constant μ . With $\alpha \rightarrow 1$, the gradients output by Gaussian smoothing are located in a subspace spanned by the columns of Q . If these gradients are used for guiding the search, the algorithm will keep staying in this subspace. On the contrary, taking $\alpha \rightarrow 0$ leads to the standard Gaussian smoothing, which estimates

the gradient in the whole space \mathbb{R}^n and hence encourages the algorithm to explore unexplored regions. In other words, the covariance matrix C controls the balance between exploration and exploitation.

We propose that an appropriate covariance matrix C should capture insensitive directions over the objective landscape. A direction is called sensitive if moving along this direction causes a rapid change on the objective value and vice versa. To see why the sensitiveness matters, consider an ideal setting where the objective is a quadratic function $f(x) = \frac{1}{2}x^\top Hx$ with symmetric positive definite Hessian $H \in \mathbb{R}^{n \times n}$. The objective function is 1-smooth w.r.t. the Euclidean norm $\|x\|^2 = x^\top Hx$ and its dual norm $\|x\|_*^2 = x^\top H^{-1}x$. In this setting, the matrix Q for minimizing the bound in (8) can be found by solving the following problem:

$$\begin{cases} \min_{Q \in \mathbb{R}^{n \times m}} & \mathbb{E}_{v \sim \mathcal{N}(0, C)}[\|v\|^2] \\ \text{s.t.} & C = (1 - \alpha)I_n + \alpha QQ^\top \\ & Q^\top Q = I_m \end{cases} \quad (10)$$

The problem is solved exactly when Q contains the principal components of H^{-1} ; see Appendix C for details. That is, the optimal Q corresponds to the m most insensitive directions.

When solving a problem whose Hessian information is not accessible, we can approximate its insensitive directions by tracking the solution trajectories. An intuitive reason is: the algorithm has to repeatedly exploit the subspace spanned by the insensitive directions, as the progress in this subspace is harder to achieve than in that spanned by sensitive directions. Here we use simulations on a 2-dimensional example to support this statement. Let us define $f(x) = 10^3x_1^2 + x_2^2$. So the most sensitive direction is $e^{[1]} = (1, 0)^\top$ and the most insensitive one is $e^{[2]} = (0, 1)^\top$. We consider the stochastic gradient descent iterations $x^{[t+1]} = x^{[t]} - 10^{-4}g^{[t]}$, where gradients are sampled from $g^{[t]} \sim \mathcal{N}(\nabla f(x^{[t]}), \epsilon^2 I_2)$ with ϵ measuring the noise. Defining

$$\gamma(\epsilon) = \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{I}\{|g_2^{[t]}| > |g_1^{[t]}|\}$$

as the metric for indicating whether the solution trajectories can capture the insensitive direction. Note here that $|g_2^{[t]}| > |g_1^{[t]}|$ means at iteration t the solution moves along a direction that makes an acute angle with the most insensitive direction $e^{[2]}$, so a high $\gamma(\epsilon)$ value would give a positive evidence to support our statement. We simulate the above iterations with ten independently runs and noise variance $\epsilon \in \{10^{-2}, 10^{-1}, \dots, 10^5\}$. The number of iterations T is set to 10^4 . The initial solution $x^{[0]}$ is set to $(10^3, 10^3)^\top$. The results are presented in Table I. It is found that when the noise level is low, the solution trajectories make an acute angle with the insensitive direction $e^{[2]}$ with high probabilities. This experiment states that trajectories of the solutions produced in a search procedure do capture the insensitive directions. Conversely, the matrix Q can be specified in (9) with principal components of solution trajectories as they are expected to coincide with the insensitive directions and therefore improve

the gradient estimation accuracy. We implement this idea in federated settings in the next section.

ϵ	10^{-2}	10^{-1}	10^0	10^1	10^2	10^3	10^4	10^5
$\gamma(\epsilon)$	0.9993	0.9423	0.8268	0.7096	0.5931	0.4979	0.4924	0.4919

TABLE I: Ratio of correctly capturing insensitive directions by tracking solution trajectories

III. THE PROPOSED METHOD

We apply the non-isotropic Gaussian smoothing described above to the federated optimization problem (1) and propose the zeroth-order federated optimization algorithm assisted by historical trajectories (ZOFedHT).

A. Implementation

The ZOFedHT algorithm is given in Algorithm 1. Apart from the initial solution x_0 , the algorithm receives several additional hyperparameters: a sequence of step-sizes η_r , a smoothing radius μ , a positive integer τ denoting the length of historical trajectories, a constant K denoting the number of local updates, and a combination coefficient $\alpha \in (0, 1)$ used for constraining the covariance matrix.

The optimization loop of ZOFedHT follows that of FedAvg, but differs in 1) the use of non-isotropic Gaussian smoothing to estimate gradients and 2) the additional communication/computation for building the non-isotropic Gaussian covariance. Precisely, we divide the optimization loop into several rounds, and in the r -th round, we specify a covariance matrix C_r to estimate gradients on the client side. This matrix is kept constant over local updates and across differ clients. In order to improve convergence, we build the matrix C_r with solution trajectories collected on the server side, using the non-isotropic Gaussian smoothing method described in Section II. We detail the implementations below.

At round r , the server samples randomly and uniformly a set of clients, denoted by \mathcal{W}_r , and broadcasts the solution x_r to these clients (lines 3-4). If r is a multiple of τ , we collect the latest τ solution trajectories $\Delta_{r-1}, \dots, \Delta_{r-\tau}$, where Δ_j denotes the change of the server-side solution computed at the end of round j (line 21). We apply QR factorization on a matrix holding these trajectories as columns (line 6) and obtain a τ -dimensional subspace whose basis is encoded in an $n \times \tau$ projection matrix Q_r . One note that, instead of the whole covariance matrix C_r , only Q_r is broadcast to the selected clients (line 7); this is the key to reducing communication overheads.

On the client side, if the round index r is a multiple of τ , the matrix Q_r can be fetched. We then recover the covariance matrix C_r via linearly combining $Q_r Q_r^\top$ with the identity matrix I_n (line 10), following the constrained model (9) described in Section II. Otherwise, i.e., r is not a multiple of τ , we keep the covariance matrix unchanged. In the case of $r < \tau$ where no enough solution trajectories are presented, we fix C_r to I_n . It means the standard Gaussian smoothing is used in the first τ rounds.

After building the covariance matrix C_r , the clients perform local updates using Gaussian smoothing based zeroth-order optimization (lines 11-17). In client i , we denote the solution at round r and before the k -th local update by $w_{r,k}^i$. At the k -th local update, we first draw a Gaussian vector $v_{r,k}^i$ from the non-isotropic Gaussian $\mathcal{N}(0, C_r)$ (line 13) and a random sample $\xi_{r,k}^i$ from the local data distribution \mathcal{D}_i (line 14). By applying the gradient estimation method (6) to the component objective $F(\cdot; \cdot)$, we obtain a gradient estimation $g_{r,k}^i$ (line 15). The solution is then updated along the negative gradient direction with a step-size η_r fixed during round r . After K steps of the local updates, all clients in \mathcal{W}_r update their local solutions to the server, and the server obtains a new global solution via aggregation (lines 18-20).

Algorithm 1 ZOFedHT

```

1: Input  $x_0 \in \mathbb{R}^n, \eta_r \in \mathbb{R}_+, \mu \in \mathbb{R}_+, \tau \in \mathbb{Z}_+$ 
2: for  $r \leftarrow 0, 1, \dots, R-1$  do
3:   Select a client set  $\mathcal{W}_r$  randomly and uniformly with
   replacement
4:   Broadcast  $x_r$  to clients in  $\mathcal{W}_r$ 
5:   if  $r \in \{\tau, 2\tau, \dots\}$  then
6:      $Q_r = \text{QR}([\Delta_{r-1}, \dots, \Delta_{r-\tau}])$ 
7:     Broadcast  $Q_r$  to clients in  $\mathcal{W}_r$ 
8:   end if
9:   for each client  $i \in \mathcal{W}_r$  do
10:     $C_r \leftarrow \begin{cases} I, & \text{if } r < \tau \\ (1-\alpha)I + \alpha Q_r Q_r^\top, & \text{if } r \in \{\tau, 2\tau, \dots\} \\ C_{r-1} & \text{otherwise} \end{cases}$ 
11:     $w_{r,0}^i \leftarrow x_r$ 
12:    for  $k \leftarrow 0, 1, \dots, K-1$  do
13:       $v_{r,k}^i \sim \mathcal{N}(0, C_r)$ 
14:       $\xi_{r,k}^i \sim \mathcal{D}_i$ 
15:       $g_{r,k}^i = \frac{F(w_{r,k}^i + \mu v_{r,k}^i; \xi_{r,k}^i) - F(w_{r,k}^i - \mu v_{r,k}^i; \xi_{r,k}^i)}{2\mu} v_{r,k}^i$ 
16:       $w_{r,k+1}^i = w_{r,k}^i - \eta_r g_{r,k}^i$ 
17:    end for
18:    Upload  $w_{r,K}^i$  to the server
19:  end for
20:   $x_{r+1} = \frac{1}{|\mathcal{W}_r|} \sum_{i \in \mathcal{W}_r} w_{r,K}^i$ 
21:   $\Delta_r = x_{r+1} - x_r$ 
22: end for

```

B. Complexity

Since the covariance matrix C_r is not built from scratch but instead recovered from the projection matrix $Q_r \in \mathbb{R}^{n \times \tau}$, only $n\tau$ additional entries need to be broadcast to the clients (line 7). In addition, the broadcast is performed at every τ rounds, so the communication overhead is $\mathcal{O}(n)$ per-round, which is insignificant compared to the standard FedAvg or its zeroth-order implementations.

Computation overheads yield on both the client side and the server side. On the server side, the thin QR factorization takes $\mathcal{O}(n\tau^2)$ time for every τ rounds, so the averaged time complexity is $\mathcal{O}(n\tau)$. On the client side, additional computation is caused by the Gaussian sampling (line 13). Recall that the

covariance matrix takes the form of $C_r = (1-\alpha)I_n + \alpha Q_r Q_r^\top$, a sample drawn from $\mathcal{N}(0, C_r)$ can be decomposed into two parts as:

$$v \sim \mathcal{N}(0, C_r) \Leftrightarrow v = \sqrt{1-\alpha}v_1 + \sqrt{\alpha}Q_r v_2$$

where $v_1 \in \mathbb{R}^n$ and $v_2 \in \mathbb{R}^\tau$ are independently sampled from $\mathcal{N}(0, I_n)$ and $\mathcal{N}(0, I_\tau)$, respectively. This means that sampling the non-isotropic Gaussian can be achieved via a matrix-vector multiplication followed by a vector addition, taking $\mathcal{O}(n\tau)$ per local update. In practice, we recommend choosing a fixed and small τ , e.g., $\tau = \mathcal{O}(1)$. In this setting, ZOFedHT enjoys the same computation and communication complexity as FedAvg.

C. Convergence properties

We make the following assumptions regarding problem (1).

Assumption 1. The objective function $F(x; \xi)$ is L -smooth in $x \in \mathbb{R}^n$ for all ξ .

Assumption 2. The client-side data sampling has bounded variance, i.e., there exists some constant $\sigma_l \in \mathbb{R}_+$ such that

$$\mathbb{E}_{\xi \sim \mathcal{D}_i} [\|\nabla F(x; \xi) - \mathbb{E}_{\xi \sim \mathcal{D}_i} [\nabla F(x; \xi)]\|] \leq \sigma_l^2$$

holds for all $i \in \{1, \dots, N\}$ and $x \in \mathbb{R}^n$.

Assumption 3. The dissimilarity between each local gradient and the global gradient is bounded, i.e., there exists some constant $\sigma_g \in \mathbb{R}_+$ such that

$$\|\mathbb{E}_{\xi \sim \mathcal{D}_i} [\nabla F(x; \xi)] - \nabla f(x)\| \leq \sigma_g$$

holds for all $i \in \{1, \dots, N\}$ and $x \in \mathbb{R}^n$.

Assumption 4. The global objective is bounded from below by some constant f_* , i.e., $f(x) \geq f_*$ for all $x \in \mathbb{R}^n$.

Assumptions 1, 2 and 4 are customary in analyzing zeroth-order stochastic optimization algorithms. Assumption 3 measures the heterogeneity of the data distribution. For example, when $\sigma_g = 0$, the data become IID and the problem degenerates to the classical distributed optimization problem.

Below we characterize the convergence property of ZOFedHT w.r.t. the standard ℓ_2 norm. The proof can be found in Appendix D.

Theorem 1. Let Assumptions 1 to 4 hold with ℓ_2 norm $\|\cdot\| = \|\cdot\|_2$. Choose constant step-sizes

$$\eta_r = \eta = \frac{1}{6} \sqrt{\frac{(f(x_0) - f_*)M}{(n+4)(\sigma_g^2 + \sigma_l^2)RKL}} \quad (11)$$

and constant client set-sizes $|\mathcal{W}_r| = M$. Suppose the number of rounds R is sufficiently large and the number of local updates satisfies $K \leq n$. Then, we have

$$\frac{1}{R} \sum_{r=0}^{R-1} \mathbb{E} [\|\nabla f(x_r)\|^2] \leq \frac{96}{1-\alpha} \sqrt{\frac{(f(x_0) - f_*)(n+4)(\sigma_g^2 + \sigma_l^2)}{RKM}} + \frac{8\mu^2 L^2 (n+6)^3}{(1-\alpha)^2}.$$

The above states that ZOFedHT achieves a convergence rate of $\mathcal{O}(\sqrt{\frac{n}{RKM}})$ when μ is sufficiently small. The dependence

on R, K , and M coincides with that of modern FedAvg implementations [15]. ZOFedHT suffers an n -dependent slowdown, which is the price paid for not knowing the gradient. This aligns with the best known convergence rate achieved by zeroth-order federated optimization given in [1]. On the other hand, the impact of non-isotropic Gaussian smoothing is unknown yet, as choosing a non-zero α does not tighten the above bound. In the next section we verify the effectiveness of using non-isotropic Gaussian smoothing via numerical studies.

IV. NUMERICAL STUDIES

We verify the performance of ZOFedHT via training three machine learning models including logistic regression (LR), support vector machine (SVM), and multilayer perceptron (MLP). The LR model is convex and smooth; we choose it to test the local exploitation ability of ZOFedHT. SVM is also convex. But it employs a hinge loss and therefore deviates from the smoothness assumption. We choose this model to test the robustness of ZOFedHT against the landscape irregularity. The MLP model has a fully connected hidden layer with 50 neurons and uses the sigmoid activation function at both the hidden and output layers. The model is smooth but non-convex, and we consider this model mainly for verifying the algorithms' global exploration ability.

We implement two competitors namely ZOFedAvg-SGD and ZOFedAvg-GD by equipping FedAvg with zeroth-order versions of SGD and GD respectively. ZOFedAvg-SGD can be considered as a special instance of ZOFedHT with $\alpha = 0$ and all other settings are kept the same as ZOFedHT. ZOFedAvg-GD differs from ZOFedAvg-SGD in that it uses the standard Gaussian smoothing to estimate the full-batch gradients on the client side. In all algorithms, the smoothing radius is $\mu = 10^{-4}$ and the number of local updates is $K = 50$. All algorithms use step-sizes decreasing over rounds as $\eta_r = \eta_0 / \sqrt{r + 1}$, where η_0 is tuned with a grid-search in $\{0.1, 1, 10\}$. In both ZOFedHT and ZOFedAvg-SGD, minibatching is used in the client-side data sampling and the batch size is fixed to 64. For ZOFedHT, the parameter α is tuned in the range $\{0.1, 0.2, \dots, 0.9\}$ and the parameter L is fixed to 5. On each test instance, all algorithms run three times independently, and we report the results from the run achieving the best final training loss.

Three widely used benchmark datasets, including mnist [16], fashion-mnist [17], and rcv1 [18], are chosen. Our experiments simulate a binary classification problem using these datasets. For mnist and fashion-mnist which containing images of digits from 0 to 9, we re-classify the samples into two categories: one with digits 0 to 4 and the other one with 5 to 9. We set the number of clients N to 100, and at each round 10 clients are sampled uniformly. Both the IID setting and the non-IID setting are considered.

We first consider the IID case. In this case, we partition the dataset randomly and uniformly into N parts and assign each part to a distinct client before the optimization. Figure 1 displays the convergence trajectories of the algorithms. ZOFedHT is the best performer on all test instances, demonstrating the effectiveness of use of non-isotropic Gaussian smoothing.

In the non-IID case, instead of partitioning the datasets randomly, we first sort the samples according to their labels and then partition the sorted dataset into N parts evenly. Clients in this way may receive samples having only a subset of all available labels. The results are shown in Figure 2. In most cases on the mnist and fashion-mnist datasets, ZOFedHT performs the best, achieving faster convergence speed and lower training loss than the competitors. On the rcv1 dataset, ZOFedHT converges fast in the early phase but is outperformed by ZOFedAvg-SGD in the long run. The performance degradation might not come as a surprise, as the rcv1 dataset is highly sparse while the isotropic Gaussian smoothing has been demonstrated to be effective in handling sparsity [19]. This finding implies that ZOFedHT is more suitable for solving dense problems.

V. RELATED WORK

Gaussian smoothing is perhaps the most well-known technique for gradient estimation [14], [20]. Although using an identity covariance matrix could be effective in certain cases [19], it has been long recognized that incorporating second-order information in Gaussian smoothing could enhance convergence. For example, [13] suggested that, for convex quadratic problems, the optimal covariance matrix in Gaussian smoothing should be proportional to the inverse of the Hessian, coinciding with our statement in Section II-B. Several approaches exist for exploiting Hessian information when using Gaussian smoothing. [21] proposed a quasi-Newton procedure for extracting the curvature information from gradient estimators output by Gaussian smoothing. [22] proposed a four-point rule for estimating the Hessian-vector product, which mimics the two-point rule in Gaussian smoothing. [23] suggested estimating the Hessian explicitly and using the inverse of the estimated Hessian as the covariance matrix.

While improving Gaussian smoothing in convex problems, using Hessian information might cause divergence in non-convex cases. A workaround is to exploit historical trajectories instead of the Hessian information. One remark example is the guided evolution strategy method for reinforcement learning [24]. It estimates gradients in the parameter space using Gaussian smoothing, and explores the principal components of latest gradients to bias the sampling procedure. [25] further improves this method via self-adapting the importance of subspace sampling. In this paper we extend these methods to federated settings and provide theoretical guarantees for those methods using historical trajectories.

Various works exist on combining zeroth-order optimization and federated learning [1]. In [26], a zeroth-order federated Newton method was proposed in the setting that the clients use full-batch sampling. The method enjoys a super-linear convergence rate, but the per-iteration complexity is costly due to the Hessian estimation. Isotropic Gaussian smoothing has also been extended to bi-level federated optimization [27], demonstrating a good empirical performance on handling hard problems. In [28], the authors also explored the historical trajectory information in zeroth-order federated optimization. Their method differs from ours in that the historical trajectories

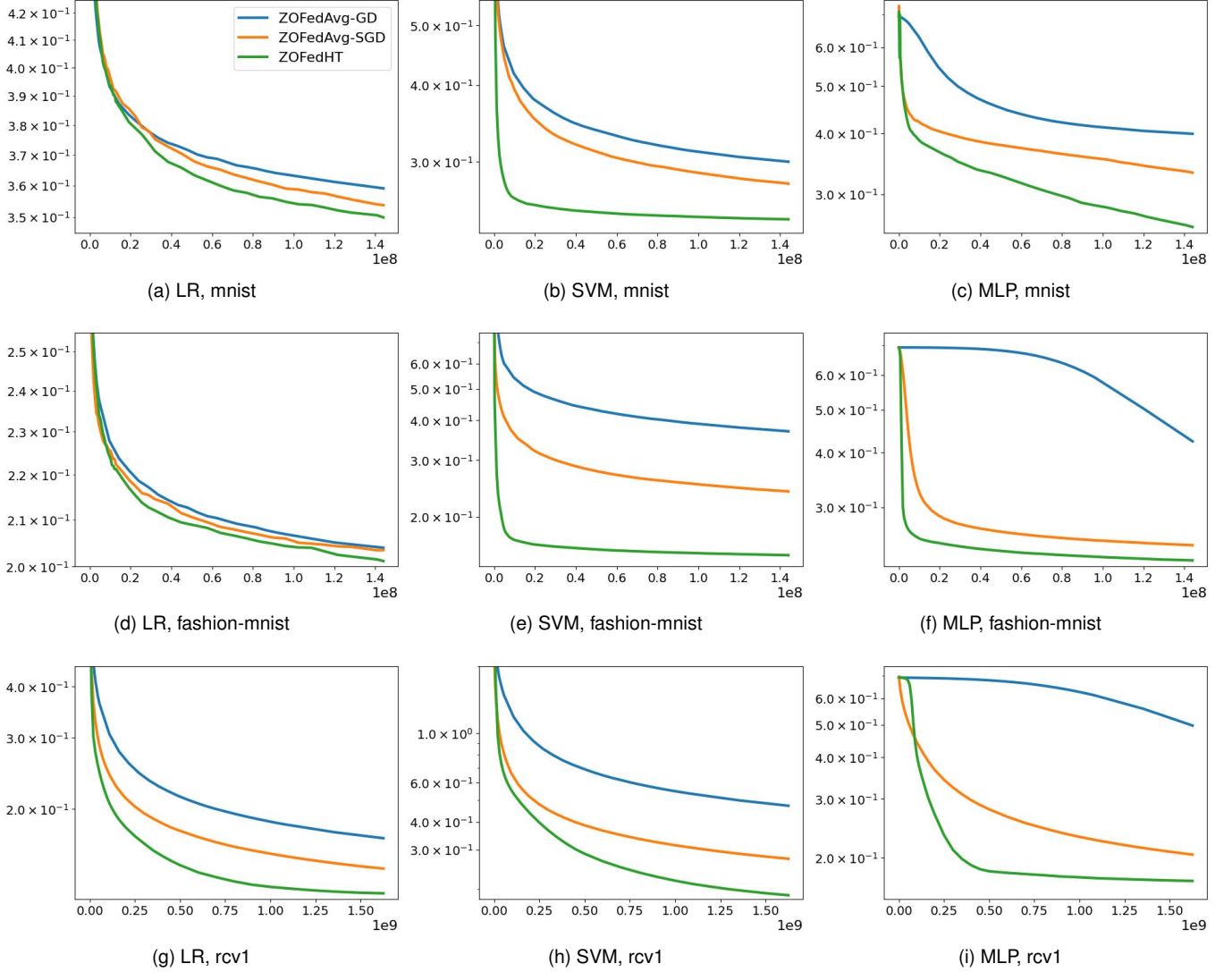


Fig. 1: Training loss versus the number of function evaluations obtained in the IID setting.

are used for variance reduction rather than improving Gaussian smoothing.

VI. CONCLUSION

We present in this article the ZOFedHT algorithm for zeroth-order federated optimization. ZOFedHT senses promising subspaces from historical trajectories of the global solution and improves convergence via using a non-isotropic Gaussian smoothing procedure on the client side. ZOFedHT aligns with existing methods in terms of the convergence rate, introducing no significant computation or communication overheads. The numerical studies suggest that ZOFedHT performs competitive to or is better than the state-of-the-arts especially in dense problems.

APPENDIX A PROOF OF LEMMA 1

By [13, lemma 1] we have

$$\nabla h_{\mu,C}(y) = \mathbb{E}_{v \sim \mathcal{N}(0,C)} \left[\frac{h(x + \mu v) - h(x)}{\mu} C^{-1} v \right].$$

Substituting v with $-v$ and using the fact $v \sim \mathcal{N}(0,C) \Leftrightarrow -v \sim \mathcal{N}(0,C)$, we have

$$\nabla h_{\mu,C}(y) = \mathbb{E}_{v \sim \mathcal{N}(0,C)} \left[\frac{h(x) - h(x - \mu v)}{\mu} C^{-1} v \right].$$

Equation (5) can be obtained by summing the above two.

APPENDIX B PROOF OF LEMMA 2

For simplicity, we denote $\mathbb{E}_{v \sim \mathcal{N}(0,C)}$ by \mathbb{E} . The first statement can be proved by definition:

$$\begin{aligned} & \|\nabla h_{\mu,C}(y) - \nabla h_{\mu,C}(x)\|_* \\ &= \|\mathbb{E}[\nabla h(y + \mu v)] - \mathbb{E}[\nabla h(x + \mu v)]\|_* \\ &= \|\mathbb{E}[\nabla h(y + \mu v) - \nabla h(x + \mu v)]\|_* \\ &\leq \mathbb{E}[\|\nabla h(y + \mu v) - \nabla h(x + \mu v)\|_*] \\ &\leq L\|y - x\|, \end{aligned}$$

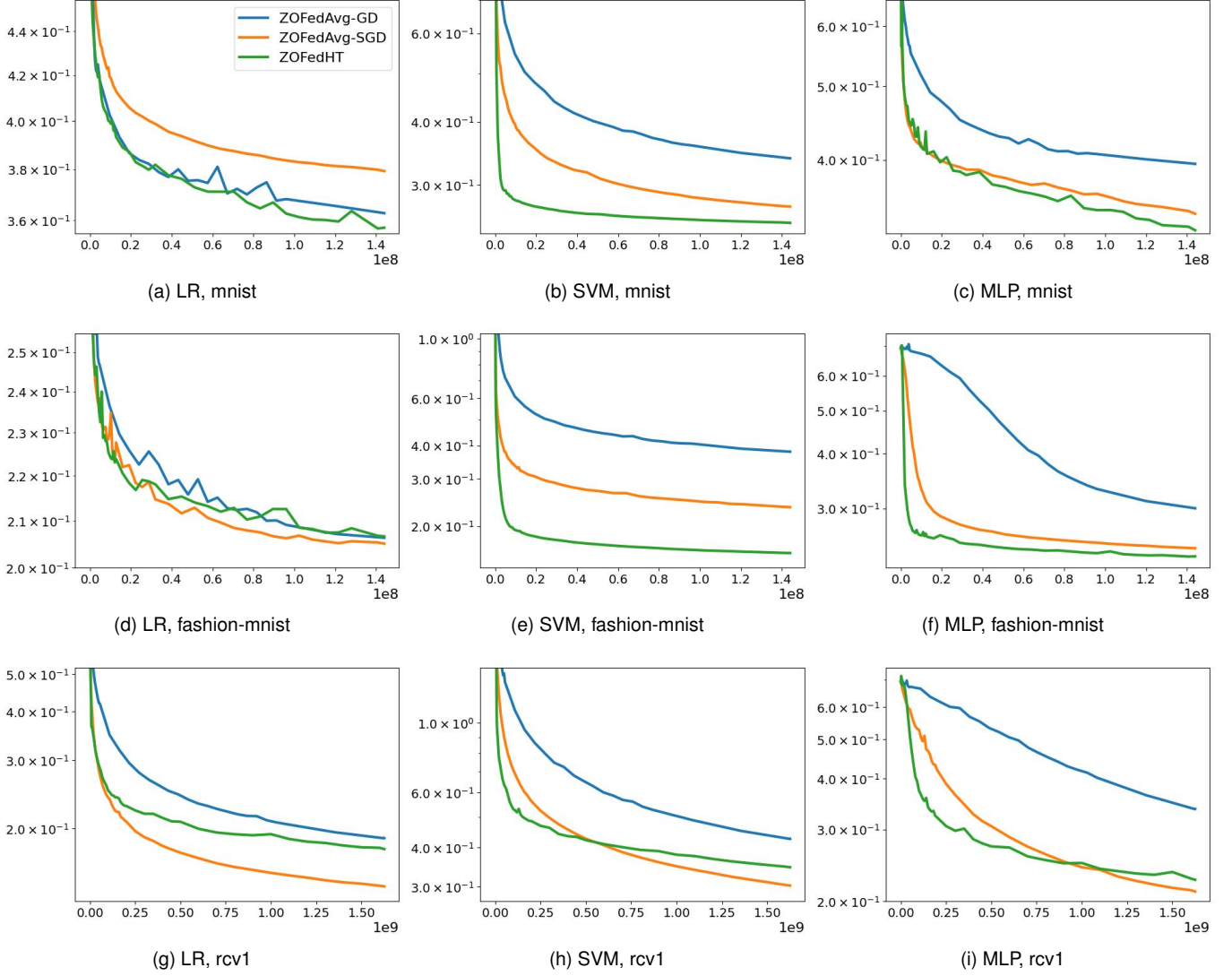


Fig. 2: Training loss versus the number of function evaluations obtained in the non-IID setting.

where the first equation follows from Definition 2, the first inequality uses Jensen's inequality, and the last step is due to the L -smoothness of h .

The second statement can be derived from the smoothness assumption and the properties of Gaussian variables:

$$\begin{aligned}
 & \|\nabla h_{\mu,C}(x) - \nabla h(x)\|_* \\
 & \stackrel{(5)}{=} \left\| \mathbb{E} \left[\frac{h(x + \mu v) - h(x - \mu v)}{2\mu} C^{-1} v \right] - \nabla h(x) \right\|_* \\
 & \stackrel{(a)}{=} \left\| \mathbb{E} \left[\frac{h(x + \mu v) - h(x - \mu v)}{2\mu} C^{-1} v \right] - C^{-1} \mathbb{E}[v v^\top] \nabla h(x) \right\|_* \\
 & = \left\| \mathbb{E} \left[\frac{h(x + \mu v) - h(x - \mu v)}{2\mu} - v^\top \nabla h(x) \right] C^{-1} v \right\|_* \\
 & = \mathbb{E} \left[\left| \frac{h(x + \mu v) - h(x - \mu v)}{2\mu} - v^\top \nabla h(x) \right| \|C^{-1} v\|_* \right] \\
 & \stackrel{(b)}{\leq} \frac{L\mu}{2} \mathbb{E}[\|v\|^2 \|C^{-1} v\|_*],
 \end{aligned}$$

where (a) uses the properties of the Gaussian distribution, and

(b) is due to

$$|h(x + \mu v) - h(x - \mu v) - 2\langle \nabla h(x), \mu v \rangle| \leq L\|\mu v\|^2, \quad (12)$$

which follows from (3).

The third statement, again, can be obtained using the smoothness assumption.

$$\begin{aligned}
 |h_{\mu,C}(x) - h(x)| &= |\mathbb{E}[h(x + \mu v)] - h(x)| \\
 &\stackrel{(a)}{=} |\mathbb{E}[h(x + \mu v) - h(x) - \mu v^\top \nabla h(x)]| \\
 &\stackrel{(b)}{\leq} \mathbb{E}[|h(x + \mu v) - h(x) - \mu v^\top \nabla h(x)|] \\
 &\stackrel{(3)}{\leq} \frac{\mu^2 L}{2} \mathbb{E}[\|v\|^2],
 \end{aligned}$$

where (a) is due to $\mathbb{E}[v] = 0$, and (b) uses Jensen's inequality.

APPENDIX C

SOLUTION TO PROBLEM (10)

Let $u_1 \sim \mathcal{N}(0, I_n)$, $u_2 \sim \mathcal{N}(0, I_m)$ be two independent vectors. Let $v = \sqrt{1 - \alpha} u_1 + \sqrt{\alpha} Q u_2$. It is then clear that

$v \sim \mathcal{N}(0, C)$. Therefore, we can write the objective function as

$$\begin{aligned}\mathbb{E}[\|v\|^2] &= \mathbb{E}[v^\top H v] \\ &= \mathbb{E}[(\sqrt{1-\alpha}u_1 + \sqrt{\alpha}Qu_2)^\top H(\sqrt{1-\alpha}u_1 + \sqrt{\alpha}Qu_2)] \\ &= (1-\alpha)\mathbb{E}[u_1^\top H u_1] + \alpha\mathbb{E}[u_2^\top Q^\top H Q u_2]\end{aligned}$$

where the first equation follows from the quadratic function assumption, and the last uses the fact $\mathbb{E}[u_1] = 0$ and $\mathbb{E}[u_2] = 0$. The first term on the rightmost side of the above does not involve Q , so

$$\begin{aligned}\arg \min \mathbb{E}[\|v\|^2] &= \arg \min \mathbb{E}[u_2^\top Q^\top H Q u_2] \\ &= \arg \min \mathbb{E}[\text{Tr}[Q^\top H Q u_2 u_2^\top]] \\ &= \arg \min \text{Tr}[Q^\top H Q \mathbb{E}[u_2 u_2^\top]] \\ &= \arg \min \text{Tr}[Q^\top H Q]\end{aligned}$$

where the last step uses the fact $\mathbb{E}[u_2 u_2^\top] = I_m$. The solution to the above minimization problem, under the constraint $Q^\top Q = I_m$, is well known, and is given by the eigenvectors corresponding to the m smallest eigenvalues of H . That is, Q corresponds to the m principal components of H^{-1} .

APPENDIX D PROOF OF THEOREM 1

We suppose that the number of rounds R is sufficiently large such that the step-size η defined in (11) satisfies the following conditions:

$$\eta \leq \frac{1}{8LK\sqrt{n+4}}, \quad (13)$$

$$\eta \leq \frac{1}{LK^{1.5}\sqrt{10M}}, \quad (14)$$

$$\eta \leq \frac{1-\alpha}{LK}, \quad (15)$$

$$\eta \leq \frac{M(1-\alpha)}{128L(n+4)}, \quad (16)$$

$$\eta \leq \frac{1}{MLK^2}, \quad (17)$$

$$\eta \leq \frac{1}{5LK}, \quad (18)$$

$$\eta \leq \frac{M}{4L(1-\alpha)}. \quad (19)$$

For simplicity, denote f^i as the local objective function of client i and \tilde{f}_r^i as its Gaussian smoothing at round r , i.e.,

$$f^i(x) = \mathbb{E}_{\xi \sim \mathcal{D}_i}[F(x; \xi)],$$

and

$$\tilde{f}_r^i(x) = \mathbb{E}_{v \sim \mathcal{N}(0, C_r)}[f^i(x + \mu v)].$$

We first state some properties of Gaussian smoothing under the ℓ_2 norm assumption.

Lemma 3. Let $h : \mathbb{R}^n \rightarrow \mathbb{R}$ be an L -smooth function w.r.t. the ℓ_2 norm $\|\cdot\| = \|\cdot\|_2$. Let g be a gradient estimation to its Gaussian smoothing $h_{\mu, C}$ given in (6) and assume the covariance matrix C is constrained in the form of (9). Then we have

$$\mathbb{E}[\|g\|^2] \leq \frac{\mu^2}{2}L^2(n+6)^3 + 2(n+4)\|\nabla h(x)\|^2$$

for all $x \in \mathbb{R}^n$. In addition, the bounds in (7) and (8) can be tightened as

$$\|\nabla h_{\mu, C}(x) - \nabla h(x)\| \leq \frac{L\mu}{2\sqrt{1-\alpha}}(n+3)^{1.5}, \quad (20)$$

and

$$|h_{\mu, C}(x) - h(x)| \leq \frac{nL\mu^2}{2}. \quad (21)$$

In the following we show that the client drifts are bounded.

Lemma 4. Under the same assumptions as in Theorem 1, we have

$$\begin{aligned}\sum_{k=0}^{K-1} \mathbb{E}[\|w_{r,k}^i - x_r\|^2] \\ \leq 2\eta^2 K^3(\psi + 8(n+4)\|\nabla f(x_r)\|^2),\end{aligned} \quad (22)$$

and

$$\sum_{k=0}^{K-1} \mathbb{E}[\|g_{r,k}^i\|^2] \leq 2K(\psi + 8(n+4)\|\nabla f(x_r)\|^2), \quad (23)$$

where the expectation is taken over all randomness involved in round r , and

$$\psi = \frac{1}{2}\mu^2 L^2(n+6)^3 + 2(n+4)(4\sigma_g^2 + \sigma_l^2). \quad (24)$$

The following bound shows that the global update of the solution gives a sufficient descent direction.

Lemma 5. Under the same assumptions as in Theorem 1, we have

$$\begin{aligned}\mathbb{E}[\langle \nabla f(x_r), x_{r+1} - x_r \rangle] &\leq -\frac{\eta K}{4}(1-\alpha)\|\nabla f(x_r)\|^2 \\ &\quad - \frac{\eta K}{2}(1-\alpha)\mathbb{E}\left[\left\|\frac{1}{NK}\sum_{i=1}^N\sum_{k=0}^{K-1}\nabla \tilde{f}_r^i(w_{r,k}^i)\right\|^2\right] \\ &\quad + 2L^2\eta^3 K^3\psi + \frac{\eta K L^2 \mu^2}{4(1-\alpha)}(n+3)^3,\end{aligned}$$

where ψ is a constant given in (24).

Lemma 6. With the same assumptions in Theorem 1, we have

$$\begin{aligned}\mathbb{E}[\|x_{r+1} - x_r\|^2] &\leq \frac{32\eta^2 K}{M}(n+4)\|\nabla f(x_r)\|^2 \\ &\quad + \eta^2 K^2 \frac{5L^2 \mu^2 (n+3)^3}{2(1-\alpha)} + \frac{4\eta^2 K}{M}\psi + 5\eta^2 K^2 \frac{\sigma_g^2}{M} \\ &\quad + \eta^2 \mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^N\sum_{k=0}^{K-1}\nabla \tilde{f}_r^i(w_{r,k}^i)\right\|^2\right]\end{aligned}$$

where ψ is a constant given in (24).

Proof of Theorem 1. Using the smoothness assumption, we have

$$\begin{aligned}\mathbb{E}[f(x_{r+1})] &\leq f(x_r) + \mathbb{E}[\langle \nabla f(x_r), x_{r+1} - x_r \rangle] \\ &\quad + \frac{L}{2}\mathbb{E}[\|x_{r+1} - x_r\|^2].\end{aligned}$$

The right-hand side can be bounded using Lemmas 5 and 6, which yields

$$\begin{aligned}
\mathbb{E}[f(x_{r+1})] &\leq f(x_r) - \frac{\eta K}{4}(1-\alpha) \|\nabla f(x_r)\|^2 \\
&\quad - \frac{\eta K}{2}(1-\alpha) \mathbb{E} \left[\left\| \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\
&\quad + 2L^2\eta^3 K^3 \psi + \frac{\eta K L^2 \mu^2}{4(1-\alpha)}(n+3)^3 \\
&\quad + \frac{16L\eta^2 K}{M}(n+4) \|\nabla f(x_r)\|^2 \\
&\quad + \frac{L}{2} \left(\eta^2 K^2 \frac{5L^2 \mu^2 (n+3)^3}{2(1-\alpha)} + \frac{4\eta^2 K}{M} \psi + 5\eta^2 K^2 \frac{\sigma_g^2}{M} \right) \\
&\quad + \frac{L}{2} \eta^2 \mathbb{E} \left[\left\| \frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\
&= f(x_r) - \eta K \left(\frac{1-\alpha}{4} - \frac{16L\eta}{M}(n+4) \right) \|\nabla f(x_r)\|^2 \\
&\quad - \frac{\eta}{2} \left(\frac{1-\alpha}{K} - L\eta \right) \mathbb{E} \left[\left\| \frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\
&\quad + 2L^2\eta^3 K^3 \psi + \frac{\eta K L^2 \mu^2}{4(1-\alpha)}(n+3)^3 \\
&\quad + \eta^2 K^2 \frac{5L^3 \mu^2 (n+3)^3}{4(1-\alpha)} + \frac{2L\eta^2 K}{M} \psi + \eta^2 K^2 \frac{5L\sigma_g^2}{2M} \\
&\stackrel{(15,16)}{\leq} f(x_r) - \eta K \frac{1-\alpha}{8} \|\nabla f(x_r)\|^2 \\
&\quad + \eta K \left(2L\eta \left(L\eta K^2 + \frac{1}{M} \right) \psi \right. \\
&\quad \left. + \frac{L^2 \mu^2 (n+3)^3}{4(1-\alpha)} (1 + 5L\eta K) + \eta K \frac{5L\sigma_g^2}{2M} \right) \\
&\stackrel{(17,18)}{\leq} f(x_r) - \eta K \frac{1-\alpha}{8} \|\nabla f(x_r)\|^2 \\
&\quad + \underbrace{\eta K \left(\frac{4}{M} L\eta \psi + \frac{L^2 \mu^2 (n+3)^3}{2(1-\alpha)} + \eta K \frac{5L\sigma_g^2}{2M} \right)}_{\mathfrak{E}}. \quad (25)
\end{aligned}$$

Now substituting ψ with (24) and using the condition $K \leq n$, we bound the term \mathfrak{E} as

$$\begin{aligned}
\mathfrak{E} &= 4L\eta \left(\frac{1}{2M} \mu^2 L^2 (n+6)^3 + \frac{2}{M} (n+4)(4\sigma_g^2 + \sigma_l^2) \right) \\
&\quad + \frac{L^2 \mu^2 (n+3)^3}{2(1-\alpha)} + \eta K \frac{5L\sigma_g^2}{2M} \\
&\leq \frac{2}{M} \eta \mu^2 L^3 (n+6)^3 + \frac{8}{M} L\eta (n+4)(4\sigma_g^2 + \sigma_l^2) \\
&\quad + \frac{L^2 \mu^2 (n+3)^3}{2(1-\alpha)} + \eta n \frac{5L\sigma_g^2}{2M} \\
&\leq \frac{2}{M} \eta \mu^2 L^3 (n+6)^3 + \frac{L^2 \mu^2 (n+3)^3}{2(1-\alpha)} \\
&\quad + \frac{9}{M} L\eta (n+4)(4\sigma_g^2 + \sigma_l^2)
\end{aligned}$$

$$\begin{aligned}
&\leq \mu^2 L^2 (n+6)^3 \left(\frac{2\eta L}{M} + \frac{1}{2(1-\alpha)} \right) \\
&\quad + \frac{9}{M} L\eta (n+4)(4\sigma_g^2 + \sigma_l^2) \\
&\stackrel{(19)}{\leq} \frac{\mu^2 L^2 (n+6)^3}{1-\alpha} + \frac{36}{M} L\eta (n+4)(\sigma_g^2 + \sigma_l^2).
\end{aligned}$$

Plugging the above into (25), we have

$$\begin{aligned}
\mathbb{E}[f(x_{r+1})] &\leq f(x_r) - \eta K \frac{1-\alpha}{8} \|\nabla f(x_r)\|^2 \\
&\quad + \eta K \left(\frac{\mu^2 L^2 (n+6)^3}{1-\alpha} + \frac{36}{M} L\eta (n+4)(\sigma_g^2 + \sigma_l^2) \right).
\end{aligned}$$

Summing over $r = 0, \dots, R-1$, taking total expectation, and using Assumption 4, we have

$$\begin{aligned}
\frac{1}{R} \sum_{r=0}^{R-1} \mathbb{E} [\|\nabla f(x_r)\|^2] &\leq \frac{8\mu^2 L^2 (n+6)^3}{(1-\alpha)^2} \\
&\quad + \frac{8}{1-\alpha} \left(\frac{f(x_0) - f_*}{R\eta K} + \frac{36}{M} L\eta (n+4)(\sigma_g^2 + \sigma_l^2) \right) \\
&\leq \frac{96}{1-\alpha} \sqrt{\frac{(f(x_0) - f_*)(n+4)(\sigma_g^2 + \sigma_l^2)}{RKM}} + \frac{8\mu^2 L^2 (n+6)^3}{(1-\alpha)^2}
\end{aligned}$$

where in the second inequality, we use the step-size setting (11). \square

APPENDIX E PROOF OF LEMMA 3

The proof follows [20, Theorem 4] with slight modifications due to the non-isotropic covariance matrix:

$$\begin{aligned}
\|g\|^2 &\stackrel{(6)}{=} \left\| \frac{h(x + \mu v) - h(x - \mu v)}{2\mu} v \right\|^2 \\
&= \frac{1}{4\mu^2} \|h(x + \mu v) - h(x - \mu v) - 2\langle \nabla h(x), \mu v \rangle v \\
&\quad + 2\langle \nabla h(x), \mu v \rangle v\|^2 \\
&\leq \frac{1}{2\mu^2} \|h(x + \mu v) - h(x - \mu v) - 2\langle \nabla h(x), \mu v \rangle v\|^2 \\
&\quad + \frac{1}{2\mu^2} \|2\langle \nabla h(x), \mu v \rangle v\|^2 \\
&= \frac{1}{2\mu^2} \|h(x + \mu v) - h(x - \mu v) - 2\langle \nabla h(x), \mu v \rangle v\|^2 \|v\|^2 \\
&\quad + 2\langle \nabla h(x), v \rangle^2 \|v\|^2 \\
&\stackrel{(12)}{\leq} \frac{1}{2\mu^2} (L\|\mu v\|^2)^2 \|v\|^2 + 2\langle \nabla h(x), v \rangle^2 \|v\|^2 \\
&= \frac{\mu^2 L^2}{2} \|v\|^6 + 2\langle \nabla h(x), v \rangle^2 \|v\|^2
\end{aligned}$$

where the first inequality uses the fact

$$\left\| \sum_{i=1}^k a_i \right\|^2 \leq k \sum_{i=1}^k \|a_i\|^2. \quad (26)$$

Now let $v = \sqrt{C}u$ where \sqrt{C} denotes the 0.5th power of C . We have

$$\begin{aligned}
\|g\|^2 &\leq \frac{\mu^2 L^2}{2} \|\sqrt{C}u\|^6 + 2\langle \sqrt{C}\nabla h(x), u \rangle^2 \|\sqrt{C}u\|^2 \\
&= \frac{\mu^2 L^2}{2} (u^\top C u)^3 + 2\langle \sqrt{C}\nabla h(x), u \rangle^2 (u^\top C u).
\end{aligned}$$

Recall that the covariance matrix in the form of (9) can be bounded as $(1 - \alpha)I_n \preceq C \preceq I_n$. We have $u^\top Cu \leq \|u\|^2$ and therefore

$$\|g\|^2 \leq \frac{\mu^2 L^2}{2} \|u\|^6 + 2(\sqrt{C} \nabla h(x), u)^2 \|u\|^2.$$

Taking expectation gives:

$$\begin{aligned} \mathbb{E}[\|g\|^2] &= \mathbb{E}_{u \sim \mathcal{N}(0, I_n)}[\|g\|^2] \\ &\leq \frac{\mu^2 L^2}{2} \mathbb{E}[\|u\|^6] + 2\mathbb{E}[(\sqrt{C} \nabla h(x), u)^2 \|u\|^2] \\ &\leq \frac{\mu^2 L^2}{2} (n+6)^3 + 2(n+4) \|\sqrt{C} \nabla h(x)\|^2 \end{aligned}$$

where the last inequality follows from [20, equations (17) and (32)]. Using the fact $\|\sqrt{C} \nabla h(x)\|^2 \leq \|h(x)\|^2$ derived from $C \preceq I_n$, we reach the first statement.

Under the ℓ_2 norm assumption, we have $\|v\|^2 = \|\sqrt{C}u\|^2 \leq \|u\|^2$ and $\|C^{-1}v\| = \|C^{-\frac{1}{2}}u\| \leq \frac{1}{\sqrt{1-\alpha}} \|u\|$. Plugging these bounds into (7) and (8) and using [20, Lemma 1], we can obtain (20) and (21), respectively.

APPENDIX F PROOF OF LEMMA 4

Applying Lemma 3 to $g_{r,k}^i$ (via replacing $h(x)$ with $F(w_{r,k}^i; \xi_{r,k}^i)$), we have

$$\begin{aligned} \mathbb{E}[\|g_{r,k}^i\|^2] &\leq \frac{\mu^2}{2} L^2 (n+6)^3 \\ &\quad + 2(n+4) \mathbb{E}[\|\nabla F(w_{r,k}^i; \xi_{r,k}^i)\|^2]. \end{aligned} \quad (27)$$

The stochastic gradient norm on the right-hand side can be bounded as

$$\begin{aligned} \mathbb{E}[\|\nabla F(w_{r,k}^i; \xi_{r,k}^i)\|^2] &\stackrel{(a)}{\leq} \mathbb{E}[\|\nabla f^i(w_{r,k}^i)\|^2] + \sigma_l^2 \\ &= \mathbb{E}[\|\nabla f^i(w_{r,k}^i) - \nabla f^i(x_r) + \nabla f^i(x_r)\|^2] + \sigma_l^2 \\ &\stackrel{(26)}{\leq} 2\|\nabla f^i(x_r)\|^2 + 2\mathbb{E}[\|\nabla f^i(w_{r,k}^i) - \nabla f^i(x_r)\|^2] + \sigma_l^2 \\ &\stackrel{(b)}{\leq} 2\|\nabla f^i(x_r)\|^2 + 2L^2 \mathbb{E}[\|w_{r,k}^i - x_r\|^2] + \sigma_l^2 \\ &= 2\|\nabla f^i(x_r) - \nabla f(x_r) + \nabla f(x_r)\|^2 \\ &\quad + 2L^2 \mathbb{E}[\|w_{r,k}^i - x_r\|^2] + \sigma_l^2 \\ &\stackrel{(26)}{\leq} 4\|\nabla f^i(x_r) - \nabla f(x_r)\|^2 + 4\|\nabla f(x_r)\|^2 \\ &\quad + 2L^2 \mathbb{E}[\|w_{r,k}^i - x_r\|^2] + \sigma_l^2 \\ &\stackrel{(c)}{\leq} 4\sigma_g^2 + 4\|\nabla f(x_r)\|^2 + 2L^2 \mathbb{E}[\|w_{r,k}^i - x_r\|^2] + \sigma_l^2 \end{aligned}$$

where (a) follows from Assumption 2, (b) is due to Assumption 1, and (c) is due to Assumption 3. Plugging the above

into (27) and summing over $k = 0, \dots, K-1$, one obtains

$$\begin{aligned} \sum_{k=0}^{K-1} \mathbb{E}[\|g_{r,k}^i\|^2] &\leq 8K(n+4) \|\nabla f(x_r)\|^2 \\ &\quad + K \left(\frac{\mu^2}{2} L^2 (n+6)^3 + 2(n+4)(4\sigma_g^2 + \sigma_l^2) \right) \\ &\quad + 4L^2(n+4) \sum_{k=0}^{K-1} \mathbb{E}[\|w_{r,k}^i - x_r\|^2] \\ &= K(\psi + 8(n+4) \|\nabla f(x_r)\|^2) \\ &\quad + 4L^2(n+4) \sum_{k=0}^{K-1} \mathbb{E}[\|w_{r,k}^i - x_r\|^2]. \end{aligned} \quad (28)$$

On the other hand, by construction and the fact $w_{r,0}^i = x_r$, we have

$$\begin{aligned} \sum_{k=0}^{K-1} \mathbb{E}[\|w_{r,k}^i - x_r\|^2] &= \sum_{k=0}^{K-1} \mathbb{E} \left[\left\| \sum_{j=0}^{k-1} w_{r,j+1}^i - w_{r,j}^i \right\|^2 \right] \\ &= \sum_{k=0}^{K-1} \mathbb{E} \left[\left\| \sum_{j=0}^{k-1} \eta g_{r,j}^i \right\|^2 \right] \\ &\stackrel{(26)}{\leq} \eta^2 \sum_{k=0}^{K-1} k \sum_{j=0}^{k-1} \mathbb{E}[\|g_{r,j}^i\|^2] \\ &\leq \eta^2 \sum_{k=0}^{K-1} K \sum_{j=0}^{K-1} \mathbb{E}[\|g_{r,j}^i\|^2] \\ &= \eta^2 K^2 \sum_{k=0}^{K-1} \mathbb{E}[\|g_{r,k}^i\|^2] \\ &\stackrel{(28)}{\leq} \eta^2 K^3 (\psi + 8(n+4) \|\nabla f(x_r)\|^2) \\ &\quad + 4\eta^2 K^2 L^2 (n+4) \sum_{k=0}^{K-1} \mathbb{E}[\|w_{r,k}^i - x_r\|^2] \\ &\stackrel{(13)}{\leq} \eta^2 K^3 (\psi + 8(n+4) \|\nabla f(x_r)\|^2) \\ &\quad + \frac{1}{2} \sum_{k=0}^{K-1} \mathbb{E}[\|w_{r,k}^i - x_r\|^2]. \end{aligned}$$

Rearranging the terms, one obtains (22). The bound in (23) can be obtained by putting (22) back into (28).

APPENDIX G PROOF OF LEMMA 5

By construction, we write the per-round descent step as

$$\begin{aligned} \mathbb{E}[x_{r+1} - x_r] &= -\mathbb{E} \left[\frac{1}{M} \sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} \eta g_{r,k}^i \right] \\ &= -\eta \mathbb{E} \left[\frac{1}{M} \sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} C_r \nabla \tilde{f}_r^i(w_{r,k}^i) \right] \end{aligned}$$

$$= -\eta C_r \frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \mathbb{E} \left[\nabla \tilde{f}_r^i(w_{r,k}^i) \right]$$

where the second equation follows from Lemma 1 and the third equation is due to the uniform sampling of the client set. It follows that

$$\begin{aligned} & \mathbb{E} [\langle \nabla f(x_r), x_{r+1} - x_r \rangle] \\ &= -\eta K \mathbb{E} \left[\left\langle \nabla f(x_r), C_r \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\rangle \right] \\ &= -\eta K \mathbb{E} \left[\left\langle \sqrt{C_r} \nabla f(x_r), \sqrt{C_r} \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\rangle \right] \\ &= -\frac{\eta K}{2} \left\| \sqrt{C_r} \nabla f(x_r) \right\|^2 \\ &\quad - \frac{\eta K}{2} \mathbb{E} \left[\left\| \sqrt{C_r} \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\ &\quad + \frac{\eta K}{2} \mathbb{E} \left[\left\| \sqrt{C_r} \nabla f(x_r) - \sqrt{C_r} \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\ &\leq -\frac{\eta K}{2} (1 - \alpha) \|\nabla f(x_r)\|^2 \\ &\quad - \frac{\eta K}{2} (1 - \alpha) \mathbb{E} \left[\left\| \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\ &\quad + \underbrace{\frac{\eta K}{2} \mathbb{E} \left[\left\| \nabla f(x_r) - \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right]}_{\mathfrak{A}}, \end{aligned}$$

where the third equation uses the identity $\langle a, b \rangle = \frac{1}{2} (\|a\|^2 + \|b\|^2 - \|a - b\|^2)$ and the inequality uses the bound on the covariance matrix $(1 - \alpha)I_n \preceq C_r \preceq I_n$.

Using the fact $f(x) = \frac{1}{N} \sum_{i=1}^N f^i(x)$, we can bound \mathfrak{A} in (29) as

$$\begin{aligned} \mathfrak{A} &= \mathbb{E} \left[\left\| \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla f^i(x_r) - \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\ &\leq \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \mathbb{E} \left[\left\| \nabla f^i(x_r) - \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\ &= \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \mathbb{E} \left[\left\| \nabla f^i(x_r) - \nabla f^i(w_{r,k}^i) \right. \right. \\ &\quad \left. \left. + \nabla f^i(w_{r,k}^i) - \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\ &\stackrel{(26)}{\leq} \frac{2}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \mathbb{E} \left[\left\| \nabla f^i(x_r) - \nabla f^i(w_{r,k}^i) \right\|^2 \right] \\ &\quad + \frac{2}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \mathbb{E} \left[\left\| \nabla f^i(w_{r,k}^i) - \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\ &\stackrel{(a)}{\leq} \frac{2L^2}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \mathbb{E} \left[\|x_r - w_{r,k}^i\|^2 \right] \\ &\quad + \frac{2}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \mathbb{E} \left[\left\| \nabla f^i(w_{r,k}^i) - \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \end{aligned}$$

$$\begin{aligned} &\stackrel{(20)}{\leq} \frac{2L^2}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \mathbb{E} \left[\|x_r - w_{r,k}^i\|^2 \right] + \frac{L^2 \mu^2}{2(1 - \alpha)} (n + 3)^3 \\ &\stackrel{(22)}{\leq} 4L^2 \eta^2 K^2 (\psi + 8(n + 4) \|\nabla f(x_r)\|^2) \\ &\quad + \frac{L^2 \mu^2}{2(1 - \alpha)} (n + 3)^3, \end{aligned}$$

where (a) uses the smoothness assumption. Plugging the above into (29), we have

$$\begin{aligned} \mathbb{E} [\langle \nabla f(x_r), x_{r+1} - x_r \rangle] &\leq -\frac{\eta K}{2} (1 - \alpha) \|\nabla f(x_r)\|^2 \\ &\quad - \frac{\eta K}{2} (1 - \alpha) \mathbb{E} \left[\left\| \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\ &\quad + 2L^2 \eta^3 K^3 (\psi + 8(n + 4) \|\nabla f(x_r)\|^2) \\ &\quad + \frac{\eta K L^2 \mu^2}{4(1 - \alpha)} (n + 3)^3 \\ &= -\frac{\eta K}{2} (1 - \alpha - 32(n + 4) \eta^2 K^2 L^2) \|\nabla f(x_r)\|^2 \\ &\quad - \frac{\eta K}{2} (1 - \alpha) \mathbb{E} \left[\left\| \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\ &\quad + 2L^2 \eta^3 K^3 \psi + \frac{\eta K L^2 \mu^2}{4(1 - \alpha)} (n + 3)^3 \\ &\stackrel{(13)}{\leq} -\frac{\eta K}{4} (1 - \alpha) \|\nabla f(x_r)\|^2 \\ &\quad - \frac{\eta K}{2} (1 - \alpha) \mathbb{E} \left[\left\| \frac{1}{NK} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right] \\ &\quad + 2L^2 \eta^3 K^3 \psi + \frac{\eta K L^2 \mu^2}{4(1 - \alpha)} (n + 3)^3. \end{aligned}$$

The proof then completes.

APPENDIX H PROOF OF LEMMA 6

By construction, we have

$$\begin{aligned} x_{r+1} - x_r &= \eta \frac{1}{M} \sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} g_{r,k}^i \\ &= \eta \frac{1}{M} \sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} \underbrace{\left(g_{r,k}^i - \nabla \tilde{f}_r^i(w_{r,k}^i) \right)}_{\mathfrak{B}} \\ &\quad + \underbrace{\eta \left(\frac{1}{M} \sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) - \frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right)}_{\mathfrak{C}} \\ &\quad + \eta \frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i). \end{aligned}$$

By the property of the Gaussian smoothing, we have $\mathbb{E}[\mathfrak{B}] = 0$. In addition, since the clients are uniformly sampled, we

have $\mathbb{E}[\mathfrak{C}] = 0$, and therefore obtain

$$\begin{aligned} \mathbb{E}[\|x_{r+1} - x_r\|^2] &= \eta^2 \mathbb{E}[\|\mathfrak{B}\|^2] + \eta^2 \mathbb{E}[\|\mathfrak{C}\|^2] \\ &\quad + \eta^2 \mathbb{E}\left[\left\|\frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i)\right\|^2\right]. \end{aligned} \quad (29)$$

The term \mathfrak{B} can be bounded as

$$\begin{aligned} \mathbb{E}[\|\mathfrak{B}\|^2] &\stackrel{(a)}{=} \frac{1}{M^2} \mathbb{E}\left[\sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} \left\|g_{r,k}^i - \nabla \tilde{f}_r^i(w_{r,k}^i)\right\|^2\right] \\ &\stackrel{(b)}{\leq} \frac{1}{M^2} \mathbb{E}\left[\sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} \|g_{r,k}^i\|^2\right] \\ &\stackrel{(c)}{=} \frac{1}{MN} \mathbb{E}\left[\sum_{i=1}^N \sum_{k=0}^{K-1} \|g_{r,k}^i\|^2\right] \\ &\stackrel{(23)}{\leq} \frac{2K}{M} \left(\psi + 8(n+4) \|\nabla f(x_r)\|^2\right), \end{aligned} \quad (30)$$

where (a) uses the fact $\mathbb{E}[g_{r,k}^i] = \nabla \tilde{f}_r^i(w_{r,k}^i)$, (b) uses the fact $\mathbb{E}[\|a - \mathbb{E}[a]\|^2] \leq \mathbb{E}[\|a\|^2]$, and (c) is due to the uniform sampling of the clients.

The term \mathfrak{C} can be written as

$$\begin{aligned} \mathfrak{C} &= \underbrace{\frac{1}{M} \sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} (\nabla \tilde{f}_r^i(w_{r,k}^i) - \nabla f^i(w_{r,k}^i))}_{\mathfrak{D}_1} \\ &\quad + \underbrace{\frac{1}{M} \sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} (\nabla f^i(w_{r,k}^i) - \nabla f^i(x_r))}_{\mathfrak{D}_2} \\ &\quad + \underbrace{\frac{1}{M} \sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} \nabla f^i(x_r) - \frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla f^i(x_r)}_{\mathfrak{D}_3} \\ &\quad + \underbrace{\frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} (\nabla f^i(x_r) - \nabla f^i(w_{r,k}^i))}_{\mathfrak{D}_4} \\ &\quad + \underbrace{\frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} (\nabla f^i(w_{r,k}^i) - \nabla \tilde{f}_r^i(w_{r,k}^i))}_{\mathfrak{D}_5}, \end{aligned}$$

and hence, we have

$$\begin{aligned} \mathbb{E}[\|\mathfrak{C}\|^2] &\stackrel{(26)}{\leq} 5\mathbb{E}[\|\mathfrak{D}_1\|^2] + 5\mathbb{E}[\|\mathfrak{D}_2\|^2] + 5\mathbb{E}[\|\mathfrak{D}_3\|^2] \\ &\quad + 5\mathbb{E}[\|\mathfrak{D}_4\|^2] + 5\mathbb{E}[\|\mathfrak{D}_5\|^2]. \end{aligned} \quad (31)$$

Using (26) and Jensen's inequality, we have

$$\begin{aligned} \mathbb{E}[\|\mathfrak{D}_1\|^2] &\leq \frac{K}{M} \mathbb{E}\left[\sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} \left\|\nabla \tilde{f}_r^i(w_{r,k}^i) - \nabla f^i(w_{r,k}^i)\right\|^2\right] \\ &\stackrel{(20)}{\leq} \frac{K^2 L^2 \mu^2 (n+3)^3}{4(1-\alpha)}. \end{aligned}$$

In addition, we obtain

$$\begin{aligned} \mathbb{E}[\|\mathfrak{D}_2\|^2] &\stackrel{(26)}{\leq} \frac{K}{M} \mathbb{E}\left[\sum_{i \in \mathcal{W}_r} \sum_{k=0}^{K-1} \left\|\nabla f^i(w_{r,k}^i) - \nabla f^i(x_r)\right\|^2\right] \\ &\stackrel{(a)}{=} \frac{K}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \mathbb{E}\left[\left\|\nabla f^i(w_{r,k}^i) - \nabla f^i(x_r)\right\|^2\right] \\ &\stackrel{(b)}{\leq} \frac{KL^2}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \mathbb{E}\left[\|w_{r,k}^i - x_r\|^2\right] \\ &\stackrel{(22)}{\leq} 2\eta^2 K^4 L^2 (\psi + 8(n+4) \|\nabla f(x_r)\|^2) \end{aligned}$$

where (a) is due to the uniform sampling of clients and (b) uses the smoothness assumption.

We can derive in the same way the bounds for \mathfrak{D}_4 and \mathfrak{D}_5 :

$$\mathbb{E}[\|\mathfrak{D}_4\|^2] \leq 2\eta^2 K^4 L^2 (\psi + 8(n+4) \|\nabla f(x_r)\|^2)$$

and

$$\mathbb{E}[\|\mathfrak{D}_5\|^2] \leq \frac{K^2 L^2 \mu^2 (n+3)^3}{4(1-\alpha)}.$$

The term \mathfrak{D}_3 can be bounded as

$$\begin{aligned} \mathbb{E}[\|\mathfrak{D}_3\|^2] &\stackrel{(26)}{\leq} \sum_{k=0}^{K-1} K \mathbb{E}\left[\left\|\frac{1}{M} \sum_{i \in \mathcal{W}_r} \nabla f^i(x_r) - \nabla f(x_r)\right\|^2\right] \\ &= K^2 \mathbb{E}\left[\frac{1}{M^2} \sum_{i \in \mathcal{W}_r} \left\|\nabla f^i(x_r) - \nabla f(x_r)\right\|^2\right] \\ &\leq \frac{K^2 \sigma_g^2}{M} \end{aligned}$$

where the equation follows from the fact $\mathbb{E}[\nabla f^i(x_r)] = \nabla f(x_r)$ (due to the uniform sampling of clients), and the second inequality uses Assumption 3.

Putting the above bounds on \mathfrak{D}_1 to \mathfrak{D}_5 into (31) gives

$$\begin{aligned} \mathbb{E}[\|\mathfrak{C}\|^2] &\leq K^2 \frac{5L^2 \mu^2 (n+3)^3}{2(1-\alpha)} + 20L^2 \eta^2 K^4 \psi \\ &\quad + 160L^2 \eta^2 K^4 (n+4) \|\nabla f(x_r)\|^2 + 5K^2 \frac{\sigma_g^2}{M}. \end{aligned}$$

Plugging the above and (30) into (29), we obtain

$$\begin{aligned} \mathbb{E}[\|x_{r+1} - x_r\|^2] &\leq \frac{2K\eta^2}{M} \left(\psi + 8(n+4) \|\nabla f(x_r)\|^2\right) \\ &\quad + \eta^2 K^2 \frac{5L^2 \mu^2 (n+3)^3}{2(1-\alpha)} + 20L^2 \eta^4 K^4 \psi \\ &\quad + 160L^2 \eta^4 K^4 (n+4) \|\nabla f(x_r)\|^2 \\ &\quad + 5\eta^2 K^2 \frac{\sigma_g^2}{M} + \eta^2 \mathbb{E}\left[\left\|\frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i)\right\|^2\right] \\ &= 16\eta^2 K \left(\frac{1}{M} + 10L^2 \eta^2 K^3\right) (n+4) \|\nabla f(x_r)\|^2 \\ &\quad + \eta^2 K^2 \frac{5L^2 \mu^2 (n+3)^3}{2(1-\alpha)} + 2\eta^2 K \left(\frac{1}{M} + 10L^2 \eta^2 K^3\right) \psi \\ &\quad + 5\eta^2 K^2 \frac{\sigma_g^2}{M} + \eta^2 \mathbb{E}\left[\left\|\frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i)\right\|^2\right] \end{aligned}$$

$$\begin{aligned}
&\leq \frac{32\eta^2 K}{M} (n+4) \|\nabla f(x_r)\|^2 + \eta^2 K^2 \frac{5L^2 \mu^2 (n+3)^3}{2(1-\alpha)} \\
&\quad + \frac{4\eta^2 K}{M} \psi + 5\eta^2 K^2 \frac{\sigma_g^2}{M} \\
&\quad + \eta^2 \mathbb{E} \left[\left\| \frac{1}{N} \sum_{i=1}^N \sum_{k=0}^{K-1} \nabla \tilde{f}_r^i(w_{r,k}^i) \right\|^2 \right]
\end{aligned}$$

where the last step uses the condition (14). The proof completes.

REFERENCES

- [1] W. Fang, Z. Yu, Y. Jiang, Y. Shi, C. N. Jones, and Y. Zhou, "Communication-efficient stochastic zeroth-order optimization for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 5058–5073, 2022.
- [2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.
- [3] H. Ma, H. Guo, and V. K. Lau, "Communication-efficient federated multitask learning over wireless networks," *IEEE Internet of Things Journal*, vol. 10, no. 1, pp. 609–624, 2022.
- [4] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in *IEEE INFOCOM 2022-IEEE conference on computer communications*. IEEE, 2022, pp. 1739–1748.
- [5] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan, "Fedala: Adaptive local aggregation for personalized federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 11 237–11 244.
- [6] Y. Shi, P. Fan, Z. Zhu, C. Peng, F. Wang, and K. B. Letaief, "Sam: An efficient approach with selective aggregation of models in federated learning," *IEEE Internet of Things Journal*, 2024.
- [7] T. Che, J. Liu, Y. Zhou, J. Ren, J. Zhou, V. S. Sheng, H. Dai, and D. Dou, "Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization," *arXiv preprint arXiv:2310.15080*, 2023.
- [8] Z. Lin, Y. Sun, Y. Shi, X. Wang, L. Huang, L. Shen, and D. Tao, "Efficient federated prompt tuning for black-box large pre-trained models," *arXiv preprint arXiv:2310.03123*, 2023.
- [9] J. Sun, Z. Xu, H. Yin, D. Yang, D. Xu, Y. Chen, and H. R. Roth, "Fedbpt: Efficient federated black-box prompt tuning for large language models," *arXiv preprint arXiv:2310.01467*, 2023.
- [10] H. Zhao, W. Du, F. Li, P. Li, and G. Liu, "Fedprompt: Communication-efficient and privacy-preserving prompt tuning in federated learning," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [11] Y. Zhang, P. Li, J. Hong, J. Li, Y. Zhang, W. Zheng, P.-Y. Chen, J. D. Lee, W. Yin, M. Hong *et al.*, "Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: A benchmark," *arXiv preprint arXiv:2402.11592*, 2024.
- [12] K. Gao and O. Sener, "Generalizing Gaussian smoothing for random search," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 7077–7101.
- [13] X. He, Z. Zheng, Z. Chen, and Y. Zhou, "Adaptive evolution strategies for stochastic zeroth-order optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 5, pp. 1271–1285, 2022.
- [14] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM journal on optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.
- [15] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, 2020.
- [16] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
- [17] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv e-prints*, pp. arXiv–1708, 2017.
- [18] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *The Journal of Machine Learning Research*, vol. 5, pp. 361–397, Dec. 2004.
- [19] K. Balasubramanian and S. Ghadimi, "Zeroth-order Nonconvex Stochastic Optimization: Handling Constraints, High-Dimensionality and Saddle-Points," *Foundations of Computational Mathematics* volume, vol. 22, pp. 35–76, Feb. 2022.
- [20] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, 2017.
- [21] R. Bollapragada and S. M. Wild, "Adaptive sampling quasi-Newton methods for zeroth-order stochastic optimization," *Mathematical Programming Computation*, Mar. 2023.
- [22] H. Zhang, H. Xiong, and B. Gu, "Zeroth-Order Negative Curvature Finding: Escaping Saddle Points without Gradients," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 332–38 344, Dec. 2022.
- [23] H. Ye, Z. Huang, C. Fang, C. J. Li, and T. Zhang, "Hessian-Aware Zeroth-Order Optimization for Black-Box Adversarial Attack," *arXiv:1812.11377 [cs, stat]*, Mar. 2019.
- [24] N. Maheswaranathan, L. Metz, G. Tucker, D. Choi, and J. Sohl-Dickstein, "Guided evolutionary strategies: Augmenting random search with surrogate gradients," in *International Conference on Machine Learning*. PMLR, 2019, pp. 4264–4273.
- [25] F.-Y. Liu, Z.-N. Li, and C. Qian, "Self-guided evolution strategies with historical estimated gradients," in *IJCAI*, 2020, pp. 1474–1480.
- [26] A. Maritan, S. Dey, and L. Schenato, "Fedzen: Towards superlinear zeroth-order federated learning via incremental hessian estimation," *arXiv preprint arXiv:2309.17174*, 2023.
- [27] Y. Qiu, U. Shanbhag, and F. Yousefian, "Zeroth-order methods for nondifferentiable, nonconvex, and hierarchical federated optimization," *Advances in Neural Information Processing Systems*, vol. 36, 2023.
- [28] Y. Shu, X. Lin, Z. Dai, and B. K. H. Low, "Federated zeroth-order optimization using trajectory-informed surrogate gradients," *arXiv preprint arXiv:2308.04077*, 2023.