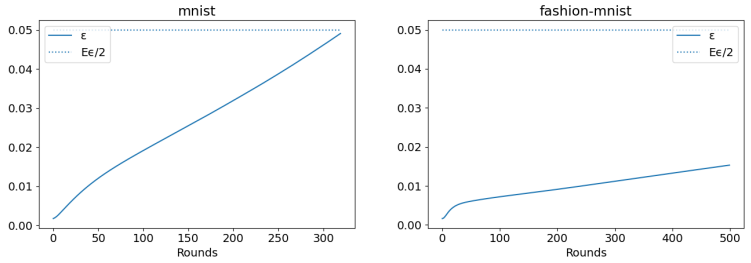
A CONVERGENCE BEHAVIOUR OF FEDSAM

The modified loss of FedSAM has the form below

$$\tilde{C}_{FEDSAM}(\omega) = C(\omega) + \frac{\varepsilon}{2mn}\sum_{i=0}^{n-1}\|\nabla C_i(\omega)\|^2 - (\frac{E\epsilon}{4mn} - \frac{\varepsilon}{2mn})\sum_{j=0}^{m-1}\|\nabla C_i(\omega) - \nabla C_{ij}(\omega)\|^2 + \frac{\varepsilon}{2mn}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\sum_{k=0}^{E-1}\|\nabla C_{ij}(\omega) - \nabla C_{ijk}(\omega)\|^2$$

if we ignore the subsidiary implicit regularizer caused by frequent application of mini-batch gradient norm penalty. The implicit regularizer for FedSAM is composed of three terms: a generalizing term that penalizes the gradient norm, a drifting term that disperses the client gradients, and a new term. One thing to notice is that all three terms depend on the variable ε. The presence of ε in the drifting term decreases the magnitude of the drifting term, reducing dispersion of client gradients. Also, it is possible to predict that the variance of mini-batch gradients will decrease when the new term, with a form of $\frac{\varepsilon}{2mn}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\sum_{k=0}^{E-1}\|\nabla C_{ij}(\omega) - \nabla C_{ijk}(\omega)\|^2$, is minimized.

We empirically checked the effect of those terms by changing the value of ε to $E\epsilon/2$, which can make the magnitude of the drifting term zero while increasing the magnitude of the new term. We have done experiments on MNIST and Fashion-MNIST on non-IID settings and full client participation. The learning rate was 0.001 and we trained the model for 300 rounds for MNIST, 500 rounds for Fashion-MNIST. The model was different from the main experiments: we used a CNN model consisting of 2 convolutional layers with 32 and 64 7×7 filters and a softmax layer for all three experiments, which is a model bigger than the model used in previous experiments for MNIST and Fashion-MNIST. We used a larger model for stability of mini-batch gradients.

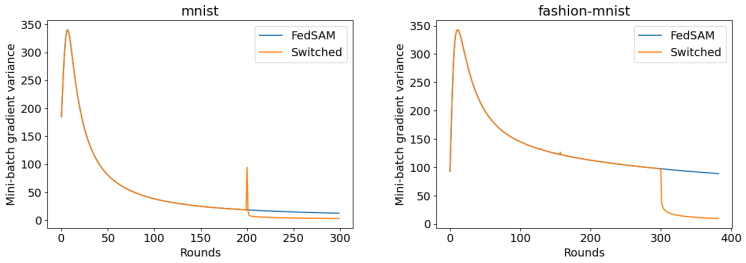


(a) (b)

Figure A.1: The value of ε during training. Training was done on (a) MNIST and (b) Fashion-MNIST with non-IID settings and full client paricipation. The dotted line denotes $E\epsilon/2$. It is possible to see that ε is smaller than $E\epsilon/2$.

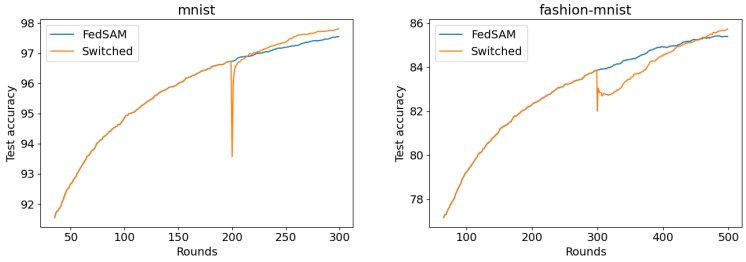
First thing done was investigating the value of ε during a normal training with FedSAM. The value of ε was set as $0.01/\|\nabla C_{ijk}(\omega)\|$. As shown in Figure A.1, while the value of ε mildly fluctuated during training, the value stayed below $E\epsilon/2$, which accords with our assumption on the value of ε. Next thing done was inspecting the effect of the mini-batch gradient variance. One attempt we made was to change the value of ε to $E\epsilon/2$ in the early stages of training. As a result, the gradient exploded and the loss became NaN, which means that the mini-batch gradient variance is heavily affecting the convergence behaviour. One of the reason for such an explosion was due to the increased variance of the mini-batch gradients. As shown in Figure A.2, the variance of mini-batch gradients rapidly increased in the early stages of training and slowly decreased in the later stages. The increased magnitude of the term $\sum_{j=0}^{m-1}\sum_{k=0}^{E-1}\|\nabla C_{ij}(\omega) - \nabla C_{ijk}(\omega)\|^2$ could heavily affect the gradient variance in the early stages.

Therefore, in additional experiments, we switched the value of ε at the late stage of training where the gradient variance is reduced and the training procedure is stable. We switched the value at 200-th round on MNIST and 300-th round on Fashion-MNIST. As a result, though the convergence behaviour became extremely unstable initially after switching, the performance quickly catched up



(a) (b)

Figure A.2: Mini-batch gradient variance during training of (a) MNIST and (b) Fashion-MNIST. Though the variance may rapidly increase initially after the switch, the variance decreases and becomes smaller than the one of FedSAM in later stages.



(a) (b)

Figure A.3: Test accuracy(%) during training of (a) MNIST and (b) Fashion-MNIST. We switched the value of ε to $E\epsilon/2$ at 200-th round for MNIST, 300-th round for Fashion-MNIST. Though the convergence behaviour became extremely unstable initially after switching, the performance quickly catched up to or surpassed the ones of FedSAM.

to or surpassed the ones of FedSAM. In spite of a huge gap between the value of ε and $E\epsilon/2$ in late rounds, the performance was similar or better. The results show that the drifting effect is mitigated more when ε is as big as $E\epsilon/2$.

Through a backward error analysis and experiments, we concluded that FedSAM enhances the convergence behaviour by mildly and gradually removing the drifting term of the modified loss, which is similar to performing an 'interpolation' between FedAvg and FedSGD.

B How a drifting term affects the convergence speed of FedAvg

In this section, we do not explicitly show an exact bound of convergence rate of FedAvg. Instead, we provide a brief overview how a drifting term slows the convergence speed of FedAvg. For ease of analysis, we only consider a situation where all clients participate in all rounds. Ignoring a generalizing term, we consider the following optimization problem:

$$\min_{\omega} \left\{ \tilde{C}(\omega) := C(\omega) - \frac{E\epsilon}{4m} \sum_{j=0}^{m-1} \|\nabla C(\omega) - \nabla C_j(\omega)\|^2 \right\} \tag{33}$$

Then we make a few assumptions on cost functions $C_0, \ldots, C_{m-1}$ for an easier analysis.

Assumption 1. *Cost functions $C_0, \ldots, C_{m-1}$ are all L-smooth: $\|\nabla C_j(x) - \nabla C_j(y)\|_2 \leq L\|x - y\|_2$ for any x, y and $L > 0$.*

Assumption 2. *The learning rate or step size ϵ is bounded: $\epsilon \leq 1/L$.*

Assumption 3. *The gradient norm of the variance of cost functions divided by the gradient norm of the average cost function is bounded:* $\frac{\|\nabla \mathrm{Var}[\nabla C_j(\omega)]\|_2}{\|\nabla C(\omega)\|_2} = \frac{\|\frac{1}{m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega)-\nabla C_j(\omega)\|_2^2\|_2}{\|\nabla C(\omega)\|_2} \leq \tilde{\sigma}$.

Overview. Our assumption that cost functions are L-smooth implies that the average cost function C is also L-smooth. This property leads to the following inequality:

$$C(y) \leq C(x) + \langle \nabla C(x), y - x\rangle + \frac{L}{2}\|y - x\|_2^2 \tag{34}$$

If we let $x = \omega$ and $y = \omega^+ = \omega - \epsilon\nabla\tilde{C}(\omega) = \omega - \epsilon\nabla C(\omega) + \frac{E\epsilon^2}{4m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2$, which is a one-step update of gradient descent from ω, we then get

$$\begin{aligned}
C(\omega^+) &= C(\omega - \epsilon\nabla C(\omega) + \frac{E\epsilon^2}{4m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2) \\
&\leq C(\omega) - \langle \nabla C(\omega), \epsilon\nabla C(\omega) - \frac{E\epsilon^2}{4m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2\rangle \\
&+ \frac{L}{2}\|\epsilon\nabla C(\omega) - \frac{E\epsilon^2}{4m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2\|_2^2 \\
&= C(\omega) - \epsilon(1 - \frac{L\epsilon}{2})\|\nabla C(\omega)\|_2^2 + \frac{E\epsilon^2}{4}(1 - L\epsilon)\langle \nabla C(\omega), \frac{1}{m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2\rangle \\
&+ \frac{LE^2\epsilon^4}{32}\|\frac{1}{m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2\|_2^2
\end{aligned} \tag{35}$$

Now an important factor for bounding $C(\omega^+)$ is whether the term $\langle \nabla C(\omega), \frac{1}{m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2\rangle$ is not smaller than 0. To know if it is true, we should know what the term means: it indicates whether $\frac{1}{m}\sum_{j=0}^{m-1}\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2$ increases or decreases if the parameter is one-step-updated following the cost function C. If $\frac{1}{m}\sum_{j=0}^{m-1}\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2$ decreases, the term $\langle \nabla C(\omega), \frac{1}{m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2\rangle$ is larger than 0.

Another thing to notice is that $\frac{1}{m}\sum_{j=0}^{m-1}\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2$ indicates the variance of client gradients. In the experiments by Johnson & Zhang (2013), the gradient variance has decreased during training in the overall perspective, which indicates that $\frac{1}{m}\sum_{j=0}^{m-1}\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2$ will also decrease if the parameter is one-step-updated following the cost function C. This empirical evidence implies that the assumption $\langle \nabla C(\omega), \frac{1}{m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2\rangle \geq 0$ is not implausible at least in intermediate stages of training. In this paper, we do not exactly show that $\frac{1}{m}\sum_{j=0}^{m-1}\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2$ will decrease during training but instead show that the approximation of the gradient variance decreases when the current parameter is far from the optimal point.

$$\begin{aligned}
\langle \nabla C(\omega), \frac{1}{m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2\rangle &= \langle \nabla C(\omega), \frac{1}{m}\sum_{j=0}^{m-1}\nabla\|\nabla C(\omega)\|_2^2 - \nabla\|\nabla C_j(\omega)\|_2^2\rangle \\
&= \frac{2}{m}\sum_{j=0}^{m-1}\langle \nabla C_j(\omega), \nabla\nabla C_j(\omega)\nabla C(\omega)\rangle \\
&- 2\langle \nabla C(\omega), \nabla\nabla C(\omega)\nabla C(\omega)\rangle
\end{aligned} \tag{36}$$

Here, as a rough explanation, we regard the current minimization problem as a maximum likelihood estimation problem and use the Fisher information matrix as an expectation of Hessian, or the current problem can be regarded as a sort of least square minimization problem and we can use a Gauss-

Newton approximation of Hessian. We use $\nabla\nabla C_j(\omega) \approx \nabla C_j(\omega)\nabla C_j(\omega)^T$ then we get

$$\langle \nabla C(\omega), \frac{1}{m}\sum_{j=0}^{m-1} \nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2 \rangle \approx \frac{2}{m}\sum_{j=0}^{m-1} \|\nabla C_j(\omega)\|_2^2 \langle \nabla C_j(\omega), \nabla C(\omega)\rangle - 2\|\nabla C(\omega)\|_2^4 \tag{37}$$

If the current parameter is far enough from the optimal point so that the angle between $\nabla C(\omega)$ and $\nabla C_j(\omega)$ is small, then we can approximate the equation above as

$$\langle \nabla C(\omega), \frac{1}{m}\sum_{j=0}^{m-1} \nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2 \rangle \approx \frac{2}{m}\sum_{j=0}^{m-1} \|\nabla C_j(\omega)\|_2^3 \|\nabla C(\omega)\|_2 - 2\|\nabla C(\omega)\|_2^4 \tag{38}$$

Reminding that $\frac{1}{m}\sum_{j=0}^{m-1} \nabla C_j(\omega) = \nabla C(\omega)$, by Jensen's inequality,

$$\langle \nabla C(\omega), \frac{1}{m}\sum_{j=0}^{m-1} \nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2 \rangle \approx \frac{2}{m}\sum_{j=0}^{m-1} \|\nabla C_j(\omega)\|_2^3 \|\nabla C(\omega)\|_2 - 2\|\nabla C(\omega)\|_2^4 \geq 0 \tag{39}$$

Now we assume that $\langle \nabla C(\omega), \frac{1}{m}\sum_{j=0}^{m-1} \nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2 \rangle \geq 0$, with assumption 2 and 3 we can bound $C(\omega^+)$ as

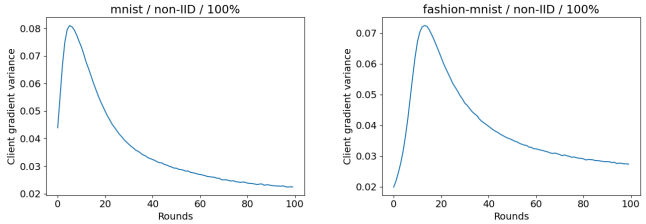
$$\begin{aligned} C(\omega^+) &\leq C(\omega) - \epsilon(1 - \frac{L\epsilon}{2})\|\nabla C(\omega)\|_2^2 + \frac{E\epsilon^2}{4}(1 - L\epsilon)\tilde{\sigma}\|\nabla C(\omega)\|_2^2 + \frac{E^2\epsilon^3}{32}\tilde{\sigma}^2\|\nabla C(\omega)\|_2^2 \\ &= C(\omega) - \epsilon(1 - \frac{L\epsilon}{2} - \frac{E\epsilon}{4}(1 - L\epsilon)\tilde{\sigma} - \frac{E^2\epsilon^2}{32}\tilde{\sigma}^2)\|\nabla C(\omega)\|_2^2 \end{aligned} \tag{40}$$

Compare this bound to the bound of gradient descent:

$$C(\omega^+) \leq C(\omega) - \epsilon(1 - \frac{L\epsilon}{2})\|\nabla C(\omega)\|_2^2 \tag{41}$$

It is possible to observe that the upper bound has become larger and the effective learning rate has decreased due to the presence of the drifting term, which hampers the convergence of FedAvg.

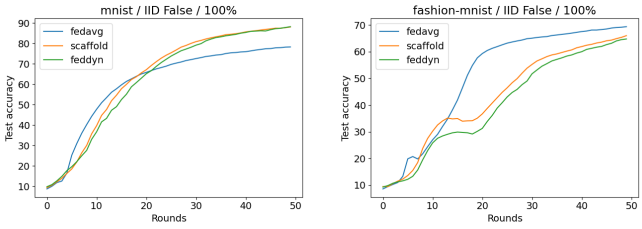
While the assumption of decreasing gradient variance might be viable in the intermediate stages of training, such an assumption can be incorrect in the very early stages of training where the norms of gradients tend to increase rapidly as depicted in Figure A.2. If the variance of gradients increase due to the increasing norm, $\langle \nabla C(\omega), \frac{1}{m}\sum_{j=0}^{m-1} \nabla\|\nabla C(\omega) - \nabla C_j(\omega)\|_2^2 \rangle$ will become negative and FedAvg might show a faster convergence than variance reduction methods. In fact, we were able to observe such a phenomenon in early stages of training during our experiments.



(a) (b)

Figure B.1: The variance of pseudo-gradients from clients. The variance rapidly increases in the initial stage and starts to decrease after a certain number of rounds. This also shows that the assumption of the decreasing gradient variance is viable in real training situations.

We checked the value of the gradient variance during training on MNIST and Fashion-MNIST to examine if there is a correlation between the accuracy of FedAvg and the fluctuation of the gradient variance. Not only were we able to verify that FedAvg can be faster in the early stages of training, but also observe that the time FedAvg starts to become slower coincides with the time the gradient variance starts to decrease. These results indicate that our analysis on the drifting term not only explains why FedAvg overall performs worse than gradient descent, but also explains why and when FedAvg can perform better than other optimization algorithms.



(a) (b)

Figure B.2: Test accuracies of various federated learning methods on (a) MNIST and (b) Fashion-MNIST. It is able to observe that FedAvg converges faster than SCAFFOLD and FedDyn in the very early stages of training. Another thing to observe is that the time FedAvg starts to become slower coincides with the time the gradient variance starts to decrease.

C IMPLICIT REGULARIZER OF CONVENTIONAL VARIANCE REDUCTION METHODS

In this section, we define some of the variables otherwise. Here, the cost function of j-th client is $C_j(\omega)$, while the client is performing a full-batch training with the learning rate ϵ. The average of all cost functions of clients is $C(\omega)$. The training is done for n rounds with m clients, and the number of iterations of each round is E. We assume that all clients participate in training for all rounds.

C.1 IMPLICIT REGULARIZER OF SCAFFOLD

During training of the j-th client at the i-th round, the parameter ω^{ij} is being updated based on the client control variate c_{ij} and the server control variate c_i. First, we need to approximate c_{ij} and c_i, which we will approximate as $\nabla C_j(\omega_0^{(i-1)j})$ and $\nabla C(\omega_0^{(i-1)j})$. Since the first round of SCAFFOLD starts with c_{ij} and c_i as zero which makes the first round equivalent to FedAvg, we consider that $\omega_0^{ij} = \omega_0^{(i-1)j} - E\epsilon\nabla C(\omega_0^{(i-1)j}) + O(E^2\epsilon^2)$ assuming the parameter updates are similar to the ones of FedAvg. It is possible to check that this assumption can be applied to all later rounds when the first round is equivalent to FedAvg. Then we can obtain $\nabla C_j(\omega_0^{(i-1)j})$ with

$$\begin{aligned}\nabla C_j(\omega_0^{(i-1)j}) &= \nabla C_j(\omega_0^{ij} + E\epsilon\nabla C(\omega_0^{(i-1)j}) + O(E^2\epsilon^2)) \\ &= \nabla C_j(\omega_0^{ij} + E\epsilon\nabla C(\omega_0^{ij} + E\epsilon\nabla C(\omega_0^{(i-1)j})) + O(E^2\epsilon^2)) \\ &= \nabla C_j(\omega_0^{ij} + E\epsilon\nabla C(\omega_0^{ij})) + O(E^2\epsilon^2) \end{aligned} \tag{42}$$

Since we will always multiply control variates with $E\epsilon$, we ignore high-order terms and approximate c_{ij} as $\nabla C_j(\omega_0^{ij} + E\epsilon\nabla C(\omega_0^{ij}))$. In the same way, we approximate c_i as $\nabla C(\omega_0^{ij} + E\epsilon\nabla C(\omega_0^{ij}))$. Now the discrete updates of the parameter ω^i during E steps can be expressed step-by-step.

$$\omega_1^{ij} = \omega_0^{ij} - \epsilon(\nabla C_j(\omega_0^{ij}) - c_{ij} + c_i) \tag{43}$$

$$\omega_2^{ij} = \omega_0^{ij} - \epsilon(\nabla C_j(\omega_0^{ij} - \epsilon(\nabla C_j(\omega_0^{ij}) - c_{ij} + c_i)) - c_{ij} + c_i) \tag{44}$$

$$\cdots$$

$$\omega_E^{ij} = \omega_0^{ij} - E\epsilon(\nabla C_j(\omega_0^{ij}) - c_{ij} + c_i) + \frac{E(E-1)}{2}\epsilon^2\nabla\nabla C_j(\omega_0^{ij})(\nabla C_j(\omega_0^{ij}) - c_{ij} + c_i) + O(E^3\epsilon^3) \tag{45}$$

Neglecting $O(E^3\epsilon^3)$ terms, parameter ω_{ij} is expressed as

$$\begin{aligned}\omega_E^{ij} = \omega_0^{ij} &- E\epsilon\nabla C(\omega_0^{ij}) - E^2\epsilon^2\nabla C(\omega_0^{ij})\nabla\nabla(C(\omega_0^{ij}) - C_j(\omega_0^{ij})) \\ &+ \frac{E(E-1)}{2}\epsilon^2\sum_{i=0}^{m-1}\nabla\nabla C_j(\omega_0^{ij})\nabla C(\omega_0^{ij}) + O(E^3\epsilon^3)\end{aligned} \tag{46}$$

After the i-th round, the client parameters are aggregated and form a parameter ω^i.

$$\omega_E^i = \omega_0^i - E\epsilon\nabla C(\omega_0^i) + \frac{E(E-1)}{4}\epsilon^2\nabla\|\nabla C(\omega_0^i)\|^2 + O(E^3\epsilon^3) \tag{47}$$

After n rounds of training, the expectation value of global parameter ω will become as

$$\begin{aligned}\mathbb{E}(\omega_{nE}) &= \omega_0 - nE\epsilon\nabla C(\omega_0) + \frac{n^2E^2\epsilon^2}{4}\nabla(\|\nabla C(\omega_0)\|^2 - \frac{1}{n}\|\nabla C(\omega_0)\|^2 \\ &+ \frac{1}{n}\|\nabla C(\omega_0)\|^2 - \frac{1}{nE}\|\nabla C(\omega_0)\|^2) + O(n^3E^3\epsilon^3) \qquad (48)\\ &= \omega_0 - nE\epsilon\nabla C(\omega_0) + \frac{n^2E^2\epsilon^2}{4}\nabla(\|\nabla C(\omega_0)\|^2 - \frac{1}{nE}\|\nabla C(\omega_0)\|^2) + O(n^3E^3\epsilon^3) \qquad (49)\end{aligned}$$

Then the modified loss under SCAFFOLD is

$$\tilde{C}_{SCAFFOLD}(\omega) = C(\omega) + \frac{\epsilon}{4}\|\nabla C(\omega)\|^2 \tag{50}$$

C.2 Implicit regularizer of FedDyn

In FedDyn, for the j-th client at the i-th round, firstly the local parameter ω^{ij} is updated along the loss function $\mathfrak{R}_j(\omega^{ij}) = C_j(\omega^{ij}) - \langle\nabla C_j(\omega_E^{(i-1)j}), \omega^{ij}\rangle + \frac{\alpha}{2}\|\omega^{ij} - \omega_0^{ij}\|^2$ while $\nabla C_j(\omega_E^{(i-1)j})$ acts as a client control variate and $\omega_{(i-1)E}$ is a global parameter from a previous round. The discrete updates of the parameter ω^{ij} during E steps can be expressed step-by-step.

$$\begin{aligned}\omega_1^{ij} &= \omega_0^{ij} - \epsilon\nabla C_j(\omega_0^{ij}) + \epsilon\nabla C_j(\omega_E^{(i-1)j}) - \epsilon\alpha(\omega_0^{ij} - \omega_0^{ij}) \qquad (51)\\ \omega_2^{ij} &= \omega_1^{ij} - \epsilon\nabla C_j(\omega_0^{ij}) + \epsilon\nabla C_j(\omega_E^{(i-1)j}) - \epsilon\alpha(\omega_1^{ij} - \omega_0^{ij})\\ &= (1-\epsilon\alpha)\omega_1^{ij} + \epsilon\alpha\omega_0^{ij} - \epsilon\nabla C_j(\omega_0^{ij}) + \epsilon\nabla C_j(\omega_E^{(i-1)j}) \qquad (52)\\ &\cdots\\ \omega_E^{ij} &= (1-\epsilon\alpha)\omega_{E-1}^{ij} + \epsilon\alpha\omega_0^{ij} + \epsilon\nabla C_j(\omega_E^{(i-1)j}) - \epsilon\nabla C_j(\omega_{E-1}^{ij}) \qquad (53)\end{aligned}$$

Then, after we organize the equations above, the updated parameter ω_E^{ij} will become

$$\begin{aligned}\omega_E^{ij} &= (1-\epsilon\alpha)^E\omega_0^{ij} + \epsilon\alpha\omega_0^{ij}(1 + \cdots + (1-\epsilon\alpha)^{E-1}) + \sum_{k=0}^{E-1}(1-\epsilon\alpha)^{E-1-k}\epsilon(\nabla C_j(\omega_E^{(i-1)j}) - \nabla C_j(\omega_k^{ij}))\\ &= \omega_0^{ij} + \frac{1-(1-\epsilon\alpha)^E}{\epsilon\alpha}\epsilon\nabla C_j(\omega_E^{(i-1)j}) - \sum_{k=0}^{E-1}(1-\epsilon\alpha)^{E-1-k}\epsilon\nabla C_j(\omega_k^{ij}) \qquad (54)\end{aligned}$$

If we ignore high-order terms, it is possible to express ω_E^{ij} as

$$\begin{aligned}\omega_E^{ij} &= \omega_0^{ij} + (E\epsilon - \frac{E(E-1)}{2}\alpha\epsilon^2)\nabla C_j(\omega_E^{(i-1)j}) - \sum_{k=0}^{E-1}(\epsilon - (E-1-k)\alpha\epsilon^2)\nabla C_j(\omega_k^{ij}) + O(E^3\epsilon^3)\\ &= \omega_0^{ij} + (E\epsilon - \frac{E(E-1)}{2}\alpha\epsilon^2)\nabla C_j(\omega_E^{(i-1)j}) - (\epsilon - (E-0)\alpha\epsilon^2)\nabla C_j(\omega_0^{ij})\\ &- (\epsilon - (E-1)\alpha\epsilon^2)\nabla C_j(\omega_0^{ij} - \epsilon\nabla C_j(\omega_0^{ij}) + \epsilon\nabla C_j(\omega_E^{(i-1)j}) + O(E^2\epsilon^2)) - \ldots\\ &= \omega_0^{ij} - E\epsilon(\nabla C_j(\omega_0^{ij}) - \nabla C_j(\omega_E^{(i-1)j})) + \frac{E(E-1)}{2}\epsilon^2(\alpha(\nabla C(\omega_0^{ij}) - \nabla C_j(\omega_E^{(i-1)j}))\\ &+ \nabla\nabla C(\omega_0^{ij})\nabla C(\omega_0^{ij}) - \nabla\nabla C(\omega_0^{ij})\nabla C_j(\omega_E^{(i-1)j})) + O(E^3\epsilon^3) \qquad (55)\end{aligned}$$

After the client updates, the server-side control variate is added to the aggregated parameter. Here, if the parameter ω^{ij} is sufficiently minimized throughout the round, the server-side control variate h^i becomes the average of local gradients, $\frac{1}{m}\sum_{j=0}^{m-1}\nabla C(\omega_0^{ij})$ (Acar et al., 2021). During the global parameter updates, we set $\alpha = \frac{1}{E\epsilon}$ then the discrete updates of the global parameter ω can be

expressed step-by-step as

$$\omega_E \approx \omega_0 - \frac{E\epsilon}{m}\sum_{j=1}^{m-1}(\nabla C_j(\omega_0) - \nabla C_j(\omega_E^{-1j}) + \nabla C_j(\omega_E^{0j})) + \frac{E(E-1)}{2}\epsilon^2\sum_{j=0}^{m-1}(\alpha(\nabla C(\omega_0) - \nabla C_j(\omega_E^{0j})) + \nabla\nabla C(\omega_0)\nabla C(\omega_0) - \nabla\nabla C(\omega_0)\nabla C_j(\omega_E^{0j})) + O(E^3\epsilon^3) \tag{56}$$

$$\begin{aligned}\omega_{2E} &\approx \omega_E - \frac{E\epsilon}{m}\sum_{j=1}^{m-1}(\nabla C_j(\omega_E) - \nabla C_j(\omega_E^{0j}) + \nabla C_j(\omega_E^{1j})) + \frac{E(E-1)}{2}\epsilon^2\sum_{j=0}^{m-1}(\alpha(\nabla C(\omega_E) - \nabla C_j(\omega_E^{1j})) \\ &+ \nabla\nabla C(\omega_E)\nabla C(\omega_E) - \nabla\nabla C(\omega_E)\nabla C_j(\omega_E^{1j})) + O(E^3\epsilon^3) \\ &= \omega_0 - \frac{E\epsilon}{m}\sum_{j=1}^{m-1}(\nabla C_j(\omega_0) - \nabla C_j(\omega_E^{-1j}) + \cancel{\nabla C_j(\omega_E^{0j})} \\ &+ \nabla C_j(\omega_E) - \cancel{\nabla C_j(\omega_E^{0j})} + \nabla C_j(\omega_E^{1j})) + \dots\end{aligned} \tag{57}$$

...

Ignoring high-order terms, then the expected value of ω_{nE} can be approximated as

$$\omega_{nE} \approx \omega_0 - nE\epsilon\nabla C(\omega_0) - E\epsilon\frac{1}{m}\sum_{j=0}^{m-1}\nabla C_j(\omega_E^{(n-1)j}) + \frac{n^2E^2\epsilon^2}{4}\nabla(\|\nabla C(\omega_0)\|^2 - \frac{1}{n}\|\nabla C(\omega_0)\|^2) + O(n^3E^3\epsilon^3) \tag{58}$$

If the parameter was updated enough, the modified loss can be approximated as

$$\tilde{C}_{FEDDYN}(\omega) \approx C(\omega) + \frac{E\epsilon}{4}\|\nabla C(\omega)\|^2 \tag{59}$$

If the value of α is smaller, FedDyn shows effect as if the learning rate has become larger to $1/\alpha$. With different α, if $1/\alpha$ is small enough, the aggregated parameter ω_{nE} can be approximated as

$$\omega_{nE} \approx \omega_0 - \frac{n}{\alpha}\nabla C(\omega_0) + \frac{n^2}{4\alpha^2}\nabla(\|\nabla C(\omega_0)\|^2 - \frac{1}{n}\|\nabla C(\omega_0)\|^2) + O(\frac{n^3}{\alpha^3}) \tag{60}$$

and the modified loss can be approximated as

$$\tilde{C}_{FEDDYN}(\omega) \approx C(\omega) + \frac{1}{4\alpha}\|\nabla C(\omega)\|^2 \tag{61}$$

However, there is one limitation to this analysis: as in Acar et al. (2021) we assumed that h^i becomes $\frac{1}{m}\sum_{j=0}^{m-1}\nabla C(\omega_0^{ij})$ when a round ends. Such an assumption is satisfied when the local loss function $\mathfrak{R}_j(\omega^{ij}) = C_j(\omega^{ij}) - \langle\nabla C_j(\omega_E^{(i-1)j}), \omega^{ij}\rangle + \frac{\alpha}{2}\|\omega^{ij} - \omega_0^{ij}\|^2$ is sufficiently minimized for a long period. The point of collision is another assumption of our analysis: a small magnitude of $E\epsilon$. A small magnitude of $E\epsilon$ hampers a sufficient minimization of the local loss function $\mathfrak{R}_j(\omega^{ij})$. Though we assumed that h^i becomes $\frac{1}{m}\sum_{j=0}^{m-1}\nabla C(\omega_0^{ij})$ following the assumption of Acar et al. (2021), it makes Equation 61 remain as an approximation of the modified loss of FedDyn.