A Derivative-Incorporated Adaptive Gradient Method for Federated Learning

Huimin Gao 1, Qingtao Wu 1, Hongyan Cao 2*, Xuhui Zhao 1, Junlong Zhu 1 and Mingchuan Zhang 1©

1 College of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China; gaomnli@haust.edu.cn (H.G.); wqt8921@haust.edu.cn (Q.W.); zxh@haust.edu.cn (X.Z.); jljzh@haust.edu.cn (J.Z.); zhang_mch@haust.edu.cn (M.Z.)
2 China Research Institute of Radiowave Propagation, Qingdao 266107, China
* Correspondence: caohy@crirp.ac.cn

Abstract: As a new machine learning technique, federated learning has received more attention in recent years, which enables decentralized model training across data silos or edge intelligent devices in the Internet of Things without exchanging local raw data. All kinds of algorithms are proposed to solve the challenges in federated learning. However, most of these methods are based on stochastic gradient descent, which undergoes slow convergence and unstable performance during the training stage. In this paper, we propose a differential adaptive federated optimization method, which incorporates an adaptive learning rate and the gradient difference into the iteration rule of the global model. We further adopt the first-order moment estimation to compute the approximate value of the differential term so as to avoid amplifying the random noise from the input data sample. The theoretical convergence guarantee is established for our proposed method in a stochastic non-convex setting under full client participation and partial client participation cases. Experiments for the image classification task are performed on two standard datasets by training a neural network model, and experiment results on different baselines demonstrate the effectiveness of our proposed method.

Keywords: federated learning; stochastic gradient descent; learning rate; differential

MSC: 68T07

1. Introduction

With huge amounts of data being generated at an increasing rate, the cost of collecting and handling data is becoming more expensive by a centralized data center. Therefore, collaborative learning [1,2] has gained more attention due to its ability to unite the collective power of devices distributed in different geographic locations. However, the exchange of information among devices may cause the risk of data leakage during the process of training. As a promising solution, federated learning [3–5] was proposed to train models across edge smart devices in the Internet of Things without exchanging local raw data, which can balance the need for data privacy and the benefits of collaborative machine learning in a distributed environment. In the federated learning setting, the central server coordinates multiple clients to jointly train a global model from amounts of dispersive local data on different clients with periodic model aggregation. Since there is no exchange and sharing of data between clients and the central server during the training process, federated learning protects the privacy of clients to some extent. Furthermore, the communication manner with the server after multiple local update steps reduces the communication frequency and saves the limited communication resources. Benefitting from the decentralized learning based on local data, federated learning has been widely applied in a mobile edge network [6], healthcare [7], vehicles [8], and other fields.

Federated Averaging (FedAvg) [4] is one of the most prevailing algorithms for federated learning where a subset of clients conduct multiple stochastic gradient update steps.
based on their local data, and then the updates are sent back to the centralized server and averaged. FedAvg degenerates into the local stochastic gradient descent algorithm [9] when the data on different clients follow the same distribution. Despite gaining lots of attention, the practical implementation of FedAvg faces many challenges. The algorithm suffers from model drift, which hinders it converging to the optimal value of the objective function. The work of [10] generalized FedAvg and proposed FedProx to tackle the statistical and systems heterogeneity problems. In [11], the authors introduced control variates to adjust the model bias across clients, which was caused by the non-identically distributed data, but this required clients to remain stateful, which is unpractical. In addition, several works introduced the momentum technique to improve the convergence performance of stochastic gradient-based methods, which used the accumulated gradient information of previous iterations in the current step. Liu et al. [12] proposed a momentum federated learning method in a deterministic setting. The literature [13] applied a discounted global momentum to embed local iterations so as to accelerate learning and control model drift. In fact, the accumulated stale information may slow down the response to changes toward the update direction, which leads to the models continuing to update in the wrong direction, even though the gradient in the current step is already pointing in the right direction. This phenomenon is called overshoot in the domain of control, which manifests as oscillation and consumes more training time and resources.

Proportional–integral–derivative (PID) control is one of the most widely used control methods in the field of industrial control. The basic idea of PID control is making use of the current, past, and future trend information between the actual system output and the desired output. In order to settle the overshoot problem based on the stochastic gradient method, the study [14] that originated from the PID method proposed an integral–derivative approach which introduces the derivative or difference of the gradient in the training process. And the study [15] proposed a complete PID optimizer to stabilize the training process. On the other hand, due to the advantages of less parameter adjustment and better convergence speed, adaptive-type methods and variants [16–18] have been used in federated learning settings and obtained good results. Inspired by these observations, we focus on both the descent direction and the learning rate for performance improvement in the design of an optimization algorithm based on stochastic gradient. We introduce the derivative information to the update rule of a federated model, which represents the future trend information of the gradient change. With reference to the ongoing debate on model-centric AI and data-centric AI, our work belongs to the category of model-centric AI, highlighting the significance of further developing model-centric approaches [19]. In a nutshell, the contributions are elaborated on as follows.

• We develop a federated optimization method, named FedDA, which utilizes the trend of gradient change and adaptive learning rate to update the federated global model.

• We provide the convergence analysis for the proposed algorithm. It achieves the convergence rate of $O(1/\sqrt{nTH})$ for full client participation and $O(1/\sqrt{Tm})$ for partial client participation in a stochastic non-convex setting, where $n$, $m$ represent the number of clients, $H$ is the number of local updates and $T$ denotes the number of communications.

• We conduct the image classification tasks on a standard dataset to compare FedDA and other baseline methods. The experiment results show the effectiveness of our proposed algorithm.

The rest of this paper is organized as follows. We review the related literature in Section 2. Next, the optimization problem and algorithm design are presented in Section 3. Then, the helpful assumptions and theoretical results are displayed in Section 4. The convergence guarantee is built in detailed in Section 5. We compare the proposed algorithm with several baseline methods by extensive experiments in Section 6. We conclude the article in Section 7.

Notation. In this paper, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$ for a positive integer $n$ and $S_i$ is the clients set with size $|S_i| = m$ for $m \in (0, n]$ in the communication round.
We use $\| \cdot \|_1$ and $\| \cdot \|$ to represent the $\ell_1$ and $\ell_2$ norm, respectively. Let $[ \cdot ]_{\ell_1}$ denote the model parameter of the $h$-th local update step of the $t$-th communication round for client $i$.

2. Related Work

Stochastic gradient descent (SGD) [20] has been extensively applied in training large-scale machine learning models because of only choosing a mini-batch data sample or even a sample for each update round for the past few years. In order to reduce the influence of noise on stochastic gradient, the momentum technique [21,22] was introduced to correct the gradient direction. The literature [23] trained deep recurrent neural networks with momentum and strong initialization. Although its advantages include its simplicity and ease of implementation, the stochastic gradient-type methods depend on the selection of hyperparameters and show a heavy-tailed behavior from empirical observations. Adaptive optimization methods and variants have been studied and applied for their good practical performance in neural network training, such as Adagrad [24], Adam [16] and AMSGrad [25]. Xie et al. [26] proposed a new variant of Adam by reformulating the Nesterov update schemes for accelerate deep neural network training. These methods generally utilize the adaptive learning rate and momentum technique, which adaptively adjust learning rates for each coordinate of model parameters and enjoy excellent convergence performance. Recently, Recht [27] has revealed the connection between PID feedback control and stochastic optimization algorithms, and the former can be viewed as an optimizer that encapsulates gradient and momentum. An et al. [14] proposed a PID controller approach to solve the overshoot problem, which added the gradient difference in the current update step to predict the change trend of the updating direction. The work of [15] proposed a complete PID algorithm by increasing the effect of current information to relieve the oscillations during the training stage. The idea of adaptive PID control [28] permeates into the stochastic optimization by taking advantage of the current, the past and predicted future information of the gradient throughout the training process. These methods are implemented in a centralized manner.

As the size of the data samples increases, distributed training has gained more attention. Despite the great advantage, it incurs privacy issues and a communication bottleneck caused by the bandwidth limits and network delays. FedAvg [4] was proposed by incorporating a local gradient update and model averaging periodically, significantly reducing the communication overhead. After that, many variants of FedAvg were developed and proposed. The convergent upper bound of FedAvg was established by [9] and improved later in [29] under different data distribution cases. Li et al. [10] introduced a proximal term and proposed the FedProx framework to solve the heterogeneity challenge. Karimireddy et al. [11] made use of the control variable to adjust the model bias across clients. Guo et al. [30] presented hybrid local SGD by means of dividing the clients into different groups to achieve a good balance between model accuracy and runtime. Most of these works used stochastic gradient descent to update local models. Some momentum-based approaches are extended to the federated learning setting. Nesterov momentum [31] was used to accelerate learning when the objective function is convex. The work of [32] added the momentum to both the server model aggregation and local model update in order to reduce high variance, which improved the iteration complexity of federated learning. More recently, the studies incorporating momentum and an adaptive technique have emerged in a federated learning setting. FedAdam [17] was first proposed as a federated version of the adaptive gradient-type method with satisfactory performance. Several adaptive gradients methods with calibrations were studied by [33]. Local AMSGrad [34] considers the adaptiveness in the local model, while FedCAMS [18] utilizes the adaptive gradient in the update of the global model. FedExp [35] was developed by using varying pseudo-gradients to adaptively decide the step size of the server. Inspired by the above studies, in this article, we introduce the gradient difference and adaptive learning rate to the server side during the update of the global model and analyze its convergence performance.
3. Problem Formulation and Algorithm Design

A typical federated learning system with \(n\) clients or devices is considered where a global model is jointly trained under the orchestration of a central server by decentralized data. The main goal for \(n\) devices in a federated learning system is to settle an optimization problem, which can be formulated as:

\[
\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x),
\]

where \(f_i(x) \triangleq \mathbb{E}_{\xi_i \sim D_i} f_i(x; \xi_i)\) denotes the local cost function for the \(i\)-th client, which assesses the average discrepancy between the learning model and the ground truth associated with a random data sample \(\xi_i\) that draws distribution \(D_i\). \(n\) is the number of decentralized clients, and \(d\) denotes the dimensionality of the model parameters. In practice, the dataset belonging to each client is often generated based on its own local environment, which means the distribution \(D_i\) could be different, i.e., \(D_i \neq D_j\), when \(i \neq j\).

In this section, we present a differential adaptive federated optimization algorithm (FedDA) to solve problem (1), where the central server updates the global model by adopting the differential of the pseudo-gradient and an adaptive learning rate. The pseudocode of FedDA is summarized in detailed in Algorithm 1. Specifically, at the beginning of communication round \(t\), the centralized server randomly selects a subset \(S_t\) of clients and sends the current global model \(x_t\) to each client \(i\), \(i \in S_t\). After conducting \(H\) stochastic gradient descent steps with the local learning rate as line 6 in Algorithm 1, each client \(i\) acquires the local model parameters \(x_t^{(i)}\) and sends its model update to the central server after computing the model difference \(\Gamma_{i,t}\), i.e., \(\Gamma_{i,t} = x_t^{(i)} - x_t\), where \(x_t\) is the global model. In fact, by recursion on the local update rule in line 6, \(\Gamma_{i,t}\) is the cumulative gradient for the client \(i\). Hence, the aggregated difference \(\Gamma_t\) as defined in line 10 of Algorithm 1 can be regarded as the pseudo-gradient to update the global model in the server. Then, based on \(\Gamma_t\), the central server computes the momentum term \(\mathbf{m}_t\) and \(\mathbf{v}_t\) following AMSGrad in [25] where \(\nu_t\) is designed to be non-decreasing, avoiding the divergence of adaptive methods, and \(\beta_1\) and \(\beta_1\) are momentum factors.

**Algorithm 1** Federated Differential Adaptive Optimization Algorithm (FedDA)

**Input:** Initial point \(x_0\), learning rate \(\eta\) and \(\eta\), \(\mathbf{m}_0 = 0, \mathbf{v}_0 = 0, \mathbf{d}_0 = 0\).

1. for \(t = 1, \ldots, T\) do
2.   for each client \(i \in S_t\) in parallel do
3.     Set \(x_t^{(i)} = x_t\)
4.     for \(h = 0, \ldots, H - 1\) do
5.       Compute stochastic gradient \(g_t^{(i)} = \nabla F_i(x_t^{(i)}, \xi_t^{(i)})\)
6.       Local update: \(x_t^{(i)+1} = x_t^{(i)} - \eta_t g_t^{(i)}\)
7.     end for
8.     Let \(\Gamma_{i,t} = x_t^{(i)} - x_t\) and upload \(\Gamma_{i,t}\) to the server
9. end for
10. Compute \(\Gamma_t = \frac{1}{|S_t|} \sum_{i \in S_t} \Gamma_{i,t}\) in the server
11. Update \(\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \Gamma_t\)
12. Update \(\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \Gamma_t^2\)
13. Let \(\mathbf{v}_t = \max\{\mathbf{v}_t, \mathbf{v}_t\}\) and \(\Gamma_t = \frac{\Gamma_t}{\sqrt{\mathbf{v}_t} + \epsilon}\)
14. Update \(\mathbf{d}_t = \beta_1 \mathbf{d}_{t-1} + (1 - \beta_1) (\Gamma_t - \Gamma_{t-1})\)
15. Update the global model in the server: \(x_{t+1} = x_t + \eta \frac{\mathbf{m}_t}{\sqrt{\mathbf{v}_t} + \epsilon} + \mathbf{v}_t \mathbf{d}_t\)
16. Broadcast \(x_{t+1}\) to each client
17. end for

In particular, we add the \(\mathbf{d}_t\) term to the update rule of the global model. It denotes the changing trend of gradient and can be interpreted as the future information along
the undate direction. To weaken the impact of noise caused by mini-batch data in each optimization step, we adopt moving average decay on the $d_t$ term, which incorporates the history value on the current update and calculates a weighted average based on these values as shown in line 14 of Algorithm 1. In addition, we introduce $\bar{f}_t = \Gamma_t / (\sqrt{v_t} + \epsilon)$ to reduce the interference of outliers and correct the gradient, because the model may be unstable and the difference in the gradient direction by random batch sampling can be large, especially in the initial stage of training. Note that the vector square and vector division throughout the paper are element-wise.

4. Main Theoretical Results

In this section, we provide the theoretical convergence results of our proposed algorithm FedDA. Before presenting the main results, we first make some assumptions required for the analytical proof.

Assumption 1. (Smoothness) The cost function $f_i(x)$ is $L$-smooth with the constant $L > 0$; that is, for $\forall x, y \in \mathbb{R}^d$ and $i \in [n]$, we have

$$\| \nabla f_i(x) - \nabla f_i(y) \| \leq L \| x - y \|. \quad (2)$$

Note that Assumption 1 suggests the critical property, i.e., $f_i(x) - f_i(y) \leq \langle \nabla f_i(y), x - y \rangle + \frac{L}{2} \| x - y \|^2$. It is a commonly used assumption in theoretical analysis of stochastic non-convex optimization problems [25,36]. Similarly, it is also used in the algorithm analysis of federated learning, such as [18,34].

Assumption 2. (Unbiased and bounded gradient) For $\forall i \in [n]$ and $t \in [T]$, the stochastic gradient $\mathbf{g}^h_{i,t}$ is unbiased, i.e., $\mathbb{E}[\mathbf{g}^h_{i,t}] = \nabla f_i(x^h_{i,t})$; each function $f_i(x)$ has a $B$-bounded stochastic gradient, i.e., $\| \mathbf{g}^h_{i,t} \| \leq B$.

Assumption 2 is a general assumption condition to give the upper bound of the gradient in adaptive gradient-type methods, such as in [17,34].

Assumption 3. (Bounded variance) For $\forall i \in [n]$ and $t \in [T]$, the variance of each stochastic gradient for a local client is bounded, i.e., $\mathbb{E}[\| \mathbf{g}^h_{i,t} - \nabla f_i(x^h_{i,t}) \| ^2 ] \leq \delta_t^2$; the deviations among each loss function $f_i(x)$ are bounded, i.e., $\frac{1}{n} \sum_{i=1}^{n} \| \nabla f_i(x) - \nabla f(x) \| ^2 \leq \delta_g^2$.

Note that the local bound $\delta_t$ in Assumption 3 quantifies the random variation of the gradient estimator for each client, and the global bound $\delta_g$ describes the deviation between each loss function $f_i(x)$, which is mainly introduced by the heterogeneous data of each client. In particular, $\delta_g = 0$ means that the data of all the clients follow the same distribution, i.e., the homogeneous data setting.

Based on these assumptions, we present the following convergence theorems for the proposed algorithm. The case in which all clients take part in the model training during each round is considered first, i.e., $|S_t| = n$, any $t \in [T]$.

Theorem 1. (Full Client Participation) Suppose that Assumptions 1–3 hold. Let local learning rate $\eta_t$ satisfy $\eta_t \leq \min\{ \frac{1}{8\epsilon^2}, \frac{1}{SHL(HB + \epsilon)(\eta \beta_t/(1 - \beta_t)^2 + \eta \epsilon^2/\epsilon^2)} \}$. Under the full client participation case, the iterations of Algorithm 1 satisfy

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\| \nabla f(x_t) \| ^2] \leq 4(\eta_t HB + \epsilon) \left[ \frac{f(x_0) - f^*}{\eta_t HT} + \frac{C_1}{T} + C_2 \right], \quad (3)$$

where $C_1 = \frac{\beta_t B^2 d}{(1 - \beta_t)^2} + \frac{\eta_t HBL^2 d}{(1 - \beta_t)^2 \epsilon^2}$ and $C_2 = \frac{5\eta_t L^2 H}{2\epsilon \epsilon} (\delta_t^2 + 6H \delta_g^2) + \frac{3\eta_t \beta_t L^2 d}{2n(1 - \beta_t)^2 \epsilon^2} + \frac{\eta_t LHB^2 \epsilon^2}{\epsilon^2}$, and $f^*$ represents the optimal function value.
Remark 1. In Theorem 1, the expected squared gradient norm in the global model is used to measure the convergence in non-convex optimization. We find that the convergence upper bound of FedDA in (3) includes three terms: the first two terms relate to T and vanish to zero when T has a sufficiently large increase. The second term 
\( C_2 \) is independent of T, which depends on the local variance \( \delta_1 \) and the global variance \( \delta_2 \). We further divide \( C_2 \) into three parts. The first part in \( C_2 \) is the cumulative variance by conducting H local iteration steps which are greatly affected by the data heterogeneity. It will be smaller when each client follows the same data distribution, i.e., \( \delta_2 = 0 \). The local variance of the gradient computed by randomly sampling data points for each client contributes to the second part of \( C_2 \). It decreases at the rate \( 1/\eta \), which means more data samples from more clients, resulting in the decreasing of noise of stochastic gradients. The last part of \( C_2 \) depends on the number of local update H. This requires a sufficiently small \( \nu \) to offset the influence of the gradient difference to make the last past in \( C_2 \) small.

With Theorem 1, we restate the above convergence result for the FedDA algorithm by choosing proper learning rates:

**Corollary 1.** Let \( \eta = O\left( \frac{1}{\sqrt{T H}} \right) \), \( \eta = O\left( \sqrt{nH} \right) \). For \( T \geq nH \), the convergence rate of the FedDA algorithm under the all client participation case is:

\[
\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\|\nabla f(x_t)\|^2 \leq O\left( \frac{1}{\sqrt{n H T}} \right).
\]

**Remark 2.** From Corollary 1, we can see that the convergence rate of the proposed algorithm is \( O\left( \frac{1}{\sqrt{n H T}} \right) \) as long as \( T \geq nH \). In [37], the generated FedAvg with local and global learning rates achieves a convergence rate of \( O\left( 1/\sqrt{nKT} \right) \), where \( K \) denotes the number of local update steps. For federated adaptive optimization, the convergence rate of FedAdam [17] is \( O\left( 1/\sqrt{nKT} \right) \), which is similar to FedYogi when the data are heterogeneous. Thus, the proposed FedDA matches the convergence result of a non-convex problem in federated learning [18].

Next, a partial client participation case is considered, which is more practical in a federated learning setting for each communication round. \( S_t \) denotes the set of clients who participate in the training of the local model and communicate with the central server in round \( t \), and the size of \( S_t \) is \( m \), i.e., \( |S_t| = m, 0 < m \leq n, \forall t \in [T] \). We adopt the random sampling strategy without replacement. There is a random selection of clients in addition to the stochastic gradient. Here, \( \mathbb{E}[\cdot] \) is still used to express the expectations for these two kinds of randomness.

**Theorem 2.** (Partial Client Participation) Suppose that Assumptions 1–3 hold. Let local learning rate \( \eta_t \) satisfy \( \eta_t \leq \min\left\{ \frac{1}{nHL(n-1)\sqrt{6} \pi^2 e}, \frac{m(n-1)^2}{\sqrt{6} \pi^2 e} \right\} \). Under the partial client participation case, the iterations of Algorithm 1 satisfy:

\[
\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\|\nabla f(x_t)\|^2 \leq 4\left( \frac{nH}{T} + 1 \right) \left[ f(x_0) - f^* \right] + \frac{C_1}{T} + C_4.
\]

where \( C_1 = \frac{\beta_1 B^4_d}{(1-\beta_1)^2 \epsilon} + \frac{\eta \beta_1 B^4 d L H^2}{(1-\beta_1)^2 \epsilon^2} \), \( C_4 = \frac{5nH^2 L^2}{2\epsilon^2} \left( \delta_1^2 + 6H^2 \delta_2^2 + 3H^2 \delta_3^2 \right) \) and \( f^* \) represents the optimal function value.

**Remark 3.** In Theorem 2, the similar convergence upper bound is obtained for partial client participation but with a larger bias term \( C_4 \), which suggests that the convergence of FedDA has not
changed fundamentally except introducing extra variance because of low client participation and the use of a difference term, which can be controlled by choosing a small $\nu$.

From Theorem 2, we restate the above result by choosing a specific learning rate:

**Corollary 2.** If we choose $\eta_1 = O\left(\frac{1}{\sqrt{T}}\right)$, $\eta = O\left(\frac{1}{\sqrt{m}}\right)$, the convergence rate of the FedDA algorithm under the partial client participation case is:

$$
\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\|\nabla f(x_t)\|_2^2 \leq O\left(\frac{1}{\sqrt{T}\sqrt{m}}\right).
$$

(6)

From Corollary 2, we find that partial client participation does not have changes in convergence in the sense of order. It implies that a larger number of $m$ can help convergence. In addition, when choosing $m \leq H^3$, we have a slightly small convergence bound compared to [18].

5. Performance Analysis

This section first gives useful lemmas and builds a theoretical convergence guarantee for the proposed algorithm FedDA. Here, we define $x_0 = x_1$, $\Gamma_0 = 0$; for $t \geq 1$, a virtual sequence $y_t$ is introduced as follows:

$$
y_t = x_t + \frac{\beta_1}{1 - \beta_1} (x_t - x_{t-1}) - \nu \hat{\Gamma}_{t-1},
$$

(7)

and we obtain the following property for the difference of $y_t$,

$$
y_{t+1} - y_t = \frac{1}{1 - \beta_1} (x_{t+1} - x_t) - \frac{\beta_1}{1 - \beta_1} (x_t - x_{t-1}) - \nu (\hat{\Gamma}_t - \hat{\Gamma}_{t-1})
$$

$$
= \frac{1}{1 - \beta_1} \left( \frac{\eta m_t}{\sqrt{\nu_t + \epsilon}} + \nu d_t \right) - \nu (\hat{\Gamma}_t - \hat{\Gamma}_{t-1}) - \frac{\beta_1}{1 - \beta_1} \left( \frac{\eta m_{t-1}}{\sqrt{\nu_{t-1} + \epsilon}} + \nu d_{t-1} \right)
$$

$$
= \frac{\eta \beta_1}{1 - \beta_1} \left( \frac{1}{\sqrt{\nu_t + \epsilon}} - \frac{1}{\sqrt{\nu_{t-1} + \epsilon}} \right) \odot m_{t-1} + \frac{\eta}{\sqrt{\nu_t + \epsilon}} \odot \Gamma_t.
$$

(8)

**Lemma 1.** Under Assumption 2 and Algorithm 1, we have $\|\nabla f(x)\| \leq B$, $\|\Gamma_t\| \leq \eta_1 H B$, $\|m_t\| \leq \eta_1 H B$, $\|v_t\| \leq \eta_1^2 H^2 B^2$ and $\|d_t\| \leq \frac{2(1 - \beta_1)}{\beta_1} \frac{\eta_1 H B}{\epsilon}$. 

**Proof of Lemma 1.** Because of the bounded stochastic gradient by Assumption 2, we have

$$
\|\nabla f(x)\| = \|\mathbb{E}_\xi \nabla f(x, \xi)\| \leq \mathbb{E}_\xi \|\nabla f(x, \xi)\| \leq B.
$$

(9)

By the definition of $\Gamma_t$ in Algorithm 1, the following relation holds regardless of full participation or partial participation cases,

$$
\|\Gamma_t\| = \left\| \frac{1}{|S_t|} \sum_{i \in S_t} \Gamma_{i,t} \right\| = \left\| \frac{1}{|S_t|} \sum_{i \in S_t} \left( -\eta i \sum_{h=0}^{H-1} S_{i,h,t} \right) \right\| \leq \frac{\eta_1}{|S_t|} \sum_{i \in S_t} \sum_{h=0}^{H-1} \|S_{i,h,t}\| \leq \eta_1 H B.
$$

(10)

Hence, we can estimate the upper bound for $m_t$ and $v_t$ based on (10),

$$
\|m_t\| \leq (1 - \beta_1) \sum_{j=1}^{t} \beta_1^{t-j} \|\Gamma_j\| \leq \eta_1 H B,
$$

(11)

$$
\|v_t\| \leq (1 - \beta_2) \sum_{j=1}^{t} \beta_2^{t-j} \|\Gamma_j\| \leq \eta_1^2 H^2 B^2.
$$

(12)
In addition, we expand \(d_t\) and note \(\hat{\Gamma}_0 = 0\):

\[
\|d_t\| = \left\| \frac{1 - \beta_1}{\beta_1} \left( \hat{\Gamma}_t - (1 - \beta_1) \sum_{j=1}^{t} \beta_1^{t-j} \hat{\Gamma}_j \right) \right\|
\]

\[
\leq \frac{1 - \beta_1}{\beta_1} \left( \|\hat{\Gamma}_t\| + (1 - \beta_1) \sum_{j=1}^{t} \beta_1^{t-j} \|\hat{\Gamma}_j\| \right)
\]

\[
\leq \frac{2(1 - \beta_1)}{\beta_1} \eta H B \epsilon, \quad (13)
\]

where the last inequality holds by \(\|\Gamma_i\| \leq \frac{eHB}{\epsilon}\).

**Lemma 2.** Under Algorithm 1, for the difference sequence of \(\frac{1}{\sqrt{v_{t-1} + \epsilon}} - \frac{1}{\sqrt{v_t + \epsilon}}\), we have \(\sum_{t=1}^{T} \left\| \frac{1}{\sqrt{v_t + \epsilon}} - \frac{1}{\sqrt{v_{t-1} + \epsilon}} \right\|_1 \leq \frac{d}{\epsilon}\) and \(\sum_{t=1}^{T} \left\| \frac{1}{\sqrt{v_{t-1} + \epsilon}} - \frac{1}{\sqrt{v_t + \epsilon}} \right\|_1 \leq \frac{d}{\epsilon^2}\).

**Proof of Lemma 2.** From the definition of \(\hat{\psi}_i = \max\{\hat{\psi}_{i-1}, v_i\}\), we have

\[
\sum_{t=1}^{T} \left\| \frac{1}{\sqrt{v_t + \epsilon}} - \frac{1}{\sqrt{v_{t-1} + \epsilon}} \right\|_1 = \sum_{t=1}^{T} \left\| \frac{1}{\sqrt{v_{t-1} + \epsilon}} \right\|_1 - \left\| \frac{1}{\sqrt{v_T + \epsilon}} \right\|_1
\]

\[
= \left\| \frac{1}{\sqrt{v_0 + \epsilon}} \right\|_1 - \left\| \frac{1}{\sqrt{v_T + \epsilon}} \right\|_1
\]

\[
\leq \frac{d}{\epsilon}. \quad (14)
\]

Similarly, according to the definition of the \(\ell_2\) norm, we have

\[
\sum_{t=1}^{T} \left\| \frac{1}{\sqrt{v_{t-1} + \epsilon}} - \frac{1}{\sqrt{v_t + \epsilon}} \right\|_2^2 = \sum_{t=1}^{T} \sum_{j=1}^{d} \left( \frac{1}{\sqrt{v_{t-1} + \epsilon}} - \frac{1}{\sqrt{v_t + \epsilon}} \right)_j^2
\]

\[
\leq \sum_{t=1}^{T} \sum_{j=1}^{d} \frac{1}{\epsilon} \left( \frac{1}{\sqrt{v_{t-1} + \epsilon}} - \frac{1}{\sqrt{v_t + \epsilon}} \right)_j
\]

\[
\leq \frac{d}{\epsilon^2}. \quad (15)
\]

where the last inequality has the non-decreasing nature of \(\hat{\psi}_t\). We finish the proof.

**Lemma 3.** Under the scheme of randomly sampling without replacement, the average of partial model difference \(\Gamma_i\) is an unbiased estimator of \(\Gamma_t\), i.e.,

\[
E_{S_t}[\Gamma_i] = \Gamma_i, \quad (16)
\]

where \(\Gamma_i = \frac{1}{n} \sum_{t=1}^{n} \Gamma_{i,t}\).

**Proof of Lemma 3.** For a partial client participation case, let \(S_t = \{l_1, \cdots, l_m\}\) with \(|S_t| = m\); because the sampling distribution of each round is identical, we have

\[
E_{S_t}[\Gamma_i] = \frac{1}{m} E_{S_t} \left[ \sum_{l_i \in S_t} \Gamma_{i,l_i} \right] = \frac{1}{m} E_{S_t} \left[ \sum_{i=1}^{m} \Gamma_{i,l_i} \right] = E_{S_t}[\Gamma_{i,l_i}] = \frac{1}{m} \sum_{i=1}^{m} \Gamma_{i,l_i} = \Gamma_i. \quad (17)
\]
Lemma 4. Under Assumptions 1–3 and Algorithm 1, when all clients participate in communication with the server in each round, the following upper bound holds for the average of the aggregated model difference $\Gamma_t$:

$$E\|\Gamma_t\|^2 \leq \frac{\eta_f^2 H \delta_t^2}{n} + \frac{\eta_f^2}{m^2} E \left\| \sum_{i=1}^n \sum_{h=0}^{H-1} \nabla f_i(x_{i,t}^h) \right\|^2, \quad (18)$$

and for a partial participation case, we have

$$E\|\Gamma_t\|^2 \leq \frac{\eta_f^2 H \delta_t^2}{m} + \frac{\eta_f^2 (n - m)}{mn(n - 1)} \left( 15 H^3 L^2 \eta_f^2 \delta_t^2 + 6 H^4 \delta_t^2 + 3 H^2 \delta_t^2 \right) + (90 H^4 L^2 \eta_f^2 + 3 H^2) \|f(x)\|^2 \leq \frac{\eta_f^2 (m - 1)}{mn(n - 1)} E \left\| \sum_{i=1}^n \sum_{h=0}^{H-1} \nabla f_i(x_{i,t}^h) \right\|^2. \quad (19)$$

Proof of Lemma 4. When $|S_t| = n$, we have

$$E\|\Gamma_t\|^2 = E \left\| \frac{1}{n} \sum_{i=1}^n \Gamma_{i,t} \right\|^2 = \frac{1}{n^2} E \left\| \sum_{i=1}^n \sum_{h=0}^{H-1} \mathbf{g}_{i,t}^h - \nabla f_i(x_{i,t}^h) \right\|^2 \leq \frac{\eta_f^2 H \delta_t^2}{n} + \frac{\eta_f^2}{n^2} \left\| \sum_{i=1}^n \sum_{h=0}^{H-1} \nabla f_i(x_{i,t}^h) \right\|^2, \quad (20)$$

where the third equation holds by $E\|z\|^2 = E\|z - Ez\|^2 + E\|Ez\|^2$, and the last inequality holds by Assumption 3 and $E\|\sum_{i=1}^n z_i\|^2 = \sum_{i=1}^n E\|z_i\|^2$ when $z_i$ is an independent random variable with zero mean.

When $|S_t| = m$, and $m \in (0, n]$, we introduce the indicator function for convenience as follows

$$E\|\Gamma_t\|^2 = E \left\| \frac{1}{m} \sum_{i \in S_t} \Gamma_{i,t} \right\|^2 = \frac{1}{m^2} E \left\| \sum_{i=1}^m \mathbb{1}_{i \in S_t} \Gamma_{i,t} \right\|^2 = \frac{\eta_f^2}{m^2} E \left[ \left\| \sum_{i=1}^m \mathbb{1}_{i \in S_t} \sum_{h=0}^{H-1} \mathbf{g}_{i,t}^h - \nabla f_i(x_{i,t}^h) \right\|^2 + \left\| \sum_{i=1}^m \mathbb{1}_{i \in S_t} \sum_{h=0}^{H-1} \nabla f_i(x_{i,t}^h) \right\|^2 \right]$$

$$= \frac{\eta_f^2}{m^2} E \left[ \left\| \sum_{i=1}^m \mathbb{1}_{i \in S_t} \sum_{h=0}^{H-1} \mathbf{g}_{i,t}^h - \nabla f_i(x_{i,t}^h) \right\|^2 \right] + \frac{\eta_f^2}{m^2} E \left[ \left\| \sum_{i=1}^m \mathbb{1}_{i \in S_t} \sum_{h=0}^{H-1} \nabla f_i(x_{i,t}^h) \right\|^2 \right] \leq \frac{\eta_f^2 H \delta_t^2}{m} + \frac{\eta_f^2}{m^2} E \left[ \left\| \sum_{i=1}^m \mathbb{1}_{i \in S_t} \sum_{h=0}^{H-1} \nabla f_i(x_{i,t}^h) \right\|^2 \right], \quad (21)$$
where the third equation follows by the definition of variance, i.e., $E ||z||^2 = E ||z - Ez||^2 + ||Ez||^2$, the fifth equation holds by the fact that $E ||\sum_{i=1}^n z_i||^2 = n^m ||z_i||^2$ when $z_i$ is an independent random variable with zero mean, and $P\{i \in S_t\} = \frac{m-i}{m}$. The last one is due to the bounded variance in Assumption 3.

Recalling the equation $||\sum_{i=1}^n z_i||^2 = n \sum_{i=1}^n ||z_i||^2 - \frac{1}{2} \sum_{i \neq j} ||z_i - z_j||^2$, for the last term in (21), we have

$$
E \left\| \sum_{i=1}^n \sum_{h=0}^{H-1} \nabla f_i(x_{i,h}^j) \right\|^2
= \sum_{i \neq j} \sum_{h=0}^{H-1} \left\langle \nabla f_i(x_{i,h}^j), \nabla f_j(x_{j,h}^i) \right\rangle + \sum_{i=1}^n \sum_{h=0}^{H-1} \left\| \nabla f_i(x_{i,h}^j) \right\|^2
= n \sum_{i=1}^n \sum_{h=0}^{H-1} \left\| \nabla f_i(x_{i,h}^j) \right\|^2 + \sum_{i \neq j} \sum_{h=0}^{H-1} \left\langle \nabla f_i(x_{i,h}^j), \nabla f_j(x_{j,h}^i) \right\rangle
= \frac{m(m-1)}{2n(n-1)} \sum_{i \neq j} \sum_{h=0}^{H-1} \left\| \nabla f_i(x_{i,h}^j) - \nabla f_j(x_{j,h}^i) \right\|^2 + \sum_{i=1}^n \sum_{h=0}^{H-1} \left\| \nabla f_i(x_{i,h}^j) \right\|^2
= \frac{m(m-1)}{n(n-1)} \sum_{i=1}^n \sum_{h=0}^{H-1} \left\| \nabla f_i(x_{i,h}^j) \right\|^2 + \frac{m(m-1)}{n(n-1)} \sum_{i=1}^n \sum_{h=0}^{H-1} \left\| \nabla f_i(x_{i,h}^j) \right\|^2,
$$
where the second equation holds by $P\{i \in S_t\} = \frac{m-i}{m}$ and $P\{i \in S_t\} = \frac{m-m}{m(m-1)}$, while the third one and last one hold by $||\sum_{i=1}^n z_i||^2 = n \sum_{i=1}^n ||z_i||^2 - \frac{1}{2} \sum_{i \neq j} ||z_i - z_j||^2$ and $(z_1, z_2) = \frac{1}{2} (||z_1||^2 + ||z_2||^2 - ||z_1 - z_2||^2)$, respectively. In addition, by plugging $\nabla f_i(x_i) \pm \nabla f_i(x_i)$ into the first term of (22), we obtain

$$
\sum_{i=1}^n \sum_{h=0}^{H-1} \left\| \nabla f_i(x_{i,h}^j) \right\|^2
= \sum_{i=1}^n \sum_{h=0}^{H-1} \left\| \nabla f_i(x_{i,h}^j) - \nabla f_i(x_i) + \nabla f_i(x_i) - \nabla f_i(x_i) + \nabla f_i(x_i) \right\|^2
\leq 3H^2 \sum_{i=1}^n \sum_{h=0}^{H-1} E \left\| x_{i,h}^j - x_i \right\|^2 + 3nH^2 \left( \delta^2 + ||\nabla f_i(x_i)||^2 \right)
\leq (90nH^2 L^2 \eta^2 + 15n^2) ||\nabla f_i(x_i)||^2 + 3nH^2 \delta^2 + 15nH^2 \eta^2 (\delta^2 + 6H^2 \delta^2),
$$
where the first inequality follows by Assumptions 1, 3 and $||\sum_{i=1}^n z_i||^2 \leq n \sum_{i=1}^n ||z_i||^2$, and the last one follows Lemma 3 in [17]. Substituting (22), (23) into (21), we obtain (19), which finishes the proof. \(\square\)

**Lemma 5.** Under Assumptions 1–3 and Algorithm 1, for the full participation case, we have

$$
\sum_{i=1}^T E \left\| \mathbf{m}_i \right\|^2 \leq \frac{TH\eta^2 \delta^2}{n} + \frac{\eta^2}{n^2} \sum_{i=1}^T \sum_{h=0}^{H-1} \left\| \nabla f_i(x_{i,h}^j) \right\|^2,
$$
and for the partial participation case, we have

$$
\sum_{i=1}^T E \left\| \mathbf{m}_i \right\|^2 \leq \frac{T\eta^2 \delta^2}{m} + \frac{\eta^2}{m(n-1)} \left[ 15TH^3 L^2 \eta^2 (\delta^2 + 6H^2 \delta^2) + 3TH^2 \delta^2 + (90H^4 L^2 \eta^2 + 3H^2) \right] \sum_{i=1}^T \left\| \nabla f_i(x_i) \right\|^2 + \frac{\eta^2}{mn(n-1)} \sum_{i=1}^T \sum_{h=0}^{H-1} \left\| \nabla f_i(x_{i,h}^j) \right\|^2.
$$
\textbf{Proof.} For the full client participation scheme, i.e., $|S| = n$, according to Algorithm 1, we have
\[
\mathbb{E}\|m_t\|^2 = \mathbb{E}\left\| (1 - \beta_1) \sum_{j=1}^{t} \beta_1^{j-1} \Gamma_j \right\|^2 \\
\leq (1 - \beta_1)^2 \sum_{i=1}^{d} \mathbb{E}\left[ \left( \sum_{j=1}^{t} \beta_1^{j-1} \Gamma_{j,i} \right)^2 \right] \\
\leq (1 - \beta_1)^2 \sum_{i=1}^{d} \mathbb{E}\left[ \left( \sum_{j=1}^{t} \beta_1^{j-1} \right)^2 \left( \sum_{j=1}^{t} \beta_1^{j-1} \Gamma_{j,i}^2 \right) \right] \\
\leq (1 - \beta_1)^2 \sum_{i=1}^{d} \mathbb{E}\left( \sum_{j=1}^{t} \beta_1^{j-1} \Gamma_{j,i} \right) \\
= (1 - \beta_1)^2 \sum_{j=1}^{t} \beta_1^{j-1} \mathbb{E}\|\Gamma_j\|^2 \\
\leq \frac{\eta_1^2 H^2 \beta_1^2}{n} + \frac{\eta_2^2}{n} \mathbb{E}\left\| \sum_{i=1}^{H-1} \sum_{h=0}^{1} \nabla f_i(x^h_{t,i}) \right\|^2, \tag{26}
\]
where the first inequality and second equation hold by the definition of the $\ell_2$ norm, and the second inequality follows by Cauchy–Schwarz inequality $(\sum_{k=1}^{n} a_k b_k)^2 \leq \sum_{k=1}^{n} a_k^2 \sum_{k=1}^{n} b_k^2$. The last inequality is due to (18) in Lemma 4 and $\beta_1 < 1$. Taking summation over $l$ from 1 to $T$ for (26), we obtain the desired results (24).

For the partial client participation scheme, i.e., $|S| = m$, the main difference is using (19) instead of (18) in Lemma 4 when bound to $\mathbb{E}\|m_t\|^2$. Following a similar proof above, we can obtain the result (25). \hfill \square

Finally, with the previous lemmas, we return to prove Theorem 1.

\textbf{Proof of Theorem 1.} Due to the smoothness of $f$, we have
\[
\mathbb{E}f(y_{t+1}) - \mathbb{E}f(y_t) \\
\leq \mathbb{E}\langle \nabla f(y_t), y_{t+1} - y_t \rangle + \frac{L}{2} \mathbb{E}\|y_{t+1} - y_t\|^2 \\
= \mathbb{E}\left\langle \nabla f(y_t), \frac{\eta}{\sqrt{\Gamma_t + \epsilon}} \odot \Gamma_t \right\rangle + \frac{\eta_1 \beta_1}{1 - \beta_1} \mathbb{E}\left\langle \nabla f(y_t), \left( \frac{1}{\sqrt{\Gamma_{t-1} + \epsilon}} - \frac{1}{\sqrt{\Gamma_t + \epsilon}} \right) m_{t-1} \right\rangle \\
+ \frac{\eta_2^2 L}{2} \mathbb{E}\left\| \frac{\Gamma_t}{\sqrt{\Gamma_t + \epsilon}} - \frac{\beta_1}{1 - \beta_1} \left( \frac{1}{\sqrt{\Gamma_{t-1} + \epsilon}} - \frac{1}{\sqrt{\Gamma_t + \epsilon}} \right) m_{t-1} \right\|^2 \\
+ \mathbb{E}\left\langle \nabla f(y_t) - \nabla f(x_t), \frac{\eta}{\sqrt{\Gamma_t + \epsilon}} \odot \Gamma_t \right\rangle. \tag{27}
\]

We first bound $A_1$ as following:
\[
A_1 = \mathbb{E}\left\langle \nabla f(x_t), \frac{\eta}{\sqrt{\Gamma_t + \epsilon}} \odot \Gamma_t \right\rangle \\
= \eta \mathbb{E}\left\langle \frac{\nabla f(x_t)}{\sqrt{\Gamma_t + \epsilon}}, \Gamma_t + \eta_1 H \nabla f(x_t) - \eta_1 H \nabla f(x_t) \right\rangle \\
= \eta \mathbb{E}\left\langle \frac{\nabla f(x_t)}{\sqrt{\Gamma_t + \epsilon}} - \frac{\eta_1}{n} \sum_{i=1}^{n} \sum_{h=0}^{1} \delta_{i,h} + \frac{\eta_1 H}{n} \sum_{i=1}^{n} \nabla f_i(x_t) \right\rangle - \eta \eta_1 H \mathbb{E}\left\| \frac{\nabla f(x_t)}{\sqrt{\Gamma_t + \epsilon}} \right\|^2. \tag{28}
\]
Note that the first term of (28) can be further estimated:

\[
\eta E\left(\frac{\nabla f(x_i)}{\sqrt{\hat{v}_i} + \epsilon} - \frac{\eta_i}{n} \sum_{h=0}^{H-1} g_{ih} \nabla f_i(x_i)\right)
\]

\[
= \eta E\left(\frac{\sqrt{\eta} H \nabla f(x_i)}{\sqrt{\hat{v}_i} + \epsilon}, - \frac{\sqrt{\eta} H}{Hn} \sum_{h=0}^{H-1} \frac{1}{n} \sum_{i=1}^{n} \left(\nabla f_i(x_{ih}) - \nabla f_i(x_i)\right)\right)
\]

\[
= \eta \eta_i \left(\sum_{h=0}^{H-1} \sum_{i=1}^{n} \left(\nabla f_i(x_{ih}) - \nabla f_i(x_i)\right)\right)^2 - \eta \eta_i \left(\sum_{h=0}^{H-1} \sum_{i=1}^{n} \frac{1}{n} \sum_{i=1}^{n} \left(\nabla f_i(x_{ih}) - \nabla f_i(x_i)\right)^2\right)
\]

\[
+ \frac{\eta \eta_i H}{2} \left(\nabla f(x_i) \right)^2 - \frac{\eta \eta_i H}{2} \left(\nabla f(x_i) \right)^2 - \frac{\eta \eta_i H}{2} \left(\nabla f(x_i) \right)^2 - \frac{\eta \eta_i H}{2} \left(\nabla f(x_i) \right)^2
\]

\[
\leq \frac{3\eta \eta_i H^2}{4} \left(\nabla f(x_i) \right)^2 + \frac{5\eta \eta_i^2 H^2}{2} (\Delta^2 + 6H\delta^2)
\]

\[
= \frac{3\eta \eta_i H^2}{4} \left(\nabla f(x_i) \right)^2 + \frac{5\eta \eta_i^2 H^2}{2} (\Delta^2 + 6H\delta^2)
\]

(29)

where the second equation is from the equality \(\langle a, b \rangle = \frac{1}{2} \left(\|a\|^2 + \|b\|^2 - \|a - b\|^2\right)\), the first inequality follows the fact that \(\|\sum_{i=1}^{n} a_i\|^2 \leq n \sum_{i=1}^{n} \|a_i\|^2\) and the smoothness of function \(f_i\). Lemma 3 in [17] and \(\eta_i \leq \frac{1}{\eta L}\) result in the last inequality.

Plugging (29) into (28), we have

\[
A_1 \leq -\frac{\eta \eta_i H^2}{4} \left(\nabla f(x_i) \right)^2 + \frac{5\eta \eta_i^2 H^2}{2} (\Delta^2 + 6H\delta^2)
\]

\[
- \frac{\eta \eta_i H^2}{2} \left(\nabla f(x_i) \right)^2 - \frac{\eta \eta_i H^2}{2} \left(\nabla f(x_i) \right)^2 - \frac{\eta \eta_i H^2}{2} \left(\nabla f(x_i) \right)^2
\]

(30)

We further estimate the term \(A_2\).

\[
A_2 = -\frac{\eta \beta_1}{1 - \beta_1} E\left(\nabla f(y_i) \left(\frac{m_{i-1}}{\sqrt{\hat{v}_{i-1}} + \epsilon} - \frac{m_{i-1}}{\sqrt{\hat{v}_{i}} + \epsilon}\right)\right)
\]

\[
\leq \frac{\eta \beta_1}{1 - \beta_1} \left\|\nabla f(y_i)\right\| \left(\frac{m_{i-1}}{\sqrt{\hat{v}_{i-1}} + \epsilon} - \frac{m_{i-1}}{\sqrt{\hat{v}_{i}} + \epsilon}\right)
\]

\[
\leq \frac{\eta \eta \beta_1 H^2}{1 - \beta_1} \left\|\frac{1}{\sqrt{\hat{v}_{i-1}} + \epsilon} - \frac{1}{\sqrt{\hat{v}_{i}} + \epsilon}\right\|_1\]

(31)

where the first inequality is from \(\langle a, b \rangle \leq \|a\|\|b\|\), and the second one holds by Lemma 1.

For the term \(A_3\) in (27), applying the inequality \(\|a + b\|^2 \leq 2(\|a\|^2 + \|b\|^2)\) and (11) in Lemma 1 yields

\[
A_3 = \frac{\eta \eta L}{2} - \beta_1 \left(\frac{m_{i-1}}{\sqrt{\hat{v}_{i-1}} + \epsilon} - \frac{m_{i-1}}{\sqrt{\hat{v}_{i}} + \epsilon}\right) + \frac{\Gamma_1}{\sqrt{\hat{v}_{i}} + \epsilon}^2
\]

\[
\leq \frac{\eta \eta \beta_1 L}{(1 - \beta_1)^2} \left(\frac{m_{i-1}}{\sqrt{\hat{v}_{i-1}} + \epsilon} - \frac{m_{i-1}}{\sqrt{\hat{v}_{i}} + \epsilon}\right)^2 + L \left\|\frac{\eta \Gamma_1}{\sqrt{\hat{v}_{i}} + \epsilon}\right\|^2
\]

\[
\leq \frac{\eta \eta \beta_1 L}{(1 - \beta_1)^2} \left(\frac{1}{\sqrt{\hat{v}_{i-1}} + \epsilon} - \frac{1}{\sqrt{\hat{v}_{i}} + \epsilon}\right)^2 + L \left\|\frac{\eta \Gamma_1}{\sqrt{\hat{v}_{i}} + \epsilon}\right\|^2
\]

(32)
Using similar techniques to bound the term $T_A$, we have

\[ A_4 = \mathbb{E} \left( \nabla f(y_t) - \nabla f(x_t), \frac{\eta}{\sqrt{v_t} + \epsilon} \odot \Gamma_t \right) \leq \mathbb{E} \left\| \nabla f(y_t) - \nabla f(x_t) \right\| \left\| \frac{\eta}{\sqrt{v_t} + \epsilon} \odot \Gamma_t \right\| \leq L \mathbb{E} \left\| y_t - x_t \right\| \left\| \frac{\eta}{\sqrt{v_t} + \epsilon} \odot \Gamma_t \right\| \leq \frac{L}{2} \left\| \frac{\eta}{1 - \beta_1} \left( \frac{m_{t-1}}{\sqrt{v_{t-1}} + \epsilon} + v d_{t-1} \right) - \frac{v \Gamma_{t-1}}{\sqrt{v_{t-1}} + \epsilon} \right\|^2 + \frac{L \eta^2}{2} \left\| \frac{\eta}{\sqrt{v_t} + \epsilon} \right\|^2 \leq \frac{3 \eta^2 L}{2} \frac{\beta_1^2}{(1 - \beta_1)^2} \left\| \frac{m_{t-1}}{\sqrt{v_{t-1}} + \epsilon} \right\|^2 + \frac{3 L \eta^2}{2} \frac{\beta_1^2}{(1 - \beta_1)^2} \left\| d_{t-1} \right\|^2 + \frac{6 L \eta^2 \mu^2 H^2 B^2 \nu^2}{e^2}. \] (33)

Summing from $t = 1, \ldots, T$ and using the inequality in (24) of $\sum_{t=1}^{T} \left\| m_t \right\|^2$, we have

\[ \sum_{t=1}^{T} T_A \leq \frac{3 \eta^2 L}{2} \frac{\beta_1^2}{(1 - \beta_1)^2} \frac{T \eta H^2 \delta_t^2}{n} + \frac{L (\eta^2 + 3 \eta^2)}{2 e^2} \sum_{t=1}^{T} \mathbb{E} \left\| \Gamma_t \right\|^2 + 6 T L \eta^2 \mu^2 H^2 B^2 \nu^2 \] (34)

Plugging (30)–(34) into (27) and summing over $t$ from 1 to $T$, we have

\[ \mathbb{E} f(y_{T+1}) - \mathbb{E} f(y_1) \leq -\frac{\eta \xi H}{4} \sum_{t=1}^{T} \mathbb{E} \left\| \nabla f(x_t) \right\|^2 - \frac{\eta \xi H}{2 n^2} \sum_{t=1}^{T} \mathbb{E} \left\| \frac{\nabla f_t(x_t)}{\sqrt{v_t} + \epsilon} \right\|^2 + \frac{\eta \xi \beta_1 H B}{1 - \beta_1} \sum_{t=1}^{T} \mathbb{E} \left\| \frac{1}{\sqrt{v_{t-1}} + \epsilon} \right\|^2 - \frac{1}{\sqrt{v_t} + \epsilon} \right\|^2 + \frac{(2 \xi + 1) L \eta^2}{2 e^2} \sum_{t=1}^{T} \mathbb{E} \left\| \Gamma_t \right\|^2 + \frac{3 \eta^2 \xi \beta_1 H B^2 L \delta_t^2}{2 n} + \frac{5 \eta^2 \mu^2 H^2 T}{2 e^2} (\delta_t^2 + 6 H \delta_t^2) \] (35)

By using Lemma 2 and rearranging the terms, we have
\[ \mathbb{E}f(y_{T+1}) - \mathbb{E}f(y_1) \leq \frac{-\eta \eta H}{4(\eta HB + \epsilon)} \sum_{t=1}^{T} \mathbb{E}\|\nabla f(x_t)\|^2 + \frac{\eta \eta_1 B^2 d}{(1 - \beta_1)^2} + \frac{3(\eta \eta_1 B^2 L \delta_1^2)}{2n(1 - \beta_1)^2 \epsilon^2} + \frac{6T \eta \eta H L^2 B^2 \nu^2}{\epsilon^2} \]
\[ - \left( \frac{\eta \eta_1}{2Hn^2(\eta HB + \epsilon)} - \frac{3\eta \eta_1 \beta_1 L^2}{2n^2(1 - \beta_1)^2 \epsilon^2} - \frac{3L(\eta^2 + \nu^2)\eta_1^2}{2n^2 \epsilon^2} \right) \sum_{t=1}^{T} \mathbb{E} \left\| \sum_{h=0}^{n-1} \nabla f_i(x_{i,t}) \right\|^2 \]
\[ + \frac{5\eta \eta_1 L^2 H^2 T}{2\epsilon}(\delta_1^2 + 6H\delta_2^2) + \frac{\eta^2 \eta_1^2 H^2 B^2 \beta_1 L_d}{(1 - \beta_1)^2 \epsilon^2} \]
\[ \leq - \frac{\eta \eta H}{4(\eta HB + \epsilon)} \sum_{t=1}^{T} \mathbb{E}\|\nabla f(x_t)\|^2 + \frac{\eta \eta_1 B^2 d}{(1 - \beta_1)^2} + \frac{5\eta \eta_1 L^2 H^2 T}{2\epsilon}(\delta_1^2 + 6H\delta_2^2) \]
\[ + \frac{3\eta^2 \eta_1^2 \beta_1 L H \delta_1^2}{2n(1 - \beta_1)^2 \epsilon^2} + \frac{5\eta \eta_1^2 H^2 B^2 \beta_1 L_d}{2\epsilon}(\delta_1^2 + 6H\delta_2^2) \]
\[ + \frac{6\eta \eta_1^2 L^2 B^2 \nu^2}{\epsilon^2}, \quad (36) \]

where the first inequality follows by \( \frac{-\eta \eta H}{4(\eta HB + \epsilon)} \geq \frac{1}{H \epsilon + \epsilon} \) and \( \epsilon \leq 1 \), and the last one holds by the fact that \( \frac{\eta \eta H}{2Hn^2(\eta HB + \epsilon)} - \frac{3\eta \eta_1 \beta_1 L^2}{2n^2(1 - \beta_1)^2 \epsilon^2} - \frac{3L(\eta^2 + \nu^2)\eta_1^2}{2n^2 \epsilon^2} \geq 0 \); that is, \( \eta \leq \frac{n \eta^2}{3HL(\eta HB + \epsilon)(\beta_1^2 + (1 - \beta_1)^2 + \nu^2 + \nu^2)} \).

Dividing by \( T \) both sides of (36), we obtain
\[ \frac{\eta \eta H}{4(\eta HB + \epsilon)T} \sum_{t=1}^{T} \mathbb{E}\|\nabla f(x_t)\|^2 \leq \frac{f(y_1) - f^*_T}{T} + \frac{\eta \eta_1 B^2 d}{T(1 - \beta_1)^2} + \frac{\eta^2 \eta_1^2 H^2 B^2 \beta_1 L_d}{T(1 - \beta_1)^2 \epsilon^2} \]
\[ + \frac{3\eta^2 \eta_1^2 \beta_1 L H \delta_1^2}{2n(1 - \beta_1)^2 \epsilon^2} + \frac{5\eta \eta_1^2 H^2 B^2 \beta_1 L_d}{2\epsilon}(\delta_1^2 + 6H\delta_2^2) \]
\[ + \frac{6\eta \eta_1^2 L^2 B^2 \nu^2}{\epsilon^2}, \quad (37) \]

Therefore,
\[ \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\|\nabla f(x_t)\|^2 \leq 4(\eta HB + \epsilon) \left[ \frac{f(y_1) - f^*}{\eta \eta H T} + \frac{\beta_1 B^2 d}{T(1 - \beta_1)^2} + \frac{3\eta \eta_1 \beta_1 L \delta_1^2}{2n(1 - \beta_1)^2 \epsilon^2} \right] \]
\[ + \frac{5\eta \eta_1^2 L^2 H}{2\epsilon}(\delta_1^2 + 6H\delta_2^2) + \frac{6\eta \eta_1^2 L^2 B^2 \nu^2}{\epsilon^2}. \quad (38) \]

When using \( C_1 = \frac{\beta_1 B^2 d}{T(1 - \beta_1)^2} + \frac{\eta \eta_1 B^2 \beta_1 L_d}{T(1 - \beta_1)^2 \epsilon^2} \) and \( C_2 = \frac{5\eta \eta_1^2 L^2 H}{2\epsilon}(\delta_1^2 + 6H\delta_2^2) + \frac{3\eta \eta_1 \beta_1 L \delta_1^2}{2n(1 - \beta_1)^2 \epsilon^2} + \frac{6\eta \eta_1^2 L^2 B^2 \nu^2}{\epsilon^2} \), (38) can be expressed as follows
\[ \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\|\nabla f(x_t)\|^2 \leq 4(\eta HB + \epsilon) \left[ \frac{f(y_1) - f^*}{\eta \eta H T} + \frac{C_1}{T} + C_2 \right]. \quad (39) \]

The proof of Theorem 1 is finished. \( \square \)

Next, we consider the case of partial client participation. Different from full client participation, the aggregated model difference \( \Gamma_t \) over the subset \( S_t \) is \( \frac{1}{m} \sum_{t \in S_t} \Gamma_{i,t} \) in round \( t \). For convenience and the consistency of notations, we denote \( \Gamma_t \) as the average of the aggregated global model difference hereafter, i.e., \( \Gamma_t = \frac{1}{n} \sum_{i=1}^{n} \Gamma_{i,t} \). In fact, \( \Gamma_t \) is an unbiased estimate of \( \Gamma_t \) by Lemma 3.
Proof of Theorem 2. According to Assumption 1 and taking expectation over the randomness in round $t$, we have

$$
\mathbb{E} f(y_{t+1}) - \mathbb{E} f(y_1)
\leq \mathbb{E} \langle \nabla f(y_t), y_{t+1} - y_t \rangle + \frac{L}{2} \mathbb{E} \| y_{t+1} - y_t \|^2
\leq \mathbb{E} \langle \nabla f(x_t), \frac{\eta}{\sqrt{\nu_t} + \epsilon} \circ \Gamma_t \rangle - \frac{\eta \beta_1}{1 - \beta_1} \mathbb{E} \langle \nabla f(y_t), \left( \frac{m_{t-1}}{\sqrt{\nu_{t-1}} + \epsilon} - \frac{m_{t-1}}{\sqrt{\nu_t + \epsilon}} \right) \rangle
+ \frac{\eta^2 L}{2} \left\| \frac{\Gamma_t}{\sqrt{\nu_t + \epsilon}} - \frac{\beta_1}{1 - \beta_1} \left( \frac{m_{t-1}}{\sqrt{\nu_{t-1}} + \epsilon} - \frac{m_{t-1}}{\sqrt{\nu_t + \epsilon}} \right) \right\|^2
+ \mathbb{E} \langle \nabla f(y_t) - \nabla f(x_t), \frac{\eta}{\sqrt{\nu_t} + \epsilon} \circ \Gamma_t \rangle.
$$

(40)

Since $\mathbb{E}_{\xi_t}[\Gamma_t] = \Gamma_t$ by Lemma 3, the term $A'_1$ can be estimated by the same upper bound as in (30). Similar to (31) and (32), we obtain the estimator for $A'_2$ and $A'_3$, respectively.

$$
\sum_{t=1}^{T} A'_2 \leq \frac{\eta \eta T L^2 B^2}{1 - \beta_1} \sum_{t=1}^{T} \mathbb{E} \left\| \frac{1}{\sqrt{\nu_{t-1}} + \epsilon} - \frac{1}{\sqrt{\nu_t + \epsilon}} \right\|^2,
$$

(41)

and

$$
\sum_{t=1}^{T} A'_3 \leq \frac{\eta^2 \eta^2 H^2 B^2 \beta_1^2 L}{(1 - \beta_1)^2} \sum_{t=1}^{T} \mathbb{E} \left\| \frac{1}{\sqrt{\nu_{t-1}} + \epsilon} - \frac{1}{\sqrt{\nu_t + \epsilon}} \right\|^2 + \frac{\eta^2 L}{e^2} \sum_{t=1}^{T} \mathbb{E} \| \nu_t \|^2,
$$

(42)

and

$$
\sum_{t=1}^{T} A'_4 \leq \frac{3\eta^2 L}{2e^2} \left( \frac{\beta_1^2}{1 - \beta_1} \right)^2 \sum_{t=1}^{T} \mathbb{E} \| m_t \|^2 + \frac{6T L \eta^3 H^2 B^2 \nu^2}{e^2} + \frac{L \eta (1 + 3 \nu^2)}{2e^2} \sum_{t=1}^{T} \mathbb{E} \| \nu_t \|^2.
$$

(43)

Summing over $t$ from 1 to $T$ for (40) and plugging (30), (41)–(43), we have

$$
\mathbb{E} f(y_{T+1}) - \mathbb{E} f(y_1)
\leq \sum_{t=1}^{T} (A'_1 + A'_2 + A'_3 + A'_4)
\leq -\frac{\eta \eta H}{4} \sum_{t=1}^{T} \mathbb{E} \left\| \frac{\nabla f(x_t)}{\sqrt{\nu_t} + \epsilon} \right\|^2 + \frac{5\eta \eta^3 L^2 H^2 T}{2e^2} (\delta_t^2 + 6H \delta_S^2) + \frac{\eta \eta \beta_1}{(1 - \beta_1) \epsilon} d
+ \frac{\eta \eta H^2 B^2 \beta_1^2 L}{1 - \beta_1^2} \mathbb{E} \| m_t \|^2 + \frac{3\eta^3 \beta_1^2 L}{2(1 - \beta_1) e^2} \sum_{t=1}^{T} \mathbb{E} \| m_t \|^2 + \frac{6T \eta^3 H^2 B^2 \nu^2}{e^2},
$$

(44)
where the inequality results from Lemma 2. Applying (19) and (25), we obtain
\[
\mathbb{E} f(y_{T+1}) - \mathbb{E} f(y_1) \leq -\frac{\eta_1 H}{4(\eta_1 H B + \epsilon)} \sum_{i=1}^{T} \mathbb{E} \| \nabla f(x_i) \|^2 + \frac{5\eta_1^3 H^2 T}{2e} (\delta^2 + 6H^2 \delta^2) + \frac{\eta_1 \beta_1 H B^2 d}{(1 - \beta_1) e} \\
- \left[ \frac{\eta_1}{2H n^2 (H B + \epsilon)} - C_3 \frac{\eta_1 \eta L (m - 1)}{mn(n - 1)} \right] \sum_{i=1}^{T} \mathbb{E} \left\| \sum_{h=0}^{H-1} \sum_{l=1}^{n} \nabla f_i(x_{h,l}) \right\|^2 \\
+ C_3 \eta L \left[ \frac{\eta_1^2 H^2 \delta^2}{m} + \frac{\eta_1^2 (n - m)}{m(n - 1)} (15H^3 L^2 \eta_1^2 (\delta^2 + 6H^2 \delta^2) + 3H^2 \delta^2) \right] \\
+ \frac{C_3 \eta_1^2 \eta L (n - m)}{m(n - 1)} (90H^4 L^2 \eta_1^2 + 3H^2) \sum_{i=1}^{T} \| \nabla f(x_i) \|^2 + \frac{\eta_1^2 \beta_1^2 H B^2 B^2 Ld}{(1 - \beta_1)^2 e^2} \\
+ \frac{6TL \eta_1^2 H^2 B^2 v^2}{e^2},
\] (45)
where $C_3 = \frac{2g + 3v^2 + 1}{2e^2} \frac{3v^2 \beta_1^2}{(1 - \beta_1)^2 e^2}$. We rearrange the terms and let $\eta_1$ satisfy the condition
\[
\frac{\eta_1}{2H n^2 (H B + \epsilon)} - C_3 \frac{\eta_1 \eta L (m - 1)}{mn(n - 1)} \geq 0, \text{ that is } \eta_1 \leq \frac{m(n - 1)}{nH L (m - 1)/2 \eta^2 + 3v^2 + 1 + 3v^2 \beta_1^2 (1 - \beta_1)^2 e^2 (H B + \epsilon)}.
\]
In addition, $\eta_1$ also satisfies
\[
\frac{\eta_1 H}{\eta_1 (H B + \epsilon)} - \frac{C_3 \eta_1 \eta L (n - m)}{m(n - 1)} (90H^4 L^2 \eta_1^2 + 3H^2) \geq \frac{\eta_1 H}{8(\eta_1 (H B + \epsilon))}.
\]
Thus, from (45), we have
\[
\frac{\eta_1 H}{8T (\eta_1 H B + \epsilon)} \sum_{i=1}^{T} \mathbb{E} \| \nabla f(x_i) \|^2 \\
\leq \frac{f(y_1)}{T} - \frac{f(y_{T+1})}{T} + \frac{5\eta_1^3 H^2 T}{2e} (\delta^2 + 6H^2 \delta^2) + \frac{\eta_1 \beta_1 H B^2 d}{T (1 - \beta_1) e} \\
+ C_3 \eta L \left[ \frac{\eta_1^2 H^2 \delta^2}{m} + \frac{\eta_1^2 (n - m)}{m(n - 1)} (15H^3 L^2 \eta_1^2 (\delta^2 + 6H^2 \delta^2) + 3H^2 \delta^2) \right] \\
+ \frac{6TL \eta_1^2 H^2 B^2 v^2}{e^2} + \frac{\eta_1^2 \beta_1^2 H B^2 B^2 Ld}{T (1 - \beta_1)^2 e^2}.
\] (46)
Therefore,
\[
\frac{1}{T} \sum_{i=1}^{T} \mathbb{E} \| \nabla f(x_i) \|^2 \leq 8(\eta_1 H B + \epsilon) \left[ \frac{f(y_1) - f_*}{\eta_1 H T} + \frac{C_4}{T} + C_4 \right],
\] (47)
where $\eta_1 = \frac{\beta_1 B^2 d}{(1 - \beta_1)^2 e^2}$ and $C_4 = \frac{5\eta_1^3 H^2}{2e} (\delta^2 + 6H^2 \delta^2) + \frac{6\eta_1 H B^2 v^2}{e^2} + C_3 \left[ \frac{\eta_1^2 \beta_1^2}{m} + \frac{\eta_1 (n - m)}{m(n - 1)} (15H^2 L^2 \eta_1^2 (\delta^2 + 6H^2 \delta^2) + 3H^2 \delta^2) \right]$.

The proof of Theorem 2 is finished. \qed

6. Experiments

In this section, the experiment for the image classification task is performed to evaluate the performance of the proposed algorithm. We first give the performance comparison between the proposed algorithm and other baseline algorithms. Then, we explore the effect of the number of local epochs and the number of clients on the algorithm performance.

We evaluate the performance of the proposed algorithm by conducting multi-class image classification tasks on the CIFAR-10 and CIFAR-100 datasets [38] with the convolutional neural network model RestNet-18 [39], respectively. The CIFAR-10 dataset consists of 60,000 $32 \times 32$ pixels color pictures, which include 50,000 training images and 10,000 test images in a total of 10 categories with 6000 images in each category. The CIFAR-100 dataset
has the same number of training samples and test samples with CIFAR-10, except it has 100 categories with 500 training images and 100 images test images per category. We first divide the training samples and test samples randomly and evenly among multiple clients. Based on the training set of each client, a local model is trained by an optimizer and sent to the centralized server. Then, the aggregated model is evaluated on a test set, which implies the generalization ability of the aggregated model. In the experiment, all the federated optimization algorithms including our proposed algorithm are implemented on a workstation equipped with two Intel(R) Xeon(R) Silver 4114 CPUs and two NVIDIA GeForce GTX 1080 Ti GPUs. The experiment environment is based on PyTorch 1.11.0. We consider a federated learning system with 100 clients in all and 0.1 as the ratio of client participation, which means 10 clients out of 100 are selected for training in each communication round. Specifically, each client randomly selected by the centralized server uses the same SGD optimizer and conducts 3 local epoch steps, and the local batch size is set to 20. The local learning rate and global learning rate are set to 0.01 and 1 by grid searching for the best performance. In addition, momentum factors $\beta_1, \beta_2$ are set to 0.9 and 0.99 by default, which follows the adaptive algorithms \[16,18\]. We compare the proposed algorithm with the classic federated average algorithm (FedAvg) \[4\] and federated adaptive-type algorithms \[17\] including FedAdam, FedYogi and FedAMS \[18\] as baselines. The experimental results are demonstrated in following figures.

Figures 1 and 2 provide the convergence results for FedDA and other baselines on the CIFAR-10 dataset and CIFAR-100 dataset with the RestNet-18 model in terms of training loss and test accuracy. In order to show the difference between different curves clearly, we zoom in on the last few rounds as subgraphs and place them in the blank space of the figure. For the CIFAR-10 dataset, from the convergence curve of training loss against the communication rounds in Figure 1, we find that FedDA goes down faster with the increase of communication rounds, which suggests faster convergence than other baselines. Specifically, FedDA has similar performance with FedYogi and FedAMS in the middle of training, while it works slightly better in training loss and has the highest test accuracy among all the baseline algorithms in the end. These results empirically demonstrate the effectiveness of the proposed algorithm. The similar experiment results are obtained from Figure 2 for the CIFAR-100 dataset. As shown, the proposed FedDA converges slightly faster at the beginning and final stages of training, and it achieves a higher test accuracy, which suggests that FedDA behaves well.

![Figure 1. Comparison for FedDA and other baselines on the CIFAR-10 dataset with the RestNet-18 model.](image-url)
As shown in Figure 3, the training loss and test accuracy of different client numbers are compared while other parameters remain the same. In this setting, we explore the impact of the client number $n$ involved in training on the performance of FedDA by choosing different parameters $n$ from \{5, 10, 20, 50\}. From the left figure of Figure 3, we observe that the convergence curve with $n = 50$ has the fastest convergence rate. On the other hand, a smaller $n$ requires more communication rounds to achieve the same target accuracy, as shown in the right figure of Figure 2.

In Figure 4, we investigate the impact of local epoch number $H$ on the performance of FedDA by training the RestNet-18 model on the CIFAR-10 dataset while keeping the other parameters the same. We choose different $H$ values from \{1, 3, 5, 10\}. From the learning curve of training loss against the communication round in Figure 4, we can find that the local training with a larger $H$ can help accelerate the convergence. From the right figure of Figure 4, we observe that to achieve the same target accuracy, a larger local epoch $H$ requires a smaller communication round.
7. Conclusions

In this paper, motivated by the idea of PID, we introduce the gradient difference to the stochastic optimization method and propose a differential adaptive method for federated learning, which aims to improve the performance while training the neural network by correcting the descent direction and adaptively updating the model parameters. We then analyze the convergence performance for our algorithm under full client participation and partial client participation cases in a stochastic non-convex optimization setting, where the convergence rate of $O(1/\sqrt{T})$ is achieved. Experiments are conducted on the CIFAR-10 dataset and CIFAR-100 dataset with the RestNet-18 model, respectively. The experiment results demonstrate that the proposed method FedDA and federated adaptive methods (FedYogi, FedAMS) have a better performance than FedAvg. FedDA surpasses the other baselines in terms of initial and final training loss, and it achieves the highest test accuracy in the end. Compared to other baselines, the proposed algorithm does not incur additional communication cost, while these performance gains are attained by increasing the computational cost of the difference terms. Thus, it is important to achieve a good trade-off between model performance and algorithm efficiency. In addition, the risk of information leakage exists when uploading model differences from clients to the centralized server. Hence, methods of privacy protection could be combined in the local training at the clients’ side. The balance between privacy levels and model accuracy can be considered as future research work.

**Author Contributions:** Conceptualization, H.G. and H.C.; methodology, H.G. and H.C.; software, X.Z. and Q.W.; validation, X.Z., J.Z. and M.Z.; formal analysis, H.G. and Q.W.; investigation, H.C. and M.Z.; resources, Q.W., X.Z., J.Z. and M.Z.; data curation, J.Z. and M.Z.; writing—original draft preparation, H.G. and Q.W.; writing—review and editing, Q.W. and J.Z.; visualization, X.Z.; supervision, J.Z. and M.Z.; project administration, J.Z. and M.Z.; funding acquisition, Q.W. and M.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants No. 61976243 and No. 61971458, in part by the Leading talents of science and technology in the Central Plain of China under Grant No. 224200510004, in part by the Science & Technology Innovation Talents in the University of Henan Province of China under Grant No. 22HASTIT014, in part by the basic research projects in the University of Henan Province, China under Grants No. 23ZX003, and in part by the International Cooperation Project of Henan Province under No. 232102521005.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.
Conflicts of Interest: The authors declare no conflict of interest.

References


33. Tong, Q.; Liang, G.; Bi, J. Calibrating the adaptive learning rate to improve convergence of ADAM. *Neurocomputing* 2022, 481, 333–356. [CrossRef] [PubMed]


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.