Convergence Analysis of Federated Learning Methods Using Backward Error Analysis

Jinwoo Lim[1], **Suhyun Kim**[2*], **Soo-Mook Moon**[1*]

[1]Seoul National University
[2]Korea Institute of Science and Technology
jinwoolim8180@snu.ac.kr, dr.suhyun.kim@gmail.com, smoon@snu.ac.kr

Abstract

Backward error analysis allows finding a modified loss function, which the parameter updates really follow under the influence of an optimization method. The additional loss terms included in this modified function is called *implicit regularizer*. In this paper, we attempt to find the implicit regularizer for various federated learning algorithms on non-IID data distribution, and explain why each method shows different convergence behavior. We first show that the implicit regularizer of FedAvg disperses the gradient of each client from the average gradient, thus increasing the gradient variance. We also empirically show that the implicit regularizer hampers its convergence. Similarly, we compute the implicit regularizers of FedSAM and SCAFFOLD, and explain why they converge better. While existing convergence analyses focus on pointing out the advantages of FedSAM and SCAFFOLD, our approach can explain their limitations in complex non-convex settings. In specific, we demonstrate that FedSAM can partially remove the bias in the first-order term of the implicit regularizer in FedAvg, whereas SCAFFOLD can fully eliminate the bias in the first-order term, but not in the second-order term. Consequently, the implicit regularizer can provide a useful insight on the convergence behavior of federated learning from a different theoretical perspective.

Introduction

Federated learning is a distributed learning technique in which a central server builds a global model by repeatedly aggregating the parameters of local models that clients update and upload. The most popular algorithm is FedAvg (McMahan et al. 2017), which aggregates the parameters simply by averaging them. Despite its privacy advantage, FedAvg suffers from lower accuracy and slower convergence due to a misalignment between local and global objectives, especially on non-IID data distribution (Zhao et al. 2018; Karimireddy et al. 2020b). Many algorithms, such as FedSAM (Caldarola, Caputo, and Ciccone 2022; Qu et al. 2022) and SCAFFOLD (Karimireddy et al. 2020b), have been proposed to reduce the drift in local updates and enhance performance. Also, there have been many researches to understand the difference of the convergence behavior with an analysis from the convex optimization perspectives (Wang et al. 2019; Yu, Yang, and Zhu 2019; Li et al. 2020; Khaled, Mishchenko, and Richtárik 2020; Haddadpour and Mahdavi 2019; Karimireddy et al. 2020b; Qu et al. 2022). However, it is still challenging to make a tight analysis on non-IID data. This paper attempts to analyze the convergence behavior of a few federated learning algorithms from a different theoretical perspective, *implicit regularization*.

The concept of implicit regularization was originally introduced to explain the generalization behavior of centralized *gradient descent* (GD) and *stochastic gradient descent* (SGD) (Barrett and Dherin 2021; Smith et al. 2021); unless otherwise stated, GD and SGD are for centralized in this paper. Their approach, inspired by *backward error analysis* (Hairer et al. 2006), is to analyze the path on which discrete parameter updates of GD or SGD lie. They found that the path of the gradient flow does not follow the original loss, but a *modified loss* under the influence of the small yet finite learning rate as well as the optimization method itself. The approximation of the modified loss function consists of two terms, the original loss term and the implicit regularizer term proportional to the learning rate. The implicit regularizer leads GD and SGD to flat minima by penalizing the norm of the gradient, thus contributing to the generalization.

Unlike GD and SGD, we found that its implicit regularizer works negatively for the convergence of FedAvg. If we assume that the number of local SGD steps (E) and the learning rate (η) are finite but small enough to disregard the high-order terms $O(E^3\eta^3)$ in the parameter updates, we found that the implicit regularizer term of FedAvg has a different form from SGD's. That is, the first-order term of the implicit regularizer contains a special term, which we call the *dispersion term*. When the modified loss is minimized, the dispersion term is also minimized, which increases the distance of each client's gradient from the average gradient. Independently from this variance, we found through experiments that the dispersion term affects the convergence of FedAvg. We also found that there is another term in the higher-order terms of the implicit regularizer (we call the *secondary dispersion term*), which can affect the convergence of FedAvg as well. So, both terms make the parameter updates deviate from the original path, which we call a *bias* in this paper.

We also obtained the implicit regularizer for variance reduction methods such as *FedSAM* (Caldarola, Caputo, and Ciccone 2022; Qu et al. 2022) and *SCAFFOLD* (Karim-

*Co-corresponding authors.
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ireddy et al. 2020b). We found that they mitigate the problem of the dispersion term by enforcing the gradients to be close to each other, thus automatically reducing the drifts. Their convergence analysis based on convex optimization clearly showed their advantage. However, our analysis based on the implicit regularizer of FedSAM and SCAFFOLD can show their limitations as well as their strengths. We can prove that FedSAM can partially remove the bias for the dispersion term. On the other hand, SCAFFOLD can completely remove the bias for the dispersion term, but not for the second-order terms of the implicit regularizer. Generally, the implicit regularizer-based analysis is useful since it can provide an insight on a more complex, non-convex setting where federated learning is often employed. Actually, there is an analysis of SCAFFOLD in a non-convex setting, which gave a good mathematical insight on its convergence speed (Karimireddy et al. 2020b). Compared to that, our analysis can provide a more intuitive insight on where it converges and why. For example, the implicit regularizer of FedAvg can also explain sharpness of the loss surface at the sub-optimal point where it converges.

Contributions. Our main contributions are as follows.

- This is the first work to analyze the implicit regularizer for federated learning on non-IID data, which can explain convergence on complex, non-convex settings.
- The implicit regularizer of FedAvg not only can explain its defect in convergence but also the sharpness of the loss surface where FedAvg converges.
- The implicit regularizer of FedSAM and SCAFFOLD can explain their limitations as well as their advantages.

Related Work

Convergence of FedAvg on non-IID data. It is widely known that FedAvg converges slower than centralized learning. Previous work such as Stich (2019), Patel and Dieuleveut (2019), and Khaled, Mishchenko, and Richtárik (2020) have analyzed the convergence behaviour of FedAvg under a different name, Local SGD. Such a slow convergence becomes more severe when the data distribution of the clients is non-IID (Karimireddy et al. 2020b). Many researches such as Zhao et al. (2018), Yu, Yang, and Zhu (2019), Wang et al. (2019), Haddadpour and Mahdavi (2019), and Li et al. (2020) have done a sharp analysis on such a situation, focusing on the asymptotic convergence speed of FedAvg in the best and worst cases. In addition to the slow convergence, Li et al. (2020) focused on the inherent bias of FedAvg. They showed that the path of parameter updates in FedAvg deviates from the path of SGD, deriving a model whose performance is lower than the SGD's.

Implicit regularization. Implicit regularization (Barrett and Dherin 2021) (Smith et al. 2021) is a key concept used in this paper. Using backward error analysis, one can find the path the gradient flow actually takes under the influence of implicit regularization, defined by a finite learning rate and an optimization method itself such as GD or SGD. In this work, we extend implicit regularization to the federated learning domain to analyze the gradient flow of FedAvg.

As a similar approach to our analysis, Glasgow, Yuan, and Ma (2022) and Gu et al. (2023) employ Stochastic Differential Equation-based approximation to analyze the gradient flow of FedAvg (Local SGD). Gu et al. (2023) observes the long-term behavior of local minimizers in FedAvg and concludes that the local minimizer of FedAvg is biased towards flat minima for IID data. Unlike this work, we focus on the short-term behavior of FedAvg on non-IID data.

Backward Error Analysis

The idea of backward error analysis

This whole subsection briefly explain the idea of backward error analysis done in Barrett and Dherin (2021) and Smith et al. (2021). The basic idea starts by considering GD as the integration of an ODE in the form $\dot{\omega} = -\nabla\mathcal{L}(\omega)$, called the *gradient flow*. Then discrete updates of GD can be seen as solving the integration problem with explicit Euler method of the first order, as $\omega(t+\eta) \approx \omega(t) - \eta\nabla\mathcal{L}(\omega(t))$. When the step size is finite, there will be a gap between a discrete solution of a GD step and the exact solution of the gradient flow equation. To bridge the gap, we introduce a modified flow of $\dot{\omega} = \tilde{f}(\omega)$ that the optimization such as GD really follows, where $\tilde{f}(\omega)$ is expressed in powers of step size η.

$$\tilde{f}(\omega) = f(\omega) + \eta f_1(\omega) + \eta^2 f_2(\omega) + \dots \quad (1)$$

This is viable when the step size η is finite but relatively small. Now, the role of backward error analysis is to find the function for each correction term $f_i(\omega)$. In Barrett and Dherin (2021), the second-derivative of the parameter is

$$\ddot{\omega}(t) = \nabla\tilde{f}(\omega(t))\dot{\omega}(t) = \nabla\tilde{f}(\omega(t))\tilde{f}(\omega(t)), \quad (2)$$

and by using this, we can obtain the perturbed parameter $\omega(t+\eta)$ taking the Taylor expansion of $\tilde{f}$.

$$\begin{aligned}&\omega(t+\eta)\\ =&\omega(t) + \eta\tilde{f}(\omega(t)) + \frac{\eta^2}{2}\nabla\tilde{f}(\omega(t))\tilde{f}(\omega(t)) + O(\eta^3)\end{aligned} \quad (3)$$

From Equation 1, it is possible to express $\tilde{f}(\omega(t))$ with the original function $f(\omega(t))$ and correction terms $f_1(\omega(t))$, $f_2(\omega(t)), \dots$, and modify Equation 3 as

$$\begin{aligned}\omega(t+\eta) = \omega(t) + \eta f(\omega(t)) + \eta^2(f_1(\omega(t))\\ + \frac{1}{2}\nabla f(\omega(t))f(\omega(t))) + O(\eta^3).\end{aligned} \quad (4)$$

In order to derive the correction terms, we need to ensure the equivalence between the parameter from the continuous modified flow and the parameter discretely updated during a single or multiple steps of optimization. For GD, as a simple example, the parameter after a single discrete update, $\omega_{t+1} = \omega_t - \eta\nabla\mathcal{L}(\omega_t)$, should be the same as the parameter, $\omega(t+\eta)$. Then $f(\omega)$ can be fixed as $-\nabla\mathcal{L}(\omega)$. Also, at order η^2, the coefficient must go to zero which means that

$$f_1(\omega) = -\frac{1}{2}\nabla\nabla\mathcal{L}(\omega)\nabla\mathcal{L}(\omega) = -\frac{1}{4}\nabla\|\nabla\mathcal{L}(\omega)\|^2. \quad (5)$$

Finally, it is possible to say that discrete GD iterates following the path of an ODE with the form of

$$\dot{\omega} = -\nabla\mathcal{L}(\omega) - (\eta/4)\nabla\|\nabla\mathcal{L}(\omega)\|^2 + O(\eta^2). \quad (6)$$

If the step size η is small enough to ignore the high order terms, the modified loss that the parameter updates of GD truly follows can be expressed as

$$\tilde{\mathcal{L}}_{GD}(\omega) \approx \mathcal{L}(\omega) + \frac{\eta}{4}\|\nabla\mathcal{L}(\omega)\|^2. \quad (7)$$

The modified loss is regarded as the the true loss function that the parameter updates of GD should minimize.

The same technique can be applied to SGD. In this case, rather than one step, a single epoch is taken into consideration to obtain the modified loss where $\nabla\mathcal{L}_k(\omega)$ is the loss of k-th mini-batch in E iterations of one epoch:

$$\tilde{\mathcal{L}}_{SGD}(\omega) \approx \mathcal{L}(\omega) + \frac{\eta}{4E}\sum_{k=0}^{E-1}\|\nabla\mathcal{L}_k(\omega)\|^2. \quad (8)$$

Backward error analysis for FedAvg

The same backward error analysis can be used to explain the convergence behavior of FedAvg. Unlike in central learning, in FedAvg, it is the client that trains the model with training samples. With their own data, the clients locally update their parameters with multiple steps (one step means one iteration of parameter update) and send their updated parameters to the server. What the server does is aggregating (averaging in FedAvg) those local parameters to produce new global parameters. A single iteration of training and aggregation is called a *round*. Therefore, unlike in GD where we try to match the solution of a single step, we try to match the solution of the multiple steps of one round to the modified continuous flow. Pseudocode for FedAvg is in Appendix.

We first define the necessary variables below, assuming that FedAvg runs with full participation of the clients.

Notations. The number of clients is m and the number of local steps in one round is E. The parameter is ω and local parameters are updated with a finite learning rate η. The loss function of the mini-batch sample from k-th step of j-th client is defined as $\mathcal{L}_{jk}(\omega)$. The mean loss function of each client is defined as $\mathcal{L}_j(\omega) = \frac{1}{E}\sum_{k=0}^{E-1}\mathcal{L}_{jk}(\omega)$, and $\mathcal{L}(\omega) = \frac{1}{m}\sum_{j=0}^{m-1}\mathcal{L}_j(\omega)$. $\nabla\mathcal{L}_j(\omega)$ is called a *client gradient* and $\nabla\mathcal{L}(\omega)$ is called the *global gradient* in this paper.

Now we assume that the learning rate η is large enough to make $O(E^2\eta^2)$ significant, yet too small to make $O(E^3\eta^3)$ significant. With these assumptions and definitions, we obtain the loss function modified under FedAvg as follows.

Theorem 1. *If local parameters of clients are discretely updated with a finite learning rate, the expectation of discrete updates of the aggregated parameter in FedAvg follows the modified loss $\tilde{\mathcal{L}}_{FedAvg}(\omega)$ which can be expressed as*

$$\tilde{\mathcal{L}}_{FedAvg}(\omega) \approx \mathcal{L}(\omega) - \frac{E\eta}{4m}\sum_{j=0}^{m-1}\underbrace{\|\nabla\mathcal{L}(\omega) - \nabla\mathcal{L}_j(\omega)\|^2}_{\textit{Dispersion term}} + \frac{\eta}{4mE}\sum_{j=0}^{m-1}\sum_{k=0}^{E-1}\|\nabla\mathcal{L}_{jk}(\omega)\|^2. \quad (9)$$

The approximation holds when $\eta \ll 1/E$. If $E = 1$, the modified loss is the same as the one of SGD.

Dispersion term. Unlike SGD, the implicit regularizer of FedAvg is composed of two terms. The latter term, which is the same as in the implicit regularizer of SGD, is the one known to aid generalization and help the converged model to achieve a high accuracy (Smith et al. 2021). One thing to note is that the latter term is affected by the size of each mini-batch, rather than the effective batch size (Lin et al. 2019) which is a size of a mini-batch multiplied by the number of local steps E. The former term, called *dispersion term* in this paper, is the one that makes a difference. Equation 9 indicates that when the modified loss is reduced (so is the the dispersion term), the dispersion term increases the distance between the client gradient and the global gradient. The presence of the dispersion term can affect convergence severely, which we will show through experiments.

Dispersion term and sharp minima. Many optimization problems such as minimization of cross-entropy can be regarded as a Maximum Likelihood Estimation problem. Such a problem setting makes an interesting point about the Hessian of the loss. If we assume that the current parameter ω is close to an optimum and the outputs of a model in the current parameter are almost identical to the ground-truth, the Hessian of the loss can be approximated as Fisher information matrix. Moreover, since the loss gradient is almost zero nearby optima, we can build a special equivalence of the trace of Fisher information matrix and the gradient variance of samples. Since we can approximate the Hessian of loss as Fisher information matrix, we can deduce that the trace of Hessian can be approximated by the gradient variance of samples (Rame, Dancette, and Cord 2022). If the number of samples in the dataset is N and the loss gradient of the i-th sample is $\nabla\tilde{\mathcal{L}}_i(\omega)$, the trace of Hessian is approximated as

$$\mathrm{tr}(\nabla\nabla\mathcal{L}(\omega)) \approx \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\|\nabla\tilde{\mathcal{L}}_i(\omega) - \nabla\mathcal{L}(\omega)\|^2 \quad (10)$$

Meanwhile, if one considers that the dispersion term of FedAvg on non-IID data increases the gradient variance, it is possible to build another link between FedAvg and the Hessian trace. When the parameter is nearby optima, the dispersion term of FedAvg can increase the trace of Hessian. Since the trace of Hessian can be a measure for *sharpness* of the loss surface (Ma and Ying 2021), it can be stated that the dispersion term leads the parameter to sharp minima (later, we will see an opposite characteristics of FedAvg, though).

Backward error analysis for FedSAM

While the dispersion term of FedAvg on non-IID data drives the parameter to converge into sharper minima, there is a well-known technique called *sharpness-aware minimization* (SAM) that can produce an opposite effect. SAM is a technique that alters the loss function into a form of

$$\mathcal{L}_{SAM}(\omega) = \mathcal{L}(\omega + \varepsilon\nabla\mathcal{L}(\omega)) \text{ where } \varepsilon \ll 1 \quad (11)$$

and it is already well-known that SAM increases the convergence speed of FedAvg (Caldarola, Caputo, and Ciccone 2022; Qu et al. 2022). However, we analyze the implicit regularization of FedSAM to show that SAM acts as a variance reduction method by partially removing the dispersion term, rather than naïvely leading the parameter to flatter minima.

For clarity, we refer to an approximation method in Zhao, Zhang, and Hu (2022) and Geiping et al. (2022) that is used for computation of Hessian-vector product. Through a finite-difference approximation, the corresponding gradient to a gradient norm penalty is computed as $\nabla \frac{1}{2}\|\nabla \mathcal{L}(\omega)\|^2 \approx \frac{\nabla \mathcal{L}(\omega+\varepsilon\nabla\mathcal{L}(\omega))-\nabla\mathcal{L}(\omega)}{\varepsilon}$ when the magnitude of ε is very small like $0.01/\|\nabla \mathcal{L}(\omega)\|$ as in Foret et al. (2021). If we put this to a loss as an explicit regularizer with coefficient λ,

$$\begin{aligned} &\mathcal{L}(\omega) + \tfrac{\lambda}{2}\|\nabla \mathcal{L}(\omega)\|^2 \\ &\approx \tfrac{\lambda}{\varepsilon}\mathcal{L}(\omega + \varepsilon \nabla \mathcal{L}(\omega)) + (1 - \tfrac{\lambda}{\varepsilon})\mathcal{L}(\omega) \end{aligned} \tag{12}$$

If $\lambda = \varepsilon$, the cost function is the same as the one of SAM. This implies that SAM is equivalent to training a model with a gradient norm penalty. However, ε of SAM could be smaller than $E\eta/2$ if the norm of the gradient is large. Also, FedSAM gives penalty to the mini-batch gradients instead of the norm of an average gradient of mini-batches of a client. In this case, if a subsidiary implicit regularization caused by gradient penalty itself can be ignored, the result below holds.

Corollary 2. *If local parameters are updated with a finite learning rate, the expected path of the global parameter updates of FedSAM approximately follows the modified loss* $\tilde{\mathcal{L}}_{FedSAM}(\omega)$ *as*

$$\begin{aligned} \tilde{\mathcal{L}}_{FedSAM}(\omega) &\approx \mathcal{L}(\omega) + \frac{\varepsilon}{2}\|\nabla \mathcal{L}(\omega)\|^2 \\ &- (\frac{E\eta}{4m} - \frac{\varepsilon}{2m}) \sum_{j=0}^{m-1} \|\nabla \mathcal{L}(\omega) - \nabla \mathcal{L}_j(\omega)\|^2 \\ &+ \frac{\varepsilon}{2mE} \sum_{j=0}^{m-1}\sum_{k=0}^{E-1} \|\nabla \mathcal{L}_j(\omega) - \nabla \mathcal{L}_{jk}(\omega)\|^2 \end{aligned} \tag{13}$$

The approximation holds when $\eta \ll 1/E$.

The modified loss of FedSAM is under the influence of ε. It can be observed that the presence of ε reduces the magnitude of the dispersion term. This indicates that FedSAM can function as a variance reduction method that decreases the dispersion of gradients. It is important to note, however, that if ε is smaller than $E\eta/2$, such effects by ε will only *partially* remove the dispersion term and might not sufficiently enhance the performance of the model as other variance reduction methods. We will evaluate it through experiments.

Backward error analysis for SCAFFOLD

Another simple solution to mitigate the effect of the dispersion term would be enforcing the gradients to be close to each other. A variance reduction method such as SCAFFOLD (Karimireddy et al. 2020b) is the one. It utilizes the control variates to remove the dispersion effect of FedAvg. The convergence behavior of SCAFFOLD is quite different from FedAvg's, and as done for FedAvg, a backward error analysis can be done for SCAFFOLD to explain its convergence behavior. For convenience, we again assume full participation of all clients. If SCAFFOLD is under an ideal situation and the learning rate is small enough to make $O(E^2\eta^2)$ negligible, we obtain the modified loss of SCAFFOLD.

Theorem 3. *If local parameters of clients are discretely updated with a finite learning rate, the expectation of discrete updates of the aggregated parameter in SCAFFOLD follows the modified loss* $\tilde{\mathcal{L}}_{SCAFFOLD}(\omega)$ *which is expressed as*

$$\begin{aligned} \tilde{\mathcal{L}}_{SCAFFOLD}(\omega) &\approx \mathcal{L}(\omega) + \frac{\eta}{4}\|\nabla \mathcal{L}(\omega)\|^2 \\ &+ \frac{\eta}{4mE}\sum_{j=0}^{m-1}\sum_{k=0}^{E-1} \|\nabla \mathcal{L}_j(\omega) - \nabla \mathcal{L}_{jk}(\omega))\|^2 \end{aligned} \tag{14}$$

The approximation holds when $\eta \ll 1/E$.

The implicit regularizer of SCAFFOLD consists of two terms, $\frac{\eta}{4}\|\nabla \mathcal{L}(\omega)\|^2$ and $\frac{\eta}{4mE}\sum_{j=0}^{m-1}\sum_{k=0}^{E-1}\|\nabla \mathcal{L}_j(\omega) - \nabla \mathcal{L}_{jk}(\omega))\|^2$. The former term has the same form as the implicit regularizer of GD and penalizes the norm of the global gradient. The latter penalizes the trace of covariance matrix of local mini-batch gradients from each client, unlike the implicit regularizer of SGD that penalizes the trace of covariance matrix of all mini-batch gradients (Smith et al. 2021).

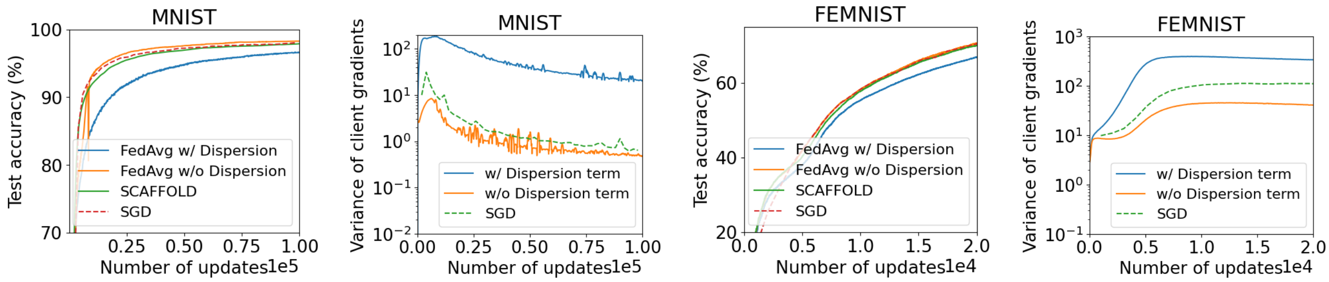
The most notable point is that the dispersion term is *absent* in the modified loss of SCAFFOLD. SCAFFOLD is mostly not affected by the effect of the dispersion term and its convergence behaviour resembles SGD more than FedAvg, which can be verified in the experiments.

Empirical analysis

One way to inspect the effect of the dispersion term is to compare the convergence behaviour of FedAvg with and without the dispersion term. In order to empirically check the effect of the dispersion term, we manually removed the dispersion term from the modified loss. The algorithm for removing the dispersion term is in the Appendix.

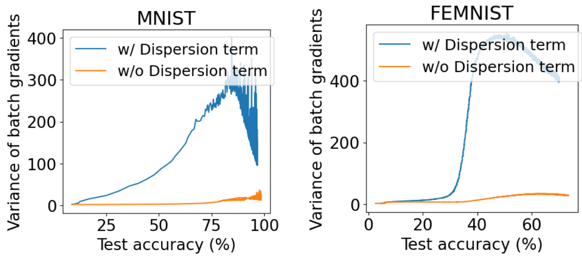
Empirical analysis on the dispersion term. We run experiments for evaluation of our analysis. To evaluate only the effect of the dispersion term, we run experiments with a simple CNN model on a simple dataset, MNIST (LeCun et al. 1998) and a relatively more complex dataset, FEMNIST (Caldas et al. 2018). Experiments were done on a non-IID environment of Dirichlet distribution with parameter 0.2, except for FEMNIST, which is naturally non-IID. The batch size was 30 for MNIST and 100 for FEMNIST. The effective batch size mentioned in **Dispersion term.** was not applied to SGD and the batch size was set the same for both FedAvg and SGD. This is to make the implicit regularizer of FedAvg the same as the one of SGD other than the dispersion term. To fully observe the effect under the gradient flow, we used a normal SGD optimizer with a small learning rate of 0.001 with no momentum and learning rate decay. More details on the experimental settings such as the model architecture and the learning rate are in the Appendix.

The empirical results are summarized in Figure 1. Overall, FedAvg without the dispersion term converged faster than the original FedAvg and the performance was identical to the ones of SGD and SCAFFOLD, whose modified loss is similar to the one of FedAvg without the dispersion term. This indicates that the dispersion term is the main reason for performance degradation of FedAvg when the size



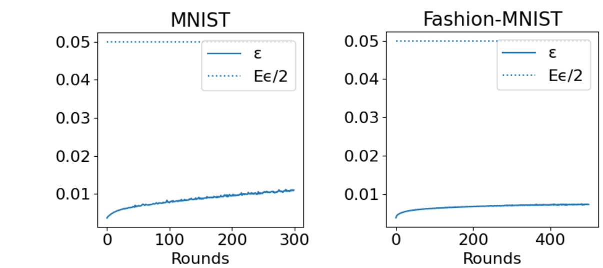
(a) (b) (c) (d)

Figure 1: Test accuracy and variance of client gradients of FedAvg, SCAFFOLD, and SGD on MNIST, and FEMNIST. The final test accuracy is higher and the variance of client gradients is significantly lower when the dispersion term is absent in the modified loss. The convergence behaviours of SCAFFOLD, SGD, and FedAvg without dispersion term are almost identical.



(a) (b)

Figure 2: Variance of mini-batch gradients in MNIST and FEMNIST.



(a) (b)

Figure 3: The value of ε of FedSAM on MNIST and Fashion-MNIST. ε is consistently lower than $E\eta/2$.

of a learning rate is small enough. Also, the variance of the client gradients, measured at the beginning of the round, was much higher when there was a dispersion term in the modified loss, which matches our theoretical observation. These result indicates that our analysis on the first-order term of implicit regularizer can explain the convergence behaviour of FedAvg and SCAFFOLD. However, one thing to note is that the main reason for performance degradation of FedAvg is *not the noise of gradients* due to the increased gradient variance, but rather the presence of the implicit regularizer itself. We will discuss this in later sections.

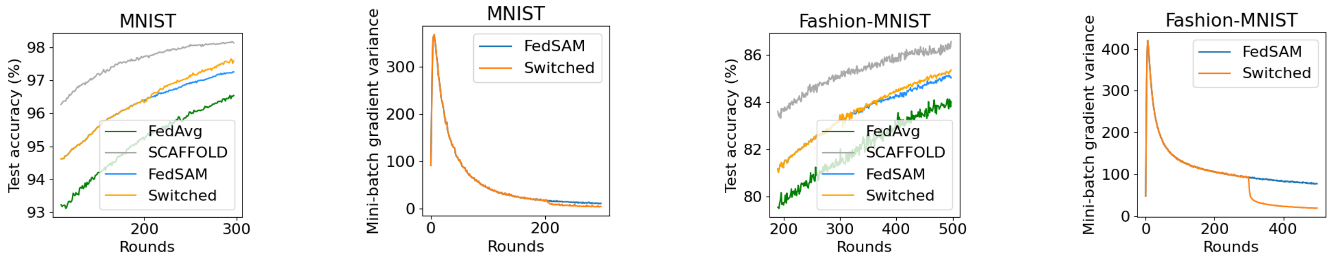
Implicit bias of FedAvg. In Figure 1 and 2, FedAvg with the dispersion term has higher variances of mini-batch gradients as well as client gradients than FedAvg without it, even when the performance of the model is the same. If the updates of FedAvg with the dispersion term slowly but surely followed the same path as FedAvg without the dispersion term, the variance of gradients should have been the same when the performance of the model was the same. However, the difference in variance implies that the parameter updates of FedAvg deviated from the original path due to the bias by implicit regularization. Such a deviation heavily affects the convergence behavior and leads FedAvg to converge into a sub-optimal solution as pointed out by Li et al. (2020).

Empirical analysis on FedSAM. We experimented with FedSAM on MNIST and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017) on non-IID with full client participation. We used simple datasets to observe the effect of the dispersion term without a disturbance of batch gradient variance. More details on the experimental settings are in the Appendix.

We first investigated the value of ε during a normal training with FedSAM. The value of ε was set as the value similar to $0.01/\sqrt{\|\nabla\mathcal{L}_{jk}(\omega)\|}$, which was much more stable than $0.01/\|\nabla\mathcal{L}_{jk}(\omega)\|$. A detailed explanation is in the pseudocode in the Appendix. As shown in Figure 3(a) and 3(b), the value of ε stayed below $E\eta/2$.

Next, we examined the performance of FedSAM. We checked if FedSAM could perform as well as SCAFFOLD even when the value of ε stays below $E\eta/2$. As shown in Figure 4(a) and 4(c), FedSAM was able to perform better than FedAvg but not as well as SCAFFOLD. This accords with our prediction that FedSAM would only partially mitigate the dispersion effect if ε is not large enough, while SCAFFOLD is able to almost remove the dispersion term.

To confirm if the slow convergence of FedSAM is due to the insufficient magnitude of ε, we changed the value of ε in Equation 13 to $E\eta/2$ during training, which makes the magnitude of the dispersion term zero. We increased the value of ε to $E\eta/2$ in the middle of training session when the gradients are relatively stable. We did not change ε in the



(a) (b) (c) (d)

Figure 4: Test accuracy, value of ε, and client gradient variance of FedSAM on MNIST and Fashion-MNIST. ε was switched to $E\eta/2$ during training and the convergence speed became faster while the variance of mini-batch gradients decreased. The values at the exact round where switching occurred were omitted for smoothing of the graphs.

early stage of training since the gradient variance tends to increase rapidly in the early stage and the gradients become extremely unstable. We switched ε at the 200-th round on MNIST and at the 300-th round on Fashion-MNIST.

As shown in Figure 4(a), 4(c), the performance of FedSAM with switched ε surpassed the performance of the original FedSAM due to a faster convergence speed. The result indicates that the convergence speed becomes faster when ε is as big as $E\eta/2$ and the dispersion term is fully removed. Also, as shown in 4(b) and 4(d), the dispersion term was mitigated, and the variance of client gradients was decreased after switching, consistent with our analysis.

High-order Terms in the Implicit Regularizer

Meanwhile, one thing was unexplained in the empirical results on the dispersion term. In Figure 1, the variance of client gradients in SGD was higher than the one of FedAvg without the dispersion term. It means that there is still an unexplained bias of FedAvg which reduces the client gradient variance. Here, we analyze the implicit regularizer of FedAvg with *high-order terms* and show that those high-order terms can reduce the client gradient variance but hamper the convergence of FedAvg. For our analysis, we use the notion of *local epochs*. To reduce the communication cost between the client and the server, federated learning often employs local epochs. That is, clients iterate over their entire local data multiple times, and one full iteration is called one *local epoch*. Since the presence of multiple local epochs increase the number of total local steps E, the high-order terms $O(E^3\eta^3)$ become significant.

Here, the number of total local steps E can be obtained with two factors: the number of local epochs and the number of local steps within a single local epoch, defined as a and K, respectively. In this paper, we deal with a case where the number of steps per a local epoch, K, is small enough to ignore $O(K^3\eta^3)$, while the number of local epochs, a, is large enough to make $O(a^3K^3\eta^3)$ significant. Now, we ignore high-order terms within a local epoch such as $O(K^3\eta^3)$, while keeping $O(a^3K^3\eta^3)$ to get a more precise, modified loss with the second-order terms in the implicit regularizer.

Corollary 4. *If local parameters are updated with a finite learning rate for multiple local epochs, the expectation of discrete updates of the aggregated parameter in FedAvg follows the modified loss $\tilde{\mathcal{L}}(\omega)$ as*

$$\tilde{\mathcal{L}}(\omega) \approx \mathcal{L}(\omega) - \frac{aK\eta}{4m}\sum_{j=0}^{m-1} \underbrace{\|\nabla\zeta_j(\omega - \frac{aK\eta}{3}\nabla\mathcal{L}_j(\omega))\|^2}_{\textit{Transformed dispersion term}}$$

$$+ \frac{a^2K^2\eta^2}{6m}\sum_{j=0}^{m-1} \underbrace{\nabla\zeta_j(\omega)^\top\nabla\nabla\mathcal{L}(\omega)\nabla\zeta_j(\omega)}_{\textit{Secondary dispersion term}}. \quad (15)$$

when $\nabla\zeta_j(\omega) = \nabla\mathcal{L}(\omega) - \nabla\mathcal{L}_j(\omega)$ and $\nabla\nabla\mathcal{L}(\omega)$ denotes the Hessian of the loss. The approximation holds when $1/E^2 \ll \eta \ll 1/E$ and $a \gg 1$.

Now, the implicit regularizer of FedAvg has been transformed. The first term of the implicit regularizer, which we call the *transformed dispersion term*, is now the client gradient variance at $\omega - \frac{aK\eta}{3}\nabla\mathcal{L}_j(\omega)$ instead of at ω. Due to a transformation of the dispersion term, the modified loss no longer maximizes the gradient variance at the current parameter ω. This will reduce the effect of the original dispersion term on increasing the gradient variance. On the other hand, we refer to the latter term of implicit regularizer as the *secondary dispersion term*. One thing to note is that secondary dispersion term is a quadratic objective function. When the loss Hessian is positive-semidefinite, the quadratic term minimizes the gap between the client gradients and the global gradient.

So, both the secondary dispersion term and the difference of the dispersion term due to transformation can reduce the variance of client gradients. Experimental results below will confirm that these high-order terms of the implicit regularizer can reduce the gradient variance. Moreover, from the link between sharpness of a loss surface and the gradient variance, it can be stated that the secondary dispersion term can lead FedAvg to flat minima. Such an effect by high-order terms contradicts the effect of the dispersion term. The dispersion term leads FedAvg to sharp minima when the number of local steps is small as mentioned previously, while high-order terms lead FedAvg to relatively flatter minima when the number of local steps is large. The latter aligns with Gu et al. (2023) which proves that local steps in Local SGD lead the parameter to flat minima.

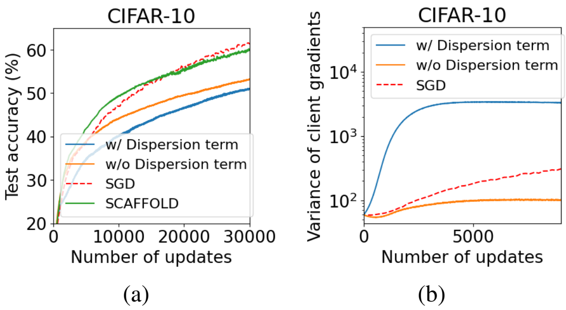


Figure 5: Test accuracy and client gradient variance for a complex model and dataset. Compared to SGD that no dispersion terms, FedAvg without the first-order dispersion term has a lower accuracy but a lower variance, while SCAFFOLD has a slightly lower accuracy, due to high order terms.

Gradient variance and performance. However, reduction of gradient variance and convergence into flat minima do not necessarily lead to better performance, as we will see in the experiments. This point differs from FedSAM which also converges to flat minima. SAM partially but explicitly removes the dispersion term, whereas high-order terms of the implicit regularizer such as the transformed dispersion term in Equation 15 can actually induce an additional bias to FedAvg. Although the high-order terms of its implicit regularizer reduce the gradient variance and lead FedAvg to flat minima, they still make parameter updates deviate from the optimal path, thus affecting convergence. On the other hand, the fact that a reduction of gradient variance can lead to degradation of performance implies that the noise of the gradients from the increased gradient variance is not a critical factor in convergence of FedAvg. This is contrary to a misconception that addition of variance in gradients is a primary reason for performance degradation of FedAvg. Though the addition of variance will certainly affect the convergence speed, the empirical results show that another factor, the bias of FedAvg, is a more critical factor.

Limits of variance reduction methods. Meanwhile, the fact that high-order terms in the implicit regularizer hamper the convergence is a thought-provoking point to all variance reduction methods that use stagnant gradients as the control variates. Traditional variance reduction methods such as SCAFFOLD use the client gradients from a *previous* round to correct deviation of client gradients and mitigate the effect of the dispersion term. However, the client gradient that composes the transformed dispersion term is actually from the *middle* of the *current* round. Such a discrepancy will inevitably cause a performance degradation, which can be a limitation of variance reduction methods. Theoretical explanation of the limitation is in the Appendix.

Empirical analysis on high-order terms

We ran experiments on a rather complex dataset, CIFAR-10 (Krizhevsky, Hinton et al. 2009), for a model with residual connections to evaluate the impact of high-order terms of the implicit regularizer (small dataset and simple model do not show the impact). Data is non-IID with a Dirichlet distribution of parameter 0.05, more extreme than the previous experiment. 100 clients were trained with a learning rate of 0.001, 3 local epochs, and the batch size of 300. We experimented with FedAvg with and without the first-order dispersion term, SCAFFOLD, and SGD. Details on experimental settings are in the Appendix.

Figure 5 shows that the gradient variance of SGD, which naturally lacks the dispersion term and the secondary dispersion term in its implicit regularizer, is higher than that of FedAvg without the dispersion term. This matches our analysis that the high-order terms of the implicit regularizer contributes to reducing the gradient variance. On the other hand, the convergence speed of SGD is higher although its gradient variance is higher. This indicates that the increased gradient variance itself is not the primary reason for slow convergence of FedAvg. The result also indicates that the high-order terms play a significant role in hampering convergence of FedAvg when a model is complex.

On the other hand, Figure 5 shows that the performance of SCAFFOLD is much higher than FedAvg's without the dispersion term and almost close to SGD's. This was expected since SCAFFOLD in Equation 14 lacks the dispersion term of FedAvg in Equation 9. However, the performance of SCAFFOLD is still inferior to SGD's. Considering that slow-down from the batch gradient variance is larger in SGD due to more parameter updates in a single epoch, it can be concluded that SCAFFOLD cannot fully mitigate the bias from the high-order terms of its implicit regularizer. It can be severe when a model becomes complex to make high-order terms significant, which has been continuously observed in Reddi et al. (2020); Karimireddy et al. (2020a); Yu et al. (2022).

Limitation and Conclusion

As in the previous work by Smith et al., one condition required to validate our analysis is that the learning rate should be small enough to make high-order terms insignificant. Such an assumption is an extreme condition as pointed out by Smith et al.. However, a small learning rate has been commonly assumed in many federated learning researches. For example, many assume that the local learning rate is smaller than the reciprocal of the product of the number of local updates and the smoothness of the loss function (Karimireddy et al. 2020b; Qu et al. 2022; Xu et al. 2021). Noting that the smoothness is the maximum eigenvalue of a Hessian, our assumption is not really extreme compared to their work.

Our goal was to understand the dynamics of FedAvg and variance reduction methods in a more intuitive way than existing convex optimization-based analysis. Despite limitation of our assumptions, we effectively analyzed the additional implicit regularization posed federated learning methods. We found that the presence of the dispersion term and secondary dispersion term of the implicit regularizer of FedAvg is the main reason that affect its the convergence, rather than the noise from the increased gradient variance itself. We also analyzed the fundamental limitations of existing variance reduction methods. Empirical results confirmed our theoretical observations.

Acknowledgements

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by Korean Government [Ministry of Science and ICT (MSIT)] under Grant RS-2023-00208245, 30%; in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by Korean Government (MSIT) under Grant 2021-0-00180, 20% and Grant 2021-0-00136, 20%; in part by the Information Technology Research Center (ITRC) support Program Supervised by IITP under Grant IITP-2021-0-01835, 20%; and in part by IITP under Artificial Intelligence Semiconductor Support Program to Nurture the Best Talents under Grant IITP-2023-RS-2023-00256081, 10%.

References

Barrett, D.; and Dherin, B. 2021. Implicit Gradient Regularization. In *International Conference on Learning Representations*.

Caldarola, D.; Caputo, B.; and Ciccone, M. 2022. Improving generalization in federated learning by seeking flat minima. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, 654–672. Springer.

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

Foret, P.; Kleiner, A.; Mobahi, H.; and Neyshabur, B. 2021. Sharpness-aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representations*.

Geiping, J.; Goldblum, M.; Pope, P.; Moeller, M.; and Goldstein, T. 2022. Stochastic Training is Not Necessary for Generalization. In *International Conference on Learning Representations*.

Glasgow, M. R.; Yuan, H.; and Ma, T. 2022. Sharp bounds for federated averaging (local sgd) and continuous perspective. In *International Conference on Artificial Intelligence and Statistics*, 9050–9090. PMLR.

Gu, X.; Lyu, K.; Huang, L.; and Arora, S. 2023. Why (and When) does Local SGD Generalize Better than SGD? In *The Eleventh International Conference on Learning Representations*.

Haddadpour, F.; and Mahdavi, M. 2019. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*.

Hairer, E.; Hochbruck, M.; Iserles, A.; and Lubich, C. 2006. Geometric numerical integration. *Oberwolfach Reports*, 3(1): 805–882.

Karimireddy, S. P.; Jaggi, M.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2020a. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606*.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020b. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 5132–5143. PMLR.

Khaled, A.; Mishchenko, K.; and Richtárik, P. 2020. Tighter theory for local SGD on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, 4519–4529. PMLR.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

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

Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2020. On the Convergence of FedAvg on Non-IID Data. arXiv:1907.02189.

Lin, T.; Stich, S. U.; Patel, K. K.; and Jaggi, M. 2019. Don't Use Large Mini-batches, Use Local SGD. In *International Conference on Learning Representations*.

Ma, C.; and Ying, L. 2021. On linear stability of sgd and input-smoothness of neural networks. *Advances in Neural Information Processing Systems*, 34: 16805–16817.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Patel, K. K.; and Dieuleveut, A. 2019. Communication trade-offs for synchronized distributed SGD with large step size. *arXiv preprint arXiv:1904.11325*.

Qu, Z.; Li, X.; Duan, R.; Liu, Y.; Tang, B.; and Lu, Z. 2022. Generalized federated learning via sharpness aware minimization. In *International Conference on Machine Learning*, 18250–18280. PMLR.

Rame, A.; Dancette, C.; and Cord, M. 2022. Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, 18347–18377. PMLR.

Reddi, S.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konečnỳ, J.; Kumar, S.; and McMahan, H. B. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

Smith, S. L.; Dherin, B.; Barrett, D.; and De, S. 2021. On the Origin of Implicit Regularization in Stochastic Gradient Descent. In *International Conference on Learning Representations*.

Stich, S. U. 2019. Local SGD Converges Fast and Communicates Little. In *International Conference on Learning Representations*.

Wang, S.; Tuor, T.; Salonidis, T.; Leung, K. K.; Makaya, C.; He, T.; and Chan, K. 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE journal on selected areas in communications*, 37(6): 1205–1221.

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

Xu, J.; Wang, S.; Wang, L.; and Yao, A. C.-C. 2021. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874*.

Yu, H.; Yang, S.; and Zhu, S. 2019. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 5693–5700.

Yu, Y.; Wei, A.; Karimireddy, S. P.; Ma, Y.; and Jordan, M. 2022. TCT: Convexifying federated learning using bootstrapped neural tangent kernels. *Advances in Neural Information Processing Systems*, 35: 30882–30897.

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

Zhao, Y.; Zhang, H.; and Hu, X. 2022. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International Conference on Machine Learning*, 26982–26992. PMLR.