On the effectiveness of partial variance reduction in federated learning with heterogeneous data

Bo Li*   Mikkel N. Schmidt   Tommy S. Alstrøm   Sebastian U. Stich
Technical University of Denmark   CISPA‡
{blia, mnsc, tsal}@dtu.dk   stich@cispa.de

Abstract

Data heterogeneity across clients is a key challenge in federated learning. Prior works address this by either aligning client and server models or using control variates to correct client model drift. Although these methods achieve fast convergence in convex or simple non-convex problems, the performance in over-parameterized models such as deep neural networks is lacking. In this paper, we first revisit the widely used FedAvg algorithm in a deep neural network to understand how data heterogeneity influences the gradient updates across the neural network layers. We observe that while the feature extraction layers are learned efficiently by FedAvg, the substantial diversity of the final classification layers across clients impedes the performance. Motivated by this, we propose to correct model drift by variance reduction only on the final layers. We demonstrate that this significantly outperforms existing benchmarks at a similar or lower communication cost. We furthermore provide proof for the convergence rate of our algorithm.

1. Introduction

Federated learning (FL) is emerging as an essential distributed learning paradigm in large-scale machine learning. Unlike in traditional machine learning, where a model is trained on the collected centralized data, in federated learning, each client (*e.g.* phones and institutions) learns a model with its local data. A centralized model is then obtained by aggregating the updates from all participating clients without ever requesting the client data, thereby ensuring a certain level of user privacy [13, 17]. Such an algorithm is especially beneficial for tasks where the data is sensitive, *e.g.* chemical hazards detection and diseases diagnosis [33].

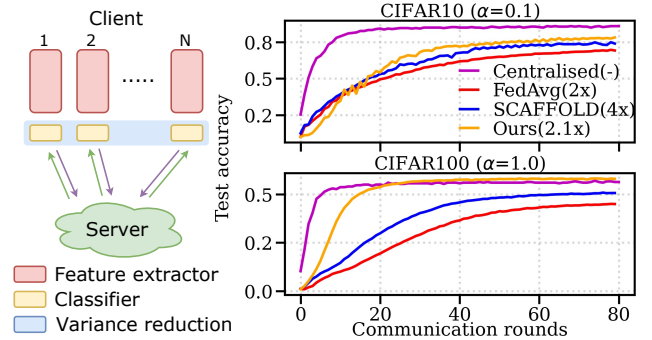Two primary challenges in federated learning are i) han-

Figure 1. Our proposed FedPVR framework with the performance (communicated parameters per round client⟺server). Smaller α corresponds to higher data heterogeneity. Our method achieves a better speedup than existing approaches by transmitting a slightly larger number of parameters than FedAvg.

dling data heterogeneity across clients [13] and ii) limiting the cost of communication between the server and clients [10]. In this setting, FedAvg [17] is one of the most widely used schemes: A server broadcasts its model to clients, which then update the model using their local data in a series of steps before sending their individual model to the server, where the models are aggregated by averaging the parameters. The process is repeated for multiple communication rounds. While it has shown great success in many applications, it tends to achieve subpar accuracy and convergence when the data are heterogeneous [14, 24, 31].

The slow and sometimes unstable convergence of FedAvg can be caused by client drift [14] brought on by data heterogeneity. Numerous efforts have been made to improve FedAvg's performance in this setting. Prior works attempt to mitigate client drift by penalizing the distance between a client model and the server model [20, 31] or by performing variance reduction techniques while updating client models [1, 14, 32]. These works demonstrate fast convergence on convex problems or for simple neural networks; however, their performance on deep neural

---

networks, which are state-of-the-art for many centralized learning tasks [11, 34], has yet to be well explored. Adapting techniques that perform well on convex problems to neural networks is non-trivial [7] due to their "intriguing properties" [38] such as over-parametrization and permutation symmetries.

To overcome the above issues, we revisit the FedAvg algorithm with a deep neural network (VGG-11 [34]) under the assumption of data heterogeneity and full client participation. Specifically, we investigate which layers in a neural network are mostly influenced by data heterogeneity. We define *drift diversity*, which measures the diversity of the directions and scales of the averaged gradients across clients per communication round. We observe that in the non-IID scenario, the deeper layers, especially the final classification layer, have the highest diversity across clients compared to an IID setting. This indicates that FedAvg learns good feature representations even in the non-IID scenario [5] and that the significant variation of the deeper layers across clients is a primary cause of FedAvg's subpar performance.

Based on the above observations, we propose to align the classification layers across clients using variance reduction. Specifically, we estimate the average updating direction of the classifiers (the last several fully connected layers) at the client c_i and server level c and use their difference as a control variate [14] to reduce the variance of the classifiers across clients. We analyze our proposed algorithm and derive a convergence rate bound.

We perform experiments on the popular federated learning benchmark datasets CIFAR10 [19] and CIFAR100 [19] using two types of neural networks, VGG-11 [34] and ResNet-8 [11], and different levels of data heterogeneity across clients. We experimentally show that we require fewer communication rounds compared to the existing methods [14, 17, 31] to achieve the same accuracy while transmitting a similar or slightly larger number of parameters between server and clients than FedAvg (see Fig. 1). With a (large) fixed number of communication rounds, our method achieves on-par or better top-1 accuracy, and in some settings it even outperforms centralized learning. Using conformal prediction [3], we show how performance can be improved further using adaptive prediction sets.

We show that applying variance reduction on the last layers increases the diversity of the feature extraction layers. This diversity in the feature extraction layers may give each client more freedom to learn richer feature representations, and the uniformity in the classifier then ensures a less biased decision. We summarize our contributions here:

- We present our algorithm for partial variance-reduced federated learning (FedPVR). We experimentally demonstrate that the key to the success of our algorithm is the diversity between the feature extraction layers and the alignment between the classifiers.

- We prove the convergence rate in the convex settings and non-convex settings, precisely characterize its weak dependence on data-heterogeneity measures and show that FedPVR provably converges as fast as the centralized SGD baseline in most practical relevant cases.

- We experimentally show that our algorithm is more communication efficient than previous works across various levels of data heterogeneity, datasets, and neural network architectures. In some cases where data heterogeneity exists, the proposed algorithm even performs slightly better than centralized learning.

2. Related work

2.1. Federated learning

Federated learning (FL) is a fast-growing field [13, 40]. We mainly describe FL methods in non-IID settings where the data is distributed heterogeneously across clients. Among the existing approaches, FedAvg [25] is the *de facto* optimization technique. Despite its solid empirical performances in IID settings [13, 25], it tends to achieve a subpar accuracy-communication trade-off in non-IID scenarios.

Many works attempt to tackle FL when data is heterogeneous across clients [1, 9, 14, 31, 39, 42]. FedProx [31] proposes a temperature parameter and proximal regularization term to control the divergence between client and server models. However, the proximal term does not bring the alignment between the global and local optimal points [1]. Similarly, some works control the update direction by introducing client-dependent control variate [1, 14, 17, 27, 32] that is also communicated between the server and clients. They have achieved a much faster convergence rate, but their performance in a non-convex setup, especially in deep neural networks, such as ResNet [11] and VGG [34], is not well explored. Besides, they suffer from a higher communication cost due to the transmission of the extra control variates, which may be a critical issue for resources-limited IoT mobile devices [10]. Among these methods, SCAFFOLD [14] is the most closely related method to ours, and we give a more detailed comparison in section 3 and 5.

Another line of work develops FL algorithms based on the characteristics, such as expressive feature representations [28] of neural networks. Collins et al. [5] show that FedAvg is powerful in learning common data representations from clients' data. FedBabu [29], TCT [41], and CCVR [24] propose to improve FL performance by fine-tuning the classifiers with a standalone dataset or features that are simulated based on the client models. However, preparing a standalone dataset/features that represents the data distribution across clients is challenging as this usually requires domain knowledge and may raise privacy concerns.
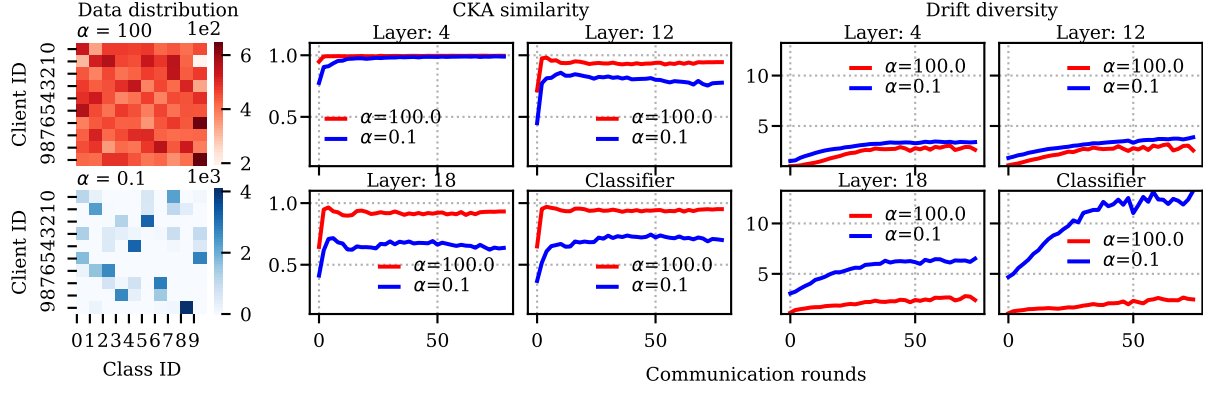
Figure 2. Data distribution (number of images per client per class) with different levels of heterogeneity, client CKA similarity, and the drift diversity of each layer in VGG-11 (20 layers) with FedAvg. **Deep layers in an over-parameterised neural network have higher disagreement and variance when the clients are heterogeneous using FedAvg.**

Moon [20] encourages the similarity of the representations across different client models by using contrastive loss [4] but with the cost of three full-size models in memory on each client, which may limit its applicability in resource-limited devices.

Other works focus on reducing the communication cost by compressing the transmitted gradients [2, 8, 21, 26, 36]. They can reduce the communication bandwidth by adjusting the number of bits sent per iteration. These works are complementary to ours and can be easily integrated into our method to save communication costs.

2.2. Variance reduction

Stochastic variance reduction (SVR), such as SVRG [12], SAGA [6], and their variants, use control variate to reduce the variance of traditional stochastic gradient descent (SGD). These methods can remarkably achieve a linear convergence rate for strongly convex optimization problems compared to the sub-linear rate of SGD. Many federated learning algorithms, such as SCAFFOLD [14] and DANE [32], have adapted the idea of variance reduction for the whole model and achieved good convergence on convex problems. However, as [7] demonstrated, naively applying variance reduction techniques gives no actual variance reduction and tends to result in a slower convergence in deep neural networks. This suggests that adapting SVR techniques in deep neural networks for FL requires a more careful design.

2.3. Conformal prediction

Conformal prediction is a general framework that computes a prediction set guaranteed to include the true class with a high user-determined probability [3, 30]. It requires no retraining of the models and achieves a finite-sum coverage guarantee [3]. As FL algorithms can hardly perform as well as centralized learning [24] when the data heterogene-

Table 1. Notations used in this paper

R, r	Number of communication rounds and round index
K, k	Number of local steps, local step index
N, i	Number of clients, client index
$\boldsymbol{y}_{i,k}^r$	client model i at step k and round r
$\boldsymbol{x}^r$	server model at round r
$\boldsymbol{c}_i^r, \boldsymbol{c}^r$	client and server control variate

ity is high, we can integrate conformal prediction in FL to improve the empirical coverage by slightly increasing the predictive set size. This can be beneficial in sensitive use cases such as detecting chemical hazards, where it is better to give a prediction set that contains the correct class than producing a single but wrong prediction.

3. Method

3.1. Problem statement

Given N clients with full participation, we formalise the problem as minimizing the average of the stochastic functions with access to stochastic samples in Eq. 1 where $\boldsymbol{x}$ is the model parameters and f_i represents the loss function at client i with dataset $\mathcal{D}_i$,

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \left(f(\boldsymbol{x}) := \frac{1}{N} \sum_{i=1}^N f_i(\boldsymbol{x}) \right), \quad (1)$$

where $f_i(\boldsymbol{x}) := \mathbb{E}_{\mathcal{D}_i}[f_i(\boldsymbol{x}; \mathcal{D}_i)]$.

3.2. Motivation

When the data $\{\mathcal{D}_i\}$ are heterogeneous across clients, FedAvg suffers from *client drift* [14], where the average of the local optimal $\bar{\boldsymbol{x}}^* = \frac{1}{N} \sum_{i \in N} \boldsymbol{x}_i^*$ is far from the global optimal $\boldsymbol{x}^*$. To understand what causes client drift, specifically which layers in a neural network are influenced most by the data heterogeneity, we perform a simple experiment using FedAvg and CIFAR10 datasets on a VGG-11. The detailed experimental setup can be found in section 4.

In an over-parameterized model, it is difficult to directly calculate client drift $||\bar{\boldsymbol{x}}^* - \boldsymbol{x}^*||^2$ as it is challenging to obtain the global optimum $\boldsymbol{x}^*$. We instead hypothesize that we can represent the influence of data heterogeneity on the model by measuring 1) drift diversity and 2) client model similarity. Drift diversity reflects the diversity in the amount each client model deviates from the server model after an update round.

Definition 1 (Drift diversity). *We define the drift diversity across N clients at round r as:*

$$\xi^r := \frac{\sum_{i=1}^{N} ||\boldsymbol{m}_i^r||^2}{|| \sum_{i=1}^{N} \boldsymbol{m}_i^r||^2} \quad \boldsymbol{m}_i^r = \boldsymbol{y}_{i,K}^r - \boldsymbol{x}^{r-1} \quad (2)$$

Drift diversity ξ is high when all the clients update their models in different directions, i.e., when dot products between client updates $\boldsymbol{m}_i$ are small. When each client performs K steps of vanilla SGD updates, ξ depends on the directions and amplitude of the gradients over N clients and is equivalent to $\frac{\sum_{i=1}^{N} || \sum_k g_i(\boldsymbol{y}_{i,k})||^2}{|| \sum_{i=1}^{N} \sum_k g_i(\boldsymbol{y}_{i,k})||^2}$, where $g_i(\boldsymbol{y}_{i,k})$ is the stochastic mini-batch gradient.

After updating client models, we quantify the client model similarity using centred kernel alignment (CKA) [18] computed on a test dataset. CKA is a widely used permutation invariant metric for measuring the similarity between feature representations in neural networks [18, 24, 28].

Fig. 2 shows the movement of ξ and CKA across different levels of data heterogeneity using FedAvg. We observe that the similarity and diversity of the early layers (*e.g.* layer index 4 and 12) are with a higher agreement between the IID ($\alpha = 100.0$) and non-IID ($\alpha = 0.1$) experiments, which indicates that FedAvg can still learn and extract good feature representations even when it is trained with non-IID data. The lower similarity on the deeper layers, especially the classifiers, suggests that these layers are strongly biased towards their local data distribution. When we only look at the model that is trained with $\alpha = 0.1$, we see the highest diversity and variance on the classifiers across clients compared to the rest of the layers. Based on the above observations, we propose to align the classifiers across clients using variance reduction. We deploy client and server control variates to control the updating directions of the classifiers.

3.3. Classifier variance reduction

Our proposed algorithm (Alg. I) consists of three parts: i) client updating (Eq. 5-6) ii) client control variate updating, (Eq. 7), and iii) server updating (Eq. 8-9)

We first define a vector $\boldsymbol{p} \in \mathbb{R}^d$ that contains 0 or 1 with v non-zero elements ($v \ll d$) in Eq. 3. We recover SCAFFOLD with $\boldsymbol{p} = 1$ and recover FedAvg with $\boldsymbol{p} = 0$. For the set of indices j where $\boldsymbol{p}_j = 1$ (S_{svr} from Eq. 4), we update the corresponding weights $\boldsymbol{y}_{i,S_{\text{svr}}}$ with variance reduction such that we maintain a state for each client ($\boldsymbol{c}_i \in \mathbb{R}^v$) and

for the server ($\boldsymbol{c} \in \mathbb{R}^v$) in Eq. 5. For the rest of the indices S_{sgd} from Eq. 4, we update the corresponding weights $\boldsymbol{y}_{iS_{\text{sgd}}}$ with SGD in Eq. 6. As the server variate $\boldsymbol{c}$ is an average of $\boldsymbol{c}_i$ across clients, we can safely initialise them as $\boldsymbol{0}$.

In each communication round, each client receives a copy of the server model $\boldsymbol{x}$ and the server control variate $\boldsymbol{c}$. They then perform K model updating steps (see Eq. 5- 6 for one step) using cross-entropy as the loss function. Once this is finished, we calculate the updated client control variate $\boldsymbol{c}_i$ using Eq. 7. The server then receives the updated $\boldsymbol{c}_i$ and $\boldsymbol{y}_i$ from all the clients for aggregation (Eq. 8-9). This completes one communication round.

$$\boldsymbol{p} := \{0,1\}^d, \quad v = \sum \boldsymbol{p} \quad (3)$$

$$S_{\text{svr}} := \{j : \boldsymbol{p}_j = 1\}, \quad S_{\text{sgd}} := \{j : \boldsymbol{p}_j = 0\} \quad (4)$$

$$\boldsymbol{y}_{i,S_{\text{svr}}} \leftarrow \boldsymbol{y}_{i,S_{\text{svr}}} - \eta_l(g_i(\boldsymbol{y}_i)_{S_{\text{svr}}} - \boldsymbol{c}_i + \boldsymbol{c}) \quad (5)$$

$$\boldsymbol{y}_{i,S_{\text{sgd}}} \leftarrow \boldsymbol{y}_{i,S_{\text{sgd}}} - \eta_l g_i(\boldsymbol{y}_i)_{S_{\text{sgd}}} \quad (6)$$

$$\boldsymbol{c}_i \leftarrow \boldsymbol{c}_i - \boldsymbol{c} + \frac{1}{K\eta_l} \left(\boldsymbol{x}_{S_{\text{svr}}} - \boldsymbol{y}_{i,S_{\text{svr}}} \right) \quad (7)$$

$$\boldsymbol{x} \leftarrow (1 - \eta_g)\boldsymbol{x} + \frac{1}{N} \sum_{i \in N} \boldsymbol{y}_i \quad (8)$$

$$\boldsymbol{c} \leftarrow \frac{1}{N} \sum_{i \in N} \boldsymbol{c}_i \quad (9)$$

Algorithm I Partial variance reduction (FedPVR)

server: initialise the server model $\boldsymbol{x}$, the control variate $\boldsymbol{c}$, and global step size η_g

client: initialise control variate $\boldsymbol{c}_i$ and local step size η_l

mask: $\boldsymbol{p} := \{0,1\}^d$, $S_{\text{sgd}} := \{j : \boldsymbol{p}_j = 0\}$, $S_{\text{svr}} := \{j : \boldsymbol{p}_j = 1\}$

1: **procedure** MODEL UPDATING
2:     **for** $r = 1 \rightarrow R$ **do**
3:         **communicate** $\boldsymbol{x}$ and $\boldsymbol{c}$ to all clients $i \in [N]$
4:         **for** On client $i \in [N]$ in parallel **do**
5:             $\boldsymbol{y}_i \leftarrow \boldsymbol{x}$
6:             **for** $k = 1 \rightarrow K$ **do**
7:                 compute minibatch gradient $g_i(\boldsymbol{y}_i)$
8:                 $\boldsymbol{y}_{i,S_{\text{sgd}}} \leftarrow \boldsymbol{y}_{i,S_{\text{sgd}}} - \eta_l g_i(\boldsymbol{y}_i)_{S_{\text{sgd}}}$
9:                 $\boldsymbol{y}_{i,S_{\text{svr}}} \leftarrow \boldsymbol{y}_{i,S_{\text{svr}}} - \eta_l(g_i(\boldsymbol{y}_i)_{S_{\text{svr}}} - \boldsymbol{c}_i + \boldsymbol{c})$
10:             **end for**
11:             $\boldsymbol{c}_i \leftarrow \boldsymbol{c}_i - \boldsymbol{c} + \frac{1}{K\eta_l}(\boldsymbol{x}_{S_{\text{svr}}} - \boldsymbol{y}_{i,S_{\text{svr}}})$
12:             **communicate** $\boldsymbol{y}_i, \boldsymbol{c}_i$
13:         **end for**
14:         $\boldsymbol{x} \leftarrow (1 - \eta_g)\boldsymbol{x} + \frac{1}{N} \sum_{i \in N} \boldsymbol{y}_i$
15:         $\boldsymbol{c} \leftarrow \frac{1}{N} \sum_{i \in N} \boldsymbol{c}_i$
16:     **end for**
17: **end procedure**

In terms of implementation, we can simply assume the control variate for the block of weights that is updated with SGD as $\boldsymbol{0}$ and implement line 8 and 9 in one step

Ours vs SCAFFOLD [14] While our work is similar to SCAFFOLD in the use of variance reduction, there are some fundamental differences. We both communicate control variates between the clients and server, but our control

variate ($2v \leq 0.1d$) is significantly smaller than the one in SCAFFOLD ($2d$). This 2x decrease in bits can be critical for some low-power IoT devices as the communication may consume more energy [10]. From the application point of view, SCAFFOLD achieved great success in convex or simple two layers problems. However, adapting the techniques that work well from convex problems to over-parameterized models is non-trivial [38], and naively adapting variance reduction techniques on deep neural networks gives little or no convergence speedup [7]. Therefore, the significant improvement achieved by our method gives essential and non-trivial insight into what matters when tackling data heterogeneity in FL in over-parameterized models.

3.4. Convergence rate

We state the convergence rate in this section. We assume functions $\{f_i\}$ are β-smooth following [16, 35]. We then assume $g_i(\boldsymbol{x}) := \nabla f_i(x; \mathcal{D}_i)$ is an unbiased stochastic gradient of f_i with variance bounded by σ^2. We assume strongly convexity ($\mu > 0$) and general convexity ($\mu = 0$) for some of the results following [14]. Furthermore, we also make assumptions about the heterogeneity of the functions.

For convex functions, we assume the heterogeneity of the function $\{f_i\}$ at the optimal point $\boldsymbol{x}^*$ (such a point always exists for a strongly convex function) following [15, 16].

Assumption 1 (ζ-heterogeneity). *We define a measure of variance at the optimum $\boldsymbol{x}^*$ given N clients as :*

$$\zeta^2 := \frac{1}{N} \sum_{i=1}^N \mathbb{E} ||\nabla f_i(\boldsymbol{x}^*)||^2. \qquad (10)$$

For the non-convex functions, such an unique optimal point $\boldsymbol{x}^*$ does not necessarily exist, so we generalize Assumption 1 to Assumption 2.

Assumption 2 ($\hat{\zeta}$-heterogeneity). *We assume there exists constant $\hat{\zeta}$ such that $\forall \boldsymbol{x} \in \mathbb{R}^d$*

$$\frac{1}{N} \sum_{i=1}^N \mathbb{E} ||\nabla f_i(\boldsymbol{x})||^2 \leq \hat{\zeta}^2. \qquad (11)$$

Given the mask $\boldsymbol{p}$ as defined in Eq. 3, we know $||\boldsymbol{p} \odot \boldsymbol{x}|| \leq ||\boldsymbol{x}||$. Therefore, we have the following propositions.

Proposition 1 (Implication of Assumption 1). *Given the mask $\boldsymbol{p}$, we define the heterogeneity of the block of weights that are not variance reduced at the optimum $\boldsymbol{x}^*$ as:*

$$\zeta_{1-p}^2 := \frac{1}{N} \sum_{i=1}^N ||(\boldsymbol{1} - \boldsymbol{p}) \odot \nabla f_i(\boldsymbol{x}^*)||^2, \qquad (12)$$

If Assumption 1 holds, then it also holds that:

$$\zeta_{1-p}^2 \leq \zeta^2. \qquad (13)$$

In Proposition 1, $\zeta_{1-p}^2 = \zeta^2$ if $\boldsymbol{p} = \boldsymbol{0}$ and $\zeta_{1-p}^2 = 0$ if $\boldsymbol{p} = \boldsymbol{1}$. If $\boldsymbol{p} \neq \boldsymbol{0}$ and $\boldsymbol{p} \neq \boldsymbol{1}$, as the heterogeneity of the shallow weights is lower than the deeper weights [41], we have $\zeta_{1-p}^2 \leq \zeta^2$. Similarly, we can validate Proposition 2.

Proposition 2 (Implication of Assumption 2). *Given the mask $\boldsymbol{p}$, we assume there exists constant $\hat{\zeta}_{1-p}$ such that $\forall \boldsymbol{x} \in \mathbb{R}^d$, the heterogeneity of the block of weights that are not variance reduced:*

$$\frac{1}{N} \sum_{i=1}^N ||(\boldsymbol{1} - \boldsymbol{p}) \odot \nabla f_i(\boldsymbol{x})||^2 \leq \hat{\zeta}_{1-p}, \qquad (14)$$

If Assumption 2 holds, then it also holds that:

$$\hat{\zeta}_{1-p}^2 \leq \hat{\zeta}^2. \qquad (15)$$

Theorem 1. *For any β-smooth function $\{f_i\}$, the output of $\mathtt{FedPVR}$ has expected error smaller than ϵ for $\eta_g = \sqrt{N}$ and some values of η_l, R satisfying:*

- ***Strongly convex:*** $\eta_l \leq \min\left(\frac{1}{80K\eta_g\beta}, \frac{26}{20\mu K\eta_g}\right)$,

$$R = \tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu N K \epsilon} + \frac{\zeta_{1-p}^2}{\mu\epsilon} + \frac{\beta}{\mu}\right), \qquad (16)$$

- ***General convex:*** $\eta_l \leq \frac{1}{80K\eta_g\beta}$,

$$R = \mathcal{O}\left(\frac{\sigma^2 D}{K N \epsilon^2} + \frac{\zeta_{1-p}^2 D}{\epsilon^2} + \frac{\beta D}{\epsilon} + F\right), \qquad (17)$$

- ***Non-convex:*** $\eta_l \leq \frac{1}{26K\eta_g\beta}$, and $R \geq 1$, then:

$$R = \mathcal{O}\left(\frac{\beta\sigma^2 F}{K N \epsilon^2} + \frac{\beta\hat{\zeta}_{1-p}^2 F}{N\epsilon^2} + \frac{\beta F}{\epsilon}\right), \qquad (18)$$

Where $D := ||\boldsymbol{x}^0 - \boldsymbol{x}^*||^2$ and $F := f(\boldsymbol{x}^0) - f^*$.

Given the above assumptions, the convergence rate is given in Theorem I. When $\boldsymbol{p} = \boldsymbol{1}$, we recover SCAFFOLD convergence guarantee as $\zeta_{1-p}^2 = 0$, $\hat{\zeta}_{1-p}^2 = 0$. In the strongly convex case, the effect of the heterogeneity of the block of weights that are not variance reduced ζ_{1-p}^2 becomes negligible if $\tilde{\mathcal{O}}\left(\frac{\zeta_{1-p}^2}{\epsilon}\right)$ is sufficiently smaller than $\tilde{\mathcal{O}}\left(\frac{\sigma^2}{NK\epsilon}\right)$. In such case, our rate is $\frac{\sigma^2}{NK\epsilon} + \frac{1}{\mu}$, which recovers the SCAFFOLD in the strongly convex without sampling and further matches that of SGD (with mini-batch size K on each worker). We also recover the FedAvg rate* at simple IID case. See Appendix. B for the full proof.

---

*FedAvg at strongly convex case has the rate $R = \tilde{\mathcal{O}}(\frac{\sigma^2}{\mu K N \epsilon} + \frac{\sqrt{\beta}G}{\mu\sqrt{\epsilon}} + \frac{\beta}{\mu})$ with G measures the gradient dissimilarity. At simple IID case, G=0 [14].

4. Experimental setup

We demonstrate the effectiveness of our approach with CIFAR10 [19] and CIFAR100 [19] on image classification tasks. We simulate the data heterogeneity scenario following [22] by partitioning the data according to the Dirichlet distribution with the concentration parameter α. The smaller the α is, the more imbalanced the data are distributed across clients. An example of the data distribution over multiple clients using the CIFAR10 dataset can be seen in Fig. 2. In our experiment, we use $\alpha \in \{0.1, 0.5, 1.0\}$ as these are commonly used concentration parameters [22]. Each client has its local data, and this data is kept to be the same during all the communication rounds. We hold out the test dataset at the server for evaluating the classification performance of the server model. Following [22], we perform the same data augmentation for all the experiments.

We use two models: VGG-11 and ResNet-8 following [22]. We perform variance reduction for the last three layers in VGG-11 and the last layer in ResNet-8. We use 10 clients with full participation following [41] (close to cross-silo setup) and a batch size of 256. Each client performs 10 local epochs of model updating. We set the server learning rate $\eta_g = 1$ for all the models [14]. We tune the clients learning rate from $\{0.05, 0.1, 0.2, 0.3\}$ for each individual experiment. The learning rate schedule is experimentally chosen from constant, cosine decay [23], and multiple step decay [22]. We compare our method with the representative federated learning algorithms FedAvg [25], FedProx [31], SCAFFOLD [14], and FedDyn [1]. All the results are averaged over three repeated experiments with different random initialization. We leave 1% of the training data from each client out as the validation data to tune the hyperparameters (learning rate and schedule) per client. See Appendix. C for additional experimental setups. The code is at github.com/lyn1874/fedpvr.

5. Experimental results

We demonstrate the performance of our proposed approach in the FL setup with data heterogeneity in this section. We compare our method with the existing state-of-the-art algorithms on various datasets and deep neural networks. For the baseline approaches, we finetune the hyperparameters and only show the best performance we get. Our main findings are 1) we are more communication efficient than the baseline approaches, 2) conformal prediction is an effective tool to improve FL performance in high data heterogeneity scenarios, and 3) the benefit of the trade-off between diversity and uniformity for using deep neural networks in FL.

5.1. Communication efficiency and accuracy

We first report the number of rounds required to achieve a certain level of Top 1% accuracy (66% for CIFAR10 and 44% for CIFAR100) in Table. 2. An algorithm is more communication efficient if it requires less number of rounds to achieve the same accuracy and/or if it transmits fewer number of parameters between the clients and server. Compared to the baseline approaches, we require much fewer number of rounds for almost all types of data heterogeneity and models. We can achieve a speedup between 1.5 and 6.7 than FedAvg. We also observe that ResNet-8 tends to converge slower than VGG-11, which may be due to the aggregation of the Batch Normalization layers that are discrepant between the local data distribution [22].

We next compare the top-1 accuracy between centralized learning and federated learning algorithms. For the centralized learning experiment, we tune the learning rate from $\{0.01, 0.05, 0.1\}$ and report the best test accuracy based on the validation dataset. We train the model for 800 epochs which is as same as the total number of epochs in the federated learning algorithms (80 communication rounds x 10 local epochs). The results are shown in Table. 3. We also show the number of copies of the parameters that need to be transmitted between the server and clients (*e.g.* 2x means we communicate $\boldsymbol{x}$ and $\boldsymbol{y}_i$)

Table. 3 shows that our approach achieves a much better Top-1 accuracy compared to FedAvg with transmitting a similar or slightly bigger number of parameters between the server and client per round. Our method also achieves slightly better accuracy than centralized learning when the data is less heterogeneous (*e.g.* $\alpha = 0.5$ for CIFAR10 and $\alpha = 1.0$ for CIFAR100).

5.2. Conformal prediction

When the data heterogeneity is high across clients, it is difficult for a federated learning algorithm to match the centralized learning performance [24]. Therefore, we demonstrate the benefit of using simple post-processing conformal prediction to improve the model performance.

We examine the relationship between the empirical coverage and the average predictive set size for the server model after 80 communication rounds for each federated learning algorithm. The empirical coverage is the percentage of the data samples where the correct prediction is in the predictive set, and the average predictive size is the average of the length of the predictive sets over all the test images [3]. See Appendix. C for more information about conformal prediction setup and results.

The results for when $\alpha = 0.1$ for both datasets and architectures are shown in Fig. 3. We show that by slightly increasing the predictive set size, we can achieve a similar accuracy as the centralized performance. Besides, our approach tends to surpass the centralized top-1 performance similar to or faster than other approaches. In sensitive use cases such as chemical threat detection, conformal prediction is a valuable tool to achieve certified accuracy at the

Table 2. The required number of communication rounds (speedup compared to FedAvg) to achieve a certain level of top-1 accuracy (66% for the CIFAR10 dataset and 44% for the CIFAR100 dataset). Our method requires fewer rounds to achieve the same accuracy.

	CIFAR10 (66%)				CIFAR100 (44%)			
	α=0.1		α=0.5		α=0.1		α=1.0	
	VGG-11	ResNet-8	VGG-11	ResNet-8	VGG-11	ResNet-8	VGG-11	ResNet-8
	No. rounds	No. rounds	No. rounds	No. rounds	No. rounds	No. rounds	No. rounds	No. rounds
FedAvg	55(1.0x)	90(1.0x)	15(1.0x)	15(1.0x)	100 + (1.0x)	100 + (1.0x)	80(1.0x)	56(1.0x)
FedProx	52(1.1x)	75(1.2x)	16(0.9x)	20(0.8x)	100 + (1.0x)	100 + (1.0x)	80(1.0x)	59(0.9x)
SCAFFOLD	39(1.4x)	57(1.6x)	14(1.0x)	9(1.7x)	80(> 1.3x)	**61(> 1.6x)**	36(2.2x)	25(2.2x)
FedDyn	**27(2.0x)**	67(1.3x)	15(1.0x)	34(0.4x)	80 + (−)	80 + (−)	24(3.3x)	51(1.1x)
Ours	**27(2.0x)**	**50(1.8x)**	**9(1.6x)**	**5(3.0x)**	**37(> 2.7x)**	66(> 1.5x)	**12(6.7x)**	**15(3.7x)**

Table 3. The top-1 accuracy (%) after running 80 communication rounds using different methods on CIFAR10 and CIFAR100, together with the number of communicated parameters between the client and the server. We train the centralised model for 800 epochs (= 80 rounds x 10 local epochs in FL). Higher accuracy is better, and we highlight the best accuracy in red colour.

| | VGG-11 | | | | | ResNet-8 | | | | |
| | CIFAR10 | | CIFAR100 | | server⇔client | CIFAR10 | | CIFAR100 | | server⇔client |
	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 1.0$		$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 1.0$	
Centralised	**87.5**		56.3		-	83.4		**56.8**		-
FedAvg	69.3	80.9	34.3	45.0	2x	64.9	79.1	38.8	47.0	2x
Fedprox	72.1	80.4	35.0	43.2	2x	66.1	77.9	42.0	47.2	2x
SCAFFOLD	74.1	83.5	43.4	50.6	4x	66.6	80.3	43.8	52.3	4x
FedDyn	77.4	80.1	43.8	45.2	2x	63.8	72.9	36.4	48.1	2x
Ours	78.2	84.9	43.5	**58.0**	2.1x	69.3	**83.6**	43.5	52.3	2.02x

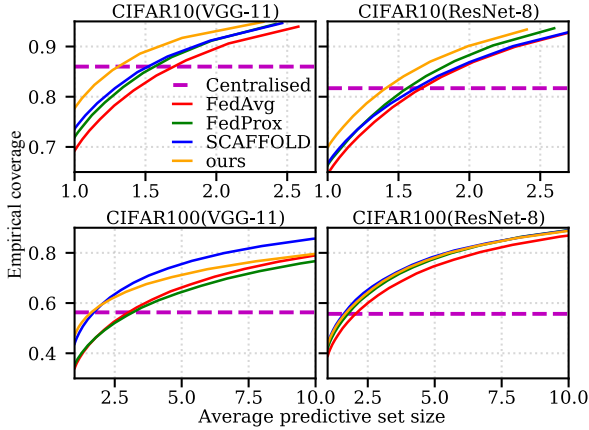cost of a slightly larger predictive set size.

Figure 3. Relation between average predictive size and empirical coverage when $\alpha = 0.1$. By slightly increasing the predictive set size, we can achieve a similar performance as the centralised model (Top-1 accuracy) even if the data are heterogeneously distributed across clients. Our method is similar to or faster than other approaches to surpass the centralised Top-1 accuracy.

5.3. Diversity and uniformity

We have shown that our algorithm achieves a better speedup and performance against the existing approaches with only lightweight modifications to FedAvg. We next investigate what factors lead to better accuracy. Specifi-
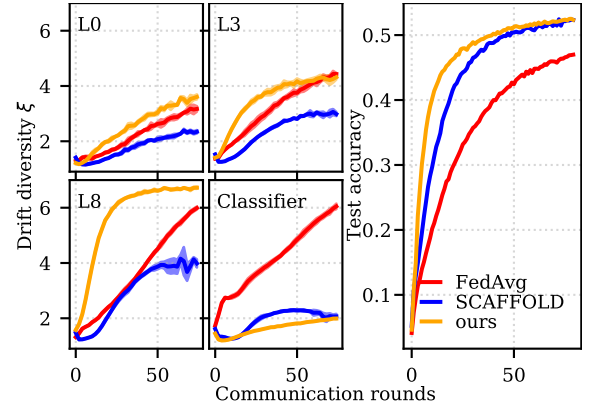

Figure 4. Drift diversity and learning curve for ResNet-8 on CIFAR100 with $\alpha = 1.0$. Compared to FedAvg, SCAFFOLD and our method can both improve the agreement between the classifiers. Compared to SCAFFOLD, our method results in a higher gradient diversity at the early stage of the communication, which tends to boost the learning speed as the curvature of the drift diversity seem to match the learning curve.

cally, we calculate the drift diversity ξ across clients after each communication round using Eq. 2 and average ξ across three runs. We show the result of using ResNet-8 and CIFAR100 with $\alpha = 1.0$ in Fig. 4.

Fig. 4 shows the drift diversity for different layers in ResNet-8 and the testing accuracy along the communication rounds. We observe that classifiers have the highest diver-

sity in FedAvg against other layers and methods. SCAF-FOLD, which applies control variate on the entire model, can effectively reduce the disagreement of the directions and scales of the averaged gradient across clients. Our proposed algorithm that performs variance reduction only on the classifiers can reduce the diversity of the classifiers even further but increase the diversity of the feature extraction layers. This high diversity tends to boost the learning speed as the curvature of the diversity movement (Fig. 4 left) seems to match the learning curve (Fig. 4 right). Based on this observation, we hypothesize that this diversity along the feature extractor and the uniformity of the classifier is the main reason for our better speedup.

To test this hypothesis, we perform an experiment where we use variance reduction starting from different layers of a neural network. If the starting position of the use of variance reduction influences the learning speed, it indicates where in a neural network we need more diversity and where we need more uniformity. We here show the result of using VGG-11 on CIFAR100 with $\alpha = 1.0$ as there are more layers in VGG-11. The result is shown in Fig. 5 where SVR:$16 \rightarrow 20$ is corresponding to our approach and SVR:$0 \rightarrow 20$ is corresponding to SCAFFOLD that applies variance reduction for the entire model. Results for using ResNet-8 is shown in Appendix. C.
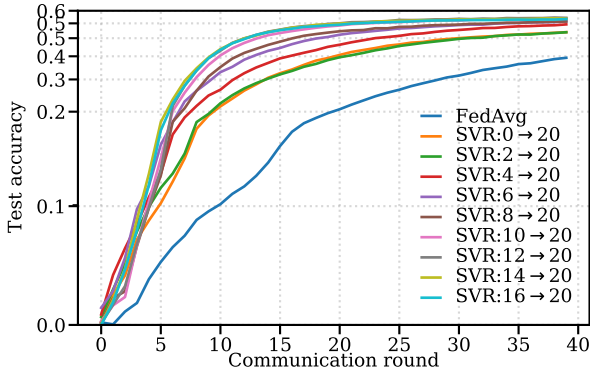

Figure 5. Influence of using stochastic variance reduction(SVR) on layers that start from different positions in a neural network on the learning speed. SVR:$0 \rightarrow 20$ applies variance reduction on the entire model (SCAFFOLD). SVR:$16 \rightarrow 20$ applies variance reduction from the layer index 16 to 20 (ours). The later we apply variance reduction, the better performance speedup we obtain. However, no variance reduction (FedAvg) performs the worst here.

We see from Fig. 5 that the deeper in a neural network we apply variance reduction, the better learning speedup we can obtain. There is no clear performance difference between where to activate the variance reduction when the layer index is over 10. However, applying no variance reduction (FedAvg) achieves by far the worst performance. We believe that these experimental results indicate that in a distributed optimization framework, to boost the learning

speed of an over-parameterized model, we need some levels of diversity in the middle and early layers for learning richer feature representation and some degrees of uniformity in the classifiers for making a less biased decision.

6. Conclusion

In this work, we studied stochastic gradient descent learning for deep neural network classifiers in a federated learning setting, where each client updates its local model using stochastic gradient descent on local data. A central model is periodically updated (by averaging local model parameters) and broadcast to the clients under a communication bandwidth constraint. When data is homogeneous across clients, this procedure is comparable to centralized learning in terms of efficiency; however, when data is heterogeneous, learning is impeded. Our hypothesis for the primary reason for this is that when the local models are out of alignment, updating the central model by averaging is ineffective and sometimes even destructive.

Examining the diversity across clients of their local model updates and their learned feature representations, we found that the misalignment between models is much stronger in the last few neural network layers than in the rest of the network. This finding inspired us to experiment with aligning the local models using a partial variance reduction technique applied only on the last layers, which we named FedPVR. We found that this led to a substantial improvement in convergence speed compared to the competing federated learning methods. In some cases, our method even outperformed centralized learning. We derived a bound on the convergence rate of our proposed method, which matches the rates for SGD when the gradient diversity across clients is sufficiently low. Compared with FedAvg, the communication cost of our method is only marginally worse, as it requires transmitting control variates for the last layers.

We believe our FedPVR algorithm strikes a good balance between simplicity and efficiency, requiring only a minor modification to the established FedAvg method; however, in our further research, we plan to pursue more optimal methods for aligning and guiding the local learning algorithms, *e.g.* using adaptive procedures. Furthermore, the degree of over-parameterization in the neural network layers (*e.g.* feature extraction vs bottlenecks) may also play an important role, which we would like to understand better.

Acknowledgements

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *CoRR*, abs/2111.04263, 2021. 1, 2, 6, 26

[2] Dan Alistarh, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: randomized quantization for communication-optimal stochastic gradient descent. *CoRR*, abs/1610.02132, 2016. 3

[3] Anastasios Nikolas Angelopoulos, Stephen Bates, Michael Jordan, and Jitendra Malik. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2021. 2, 3, 6

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020. 3

[5] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Fedavg with fine tuning: Local updates lead to representation learning, 2022. 2

[6] Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *CoRR*, abs/1407.0202, 2014. 3

[7] Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *CoRR*, abs/1812.04529, 2018. 2, 3, 5

[8] Aritra Dutta, El Houcine Bergou, Ahmed M. Abdelmoniem, Chen-Yu Ho, Atal Narayan Sahu, Marco Canini, and Panos Kalnis. On the discrepancy between the theoretical analysis and practical implementations of compressed communication for distributed deep learning. *CoRR*, abs/1911.08250, 2019. 3

[9] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10102–10111. IEEE, 2022. 2

[10] Malka N. Halgamuge, Moshe Zukerman, Kotagiri Ramamohanarao, and Hai Le Vu. An estimation of sensor energy consumption. *Progress in Electromagnetics Research B*, 12:259–295, 2009. 1, 2, 5

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 2

[12] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, 2013. 3

[13] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *CoRR*, abs/1912.04977, 2019. 1, 2

[14] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 13–18 Jul 2020. 1, 2, 3, 4, 5, 6, 11, 12, 13, 21

[15] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Better communication complexity for local SGD. *CoRR*, abs/1909.04746, 2019. 5

[16] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized SGD with changing topology and local updates. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5381–5393. PMLR, 13–18 Jul 2020. 5, 11, 13

[17] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *CoRR*, abs/1610.02527, 2016. 1, 2

[18] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. *CoRR*, abs/1905.00414, 2019. 4

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 2, 6

[20] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. *CoRR*, abs/2103.16257, 2021. 1, 3

[21] Zhize Li, Dmitry Kovalev, Xun Qian, and Peter Richtárik. Acceleration for compressed gradient descent in distributed and federated optimization, 2020. 3

[22] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 6, 25, 26

[23] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016. 6

[24] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *CoRR*, abs/2106.05001, 2021. 1, 2, 3, 4, 6

[25] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016. 2, 6

[26] Konstantin Mishchenko, Eduard Gorbunov, Martin Takác, and Peter Richtárik. Distributed learning with compressed gradient differences. *CoRR*, abs/1901.09269, 2019. 3

[27] Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, and Peter Richtárik. ProxSkip: Yes! local gradient steps provably lead to communication acceleration! finally! *International Conference on Machine Learning (ICML)*, 2022. 2

[28] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *CoRR*, abs/2010.15327, 2020. 2, 4

[29] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. *CoRR*, abs/2106.06042, 2021. 2

[30] Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with valid and adaptive coverage. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. 3

[31] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *CoRR*, abs/1812.06127, 2018. 1, 2, 6, 26

[32] Ohad Shamir, Nathan Srebro, and Tong Zhang. Communication efficient distributed optimization using an approximate newton-type method. *CoRR*, abs/1312.7853, 2013. 1, 2, 3

[33] Micah J. Sheller, Brandon Edwards, G. Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R. Colen, and Spyridon Bakas. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports*, 10(1):12598, Jul 2020. 1

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 2

[35] Sebastian U. Stich. Unified optimal analysis of the (stochastic) gradient method. *CoRR*, abs/1907.04232, 2019. 5, 11, 14

[36] Sebastian U. Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified SGD with memory. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 4452–4463. Curran Associates, Inc., 2018. 3

[37] Sebastian U. Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication. *CoRR*, abs/1909.05350, 2019. 15

[38] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 2, 5

[39] F Varno, M Saghayi, L Rafiee, S Gupta, S Matwin, and M Havaei. Minimizing client drift in federated learning via adaptive bias estimation. *ArXiv*, abs/2204.13170, 2022. 2

[40] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H. Brendan McMahan, Blaise Agüera y Arcas, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, Suhas N. Diggavi, Hubert Eichner, Advait Gadhikar, Zachary Garrett, Antonious M. Girgis, Filip Hanzely, Andrew Hard, Chaoyang He, Samuel Horváth, Zhouyuan Huo, Alex Ingerman, Martin Jaggi, Tara Javidi, Peter Kairouz, Satyen Kale, Sai Praneeth Karimireddy, Jakub Konečný, Sanmi Koyejo, Tian Li, Luyang Liu, Mehryar Mohri, Hang Qi, Sashank J. Reddi, Peter Richtárik, Karan Singhal, Virginia Smith, Mahdi Soltanolkotabi, Weikang Song, Ananda Theertha Suresh, Sebastian U. Stich, Ameet Talwalkar, Hongyi Wang, Blake E. Woodworth, Shanshan Wu, Felix X. Yu, Honglin Yuan, Manzil Zaheer, Mi Zhang, Tong Zhang, Chunxiang Zheng, Chen Zhu, and Wennan Zhu. A field guide to federated optimization. *CoRR*, abs/2107.06917, 2021. 2

[41] Yaodong Yu, Alexander Wei, Sai Praneeth Karimireddy, Yi Ma, and Michael I. Jordan. TCT: Convexifying federated learning using bootstrapped neural tangent kernels, 2022. 2, 5, 6

[42] Haoyu Zhao, Zhize Li, and Peter Richtárik. Fedpage: A fast local stochastic gradient method for communication-efficient federated learning. *CoRR*, abs/2108.04755, 2021. 2

Appendix

A. Technicalities

We first summarize the assumptions that are needed for the proof of convergence in the Section A.1 based on the previous literature [14, 16]. We then demonstrate the implications of these assumptions for our proof in the Section A.2. Following that, we summarize some of the useful and well-known lemmas in the Section A.3.

A.1. Assumptions

Assumptions on the objective function

For some of our results we assume (strong) convexity.

Assumption A-1 (μ-convex). *f_i is μ-convex for $\mu \geq 0$ and satisfies:*

$$\langle \nabla f_i(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \leq -(f_i(\boldsymbol{x}) - f_i(\boldsymbol{y}) + \frac{\mu}{2}||\boldsymbol{x} - \boldsymbol{y}||^2), \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, i \in [N]. \tag{A.1}$$

Here, we allow $\mu = 0$ (we refer generate convex case as when $\mu = 0$)

For all our theoretical analysis, we assume f_i is smooth.

Assumption A-2 (β-smooth). *f_i is β-smooth and satisfy:*

$$||\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})|| \leq \beta||\boldsymbol{x} - \boldsymbol{y}||, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, i \in [N], . \tag{A.2}$$

$$f_i(\boldsymbol{y}) \leq f_i(\boldsymbol{x}) + \langle \nabla f_i(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\beta}{2}||\boldsymbol{y} - \boldsymbol{x}||^2, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, i \in [N]. \tag{A.3}$$

Remark 1. *If functions $\{f_i\}$ are convex and $\boldsymbol{x}^*$ is a minimizer of f, then $\sum_i \nabla f_i(\boldsymbol{x}^*) = 0$, Assumption. A-2 implies:*

$$\frac{1}{N} \sum_{i=1}^{N} ||\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}^*)||^2 \leq 2\beta(f(\boldsymbol{x}) - f^*) \quad \forall \boldsymbol{x} \in \mathbb{R}^d. \tag{A.4}$$

Assumptions on the noise

For the convergence analysis of SGD on convex functions, it is usually enough to assume a bound on the noise on the optimum only([16,35]). Similarity, to express the function heterogeneity at the optimum point $\boldsymbol{x}^*$ (such a point always exist for the convex function), we make the following assumption.

Assumption A-3 (ζ-heterogeneity). *We define a measure of variance at the optimum $\boldsymbol{x}^*$ given N clients as :*

$$\zeta^2 := \frac{1}{N} \sum_{i=1}^{N} ||\nabla f_i(\boldsymbol{x}^*)||^2. \tag{A.5}$$

For the non-convex function, such an unique optimum point $\boldsymbol{x}^*$ does not necessarily exist, so we generalise assumption A-3 to:

Assumption A-4 ($\hat{\zeta}$-heterogeneity). *We assume that there exists constants $\hat{\zeta}$ such that $\forall \boldsymbol{x} \in \mathbb{R}^d$:*

$$\frac{1}{N} \sum_{i=1}^{N} ||\nabla f_i(\boldsymbol{x})||^2 \leq \hat{\zeta}^2. \tag{A.6}$$

Another assumption that is usually common is to assume the stochastic gradients are bounded as:

Assumption A-5 (bounded variance). *$g_i(\boldsymbol{x}) := \nabla f_i(\boldsymbol{x}; \mathcal{D}_i)$ is unbiased stochastic gradient of f_i with bounded variance:*

$$\mathbb{E}_{\mathcal{D}_i}[||g_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x})||^2] \leq \sigma^2, \quad \forall x \in \mathbb{R}^d, i \in [N]. \tag{A.7}$$

A.2. Implications of the assumptions

Given a binary mask p that has the same length as x, it holds that:

$$||p \odot x|| \leq ||x||. \tag{A.8}$$

Based on Eq. A.8, we have the following propositions:

Proposition 3 (Implications of the smoothness Assumption A-2). *Given a binary mask p, we define the block of weights that are variance reduced as β_p-smooth:*

$$||p \odot (\nabla f_i(x) - f_i(y))|| \leq \beta_p ||x - y||, \quad \forall x, y \in \mathbb{R}^d, i \in [N]. \tag{A.9}$$

If Assumption. A-2 holds, then it also holds that:

$$\beta_p \leq \beta. \tag{A.10}$$

Proposition 4 (Implication of the **convex** function heterogeneity Assumption A-3). *Given a binary mask p, we define the heterogeneity of the block of weights that are not variance reduced at the optimum x^* as:*

$$\zeta_{1-p}^2 := \frac{1}{N} \sum_{i=1}^{N} ||(1 - p) \odot \nabla f_i(x^*)||^2, \tag{A.11}$$

If Assumption A-3 holds, then it also holds that:

$$\zeta_{1-p}^2 \leq \zeta^2. \tag{A.12}$$

Proposition 5 (Implication of the **non-convex** function heterogeneity Assumption A-4). *Given a binary mask p, we assume the heterogeneity of the block of weights that are not variance reduced as:*

$$\frac{1}{N} \sum_{i=1}^{N} ||(1 - p) \odot \nabla f_i(x)||^2 \leq \hat{\zeta}_{1-p}, \tag{A.13}$$

If Assumption A-4 holds, then it also holds that:

$$\hat{\zeta}_{1-p}^2 \leq \hat{\zeta}^2. \tag{A.14}$$

Proposition 6 (Implication of the bounded variance Assumption. A-5). *If Assumption. A-5 holds, then:*

$$\mathbb{E}_{\mathcal{D}_i}[||p \odot (g_i(x) - \nabla f_i(x))||^2] \leq \sigma_p^2, \quad \forall x \in \mathbb{R}^d, i \in [N] \tag{A.15a}$$

$$\sigma_p^2 \leq \sigma^2. \tag{A.15b}$$

A.3. Some technical lemmas

We summarize some of the well-known lemmas in this subsection.

Lemma 2 (triangle inequality). *For arbitrary set of n vectors $\{v_i\}_{i=1}^{N}$ with $v_i \in \mathbb{R}^d$, then the following are true:*

$$||\sum_{i=1}^{N} v_i||^2 \leq N \sum_{i=1}^{N} ||v_i||^2. \tag{A.16}$$

$$||v_i + v_j||^2 \leq (1 + \alpha)||v_i||^2 + (1 + \alpha^{-1})||v_j||^2, \quad \forall \alpha > 0. \tag{A.17}$$

Lemma 3 (separating the mean and variance, [14, Lemma.4]). *Let $\{v_1, ..., v_\tau\}$ be τ random variables in $\mathbb{R}^d$ which are not necessarily dependent. First, suppose that their mean is $\mathbb{E}[v_i] = \Xi_i$ and the variance is bounded as $\mathbb{E}||v_i - \Xi_i||^2 \leq \sigma^2$, then the following holds:*

$$\mathbb{E}||\sum_{i=1}^{\tau} v_i||^2 \leq ||\sum_{i=1}^{\tau} \Xi_i||^2 + \tau^2 \sigma^2. \tag{A.18}$$

Now, suppose their condition mean is $\mathbb{E}[v_i | v_{i-1}, v_{i-1}, .., v_1] = \Xi_i$, the variance is bounded as $\mathbb{E}||v_i - \Xi_i||^2 \leq \sigma^2$, then we can show tighter bounds:

$$\mathbb{E}||\sum_{i=1}^{\tau} v_i||^2 \leq 2||\sum_{i=1}^{\tau} \Xi_i||^2 + 2\tau \sigma^2. \tag{A.19}$$

Proof. For any random variables X, $\mathbb{E}[X^2] = (\mathbb{E}[X - \mathbb{E}[X]])^2 + (\mathbb{E}[X])^2$, this implies:

$$\mathbb{E}||\sum_{i=1}^{\tau} \boldsymbol{v}_i||^2 \leq \mathbb{E}||\sum_{i=1}^{\tau} \boldsymbol{v}_i - \boldsymbol{\Xi}_i||^2 + \mathbb{E}||\sum_{i=1}^{\tau} \boldsymbol{\Xi}_i||^2 \leq \tau^2 \sigma^2 + ||\sum_{i=1}^{\tau} \boldsymbol{\Xi}_i||^2$$

For the second statement, $\boldsymbol{\Xi}_i$ is not determinant and is dependent on $[v_{i-1}, v_{i-2}, ..., v_1]$. Based on Lemma 2:

$$\mathbb{E}||\sum_{i=1}^{\tau} \boldsymbol{v}_i||^2 \leq 2\mathbb{E}||\sum_{i=1}^{\tau} \boldsymbol{v}_i - \boldsymbol{\Xi}_i||^2 + 2\mathbb{E}||\sum_{i=1}^{\tau} \boldsymbol{\Xi}_i||^2$$

$$\mathbb{E}||\sum_{i=1}^{\tau} \boldsymbol{v}_i - \boldsymbol{\Xi}_i||^2 = \sum_{i=1}^{\tau} \mathbb{E}||\boldsymbol{v}_i - \boldsymbol{\Xi}_i||^2 + 2\sum_{i,j} \mathbb{E}\langle \boldsymbol{v}_i - \boldsymbol{\Xi}_i, \boldsymbol{v}_j - \boldsymbol{\Xi}_j \rangle \leq \tau\sigma^2$$

$\square$

Lemma 4 (contrastive mapping, [14, Lemma.6])**.** *For any β-smooth and μ-strongly convex function h, points $\boldsymbol{x}$ and $\boldsymbol{y}$ in the domain of h, and step-size $\eta \leq \frac{1}{\beta}$, the following is true:*

$$||\boldsymbol{x} - \eta\nabla h(\boldsymbol{x}) + \eta\nabla h(\boldsymbol{y}) + \boldsymbol{y}|| \leq (1 - \mu\eta)||\boldsymbol{x} - \boldsymbol{y}||^2. \tag{A.21}$$

Proof.

$$\begin{aligned}
||\boldsymbol{x} - \eta\nabla h(\boldsymbol{x}) - y + \eta\nabla h(\boldsymbol{y})||^2 &= ||\boldsymbol{x} - \boldsymbol{y}||^2 + \eta^2||\nabla h(\boldsymbol{x}) - \nabla h(\boldsymbol{y})||^2 - 2\eta\langle \nabla h(\boldsymbol{x}) - \nabla h(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \\
&\leq ||\boldsymbol{x} - \boldsymbol{y}||^2 + (\eta^2\beta - 2\eta)\langle \nabla h(\boldsymbol{x}) - \nabla h(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \\
&\leq (1 - \eta\mu)||\boldsymbol{x} - \boldsymbol{y}||^2.
\end{aligned}$$

The second step uses the smoothness Assumption. A-2. The last step uses our bound on the step size $\eta \leq \frac{1}{\beta}$ (implies $\eta^2\beta - 2\eta \leq -\eta$) $\square$

Lemma 5 (Perturbed strong convexity, [14, Lemma.5])**.** *The following holds for any β-smooth and μ-strongly convex function h and any $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ in the domain of h:*

$$\langle \nabla h(\boldsymbol{x}), \boldsymbol{z} - \boldsymbol{y} \rangle \geq h(\boldsymbol{z}) - h(\boldsymbol{y}) + \frac{\mu}{4}||\boldsymbol{y} - \boldsymbol{z}||^2 - \beta||\boldsymbol{z} - \boldsymbol{y}||^2. \tag{A.22}$$

Proof. Given any $\boldsymbol{x}, \boldsymbol{y}$, and $\boldsymbol{z}$, we can get the following inequalities using smoothness and strong convexity of h:

$$\langle \nabla h(\boldsymbol{x}), \boldsymbol{z} - \boldsymbol{x} \rangle \geq h(\boldsymbol{z}) - h(\boldsymbol{x}) - \frac{\beta}{2}||\boldsymbol{z} - \boldsymbol{x}||^2,$$

$$\langle \nabla h(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{y} \rangle \geq h(\boldsymbol{x}) - h(\boldsymbol{y}) - \frac{\mu}{2}||\boldsymbol{x} - \boldsymbol{y}||^2.$$

Applying relaxed triangle inequality:

$$\frac{\mu}{2}||\boldsymbol{y} - \boldsymbol{x}||^2 \geq \frac{\mu}{4}||\boldsymbol{y} - \boldsymbol{z}||^2 - \frac{\mu}{2}||\boldsymbol{x} - \boldsymbol{z}||^2.$$

Combining the above three equations together:

$$\langle \nabla h(\boldsymbol{x}), \boldsymbol{z} - \boldsymbol{y} \rangle \geq h(\boldsymbol{z}) - h(\boldsymbol{y}) + \frac{\mu}{4}||\boldsymbol{y} - \boldsymbol{z}||^2 - \frac{\beta + \mu}{2}||\boldsymbol{z} - \boldsymbol{x}||^2.$$

The lemma follows since $\beta \geq \mu$ $\square$

Lemma 6 (Tunning the stepsize, [16, Lemma.17])**.** *For any parameters $r_0 \geq 0, b \geq 0, e \geq 0, \gamma \geq 0$, there exists constant step size $\eta \leq \frac{1}{\gamma}$ such that:*

$$\Psi_T := \frac{r_0}{\eta(T+1)} + b\eta + e\eta^2 \leq 2\left(\frac{br_0}{T+1}\right)^{\frac{1}{2}} + 2e^{1/3}\left(\frac{r_0}{T+1}\right)^{\frac{2}{3}} + \frac{\gamma r_0}{T+1}. \tag{A.24}$$

Proof. Choosing $\eta = \min\left\{ \left(\frac{r_0}{b(T+1)}\right)^{\frac{1}{2}}, \left(\frac{r_0}{e(T+1)}\right)^{\frac{1}{3}}, \frac{1}{\gamma} \right\} \leq \frac{1}{\gamma}$, we have three cases:

- $\eta = \frac{1}{\gamma}$ and is smaller than the other two terms, then

$$\Psi_T \leq \frac{dr_0}{T+1} + \frac{b}{\gamma} + \frac{e}{d^2} \leq \left(\frac{br_0}{T+1}\right)^{\frac{1}{2}} + \frac{dr_0}{T+1} + e^{1/3}\left(\frac{r_0}{T+1}\right)^{\frac{2}{3}},$$

- $\eta = \left(\frac{r_0}{b(T+1)}\right)^{\frac{1}{2}} < \left(\frac{r_0}{e(T+1)}\right)^{\frac{1}{3}}$, then:

$$\Psi_T \leq 2\left(\frac{br_0}{T+1}\right)^{\frac{1}{2}} + e\left(\frac{r_0}{b(T+1)}\right) \leq 2\left(\frac{br_0}{T+1}\right)^{\frac{1}{2}} + e^{1/3}\left(\frac{r_0}{T+1}\right)^{\frac{2}{3}},$$

- The last case $\eta = \left(\frac{r_0}{e(T+1)}\right)^{\frac{1}{3}} < \left(\frac{r_0}{b(T+1)}\right)^{\frac{1}{2}}$, then

$$\Psi_T \leq 2e^{1/3}\left(\frac{r_0}{T+1}\right) + b\left(\frac{r_0}{e(T+1)}\right)^{\frac{1}{3}} \leq 2e^{1/3}\left(\frac{r_0}{T+1}\right)^{\frac{2}{3}} + \left(\frac{br_0}{T+1}\right)^{\frac{1}{2}}.$$

$\square$

Lemma 7 (tunning the stepsize, [35, Lemma.2]). *If there exists two non-negative sequences $\{r_t\}_{t\geq 0}$, $\{s_t\}_{t\geq 0}$ and $a > 0$ that satisfy the relation:*

$$r_{t+1} \leq (1 - a\mu_t)r_t - b\mu_t s_t + c\mu_t^2, \tag{A.25}$$

Then there exists a constant step size $\mu_t \equiv \mu \leq \frac{1}{\gamma}$ such that for weight $w_t := (1 - a\mu)^{-t-1}$ and $W_T := \sum_{t=0}^{T} w_t$, it holds:

$$\frac{b}{W_t}\sum_{t=0}^{T} s_t w_t + ar_{T+1} = \tilde{\mathcal{O}}\left(\gamma r_0 \exp\left[-\frac{aT}{b}\right] + \frac{c}{aT}\right). \tag{A.26}$$

Proof. Choosing $\eta = \min\left\{ \frac{\ln\left(\max\{2, a^2 r_0 T^2/c\}\right)}{aT}, \frac{1}{\gamma} \right\}$, we have two cases:

- If $\frac{1}{\gamma} \geq \frac{\ln\left(\max\{2, a^2 r_0 T^2/c\}\right)}{aT}$, then we choose $\eta = \frac{\ln\left(\max\{2, a^2 r_0 T^2/c\}\right)}{aT}$:

$$\frac{b}{W_t}\sum_{t=0}^{T} s_t w_t + ar_{T+1} = \tilde{\mathcal{O}}\left(\frac{c}{aT}\right),$$

- If $\frac{1}{\gamma} < \frac{\ln\left(\max\{2, a^2 r_0 T^2/c\}\right)}{aT}$, then we chose $\eta = \frac{1}{\gamma}$:

$$\frac{b}{W_t}\sum_{t=0}^{T} s_t w_t + ar_{T+1} = \tilde{\mathcal{O}}\left(\gamma r_0 \exp\left[-\frac{aT}{\gamma}\right] + \frac{c}{aT}\right).$$

$\square$

B. Convergence of `FedPVR`

We first state the convergence theorem, then provide the proof for the convergence rate.

Remark 2. *Proving convergence for a randomly picked iterate $\bar{\boldsymbol{x}}^R \in \{\boldsymbol{x}^r\}_{r=0}^R$ is equivalent to show the convergence of a (weighted) average of the output criterion, e.g. $\frac{1}{R+1}\sum_{r=0}^R \mathbb{E}[f(\boldsymbol{x}^r)] - f^*$ for convex functions [37, Remark 17]. Following this, we assume there exists some weights of $\{w_r\}$ such that:*

$$\bar{\boldsymbol{x}}^R = \boldsymbol{x}^{r-1} \quad \text{with probability} \quad \frac{w_r}{\sum_\tau w_\tau} \quad \text{for} \quad r \in \{1, ..., R+1\}$$

Theorem II. *Suppose functions $\{f_i\}$ satisfies Assumption A-2 and A-5. Then in each of the cases, there exist weights $\{w_r\}$ and local step size η_l such that for any $\eta_g \geq 1$, the output of `FedPVR`, satisfies:*

- **Strongly convex:** *f_i satisfies Assumption. A-1 and A-3 for $\mu > 0$, $\eta_l \leq \min\left(\frac{1}{80K\eta_g\beta}, \frac{26}{20\mu K\eta_g}\right)$, $R \geq \max\left(\frac{20}{13}, \frac{160\beta}{\mu}\right)$, then*

$$\mathbb{E}[f(\bar{\boldsymbol{x}}^R)] - f(\boldsymbol{x}^*) \leq \tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu NKR}(1 + \frac{N}{\eta_g^2}) + \frac{\zeta_{1-p}^2}{\mu R} + \mu D \exp\left(-\min\left\{\frac{13}{20}, \frac{\mu}{160\beta}\right\} R\right)\right). \tag{B.1}$$

- **General convex:** *f_i satisfies Assumption. A-1 and A-3 for $\mu = 0$, $\eta_l \leq \frac{1}{80K\eta_g\beta}$, then:*

$$\mathbb{E}[f(\bar{\boldsymbol{x}}^R)] - f(\boldsymbol{x}^*) \leq \mathcal{O}\left(\frac{\sigma\sqrt{D}}{\sqrt{RKN}}\sqrt{1 + \frac{N}{\eta_g^2}} + \frac{\zeta_{1-p}\sqrt{D}}{\sqrt{R}} + \frac{\beta D}{R} + F\right). \tag{B.2}$$

- **Non convex:** *f_i satisfies Assumption. A-4, $\eta_l \leq \frac{1}{26K\eta_g\beta}$, and $R \geq 1$, then:*

$$\mathbb{E}||\nabla f(\bar{\boldsymbol{x}}^R)||^2 \leq \mathcal{O}\left(\frac{\sigma\sqrt{F}}{\sqrt{KNR}}\sqrt{\beta(\frac{N}{\eta_g^2} + 1)} + \frac{\hat{\zeta}_{1-p}\sqrt{F}}{\sqrt{R}}\sqrt{\frac{\beta}{\eta_g^2}} + \frac{\beta F}{R}\right) \tag{B.3}$$

Here $D := ||\boldsymbol{x}^0 - \boldsymbol{x}^*||^2$ and $F := f(\boldsymbol{x}^0) - f(\boldsymbol{x}^*)$

Instead of showing the convergence dependent on the number of round R, we can also state the convergence dependent on the expected error ϵ with choosing $\eta_g = \sqrt{N}$.

Corollary I. *Suppose function $\{f_i\}$ satisfy Assumption A-2 and Assumption A-5. Then the output of `FedPVR` has expected error smaller than ϵ for $\eta_g = \sqrt{N}$ and some values of η_l, R satisfying:*

- **Strongly convex:** *f_i satisfies Assumption. A-1 and A-3 for $\mu > 0$, $\eta_l \leq \min\left(\frac{1}{80K\eta_g\beta}, \frac{26}{20\mu K\eta_g}\right)$*

$$R = \tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu NK\epsilon} + \frac{\zeta_{1-p}^2}{\mu\epsilon} + \frac{\beta}{\mu}\right), \tag{B.4}$$

- **General convex:** *($\mu = 0$): f_i satisfies Assumption. A-1 and A-3 for $\mu = 0$, $\eta_l \leq \frac{1}{80K\eta_g\beta}$,*

$$R = \mathcal{O}\left(\frac{\sigma^2 D}{KN\epsilon^2} + \frac{\zeta_{1-p}^2 D}{\epsilon^2} + \frac{\beta D}{\epsilon} + F\right), \tag{B.5}$$

- **Non-convex:** *f_i satisfies Assumption. A-4, $\eta_l \leq \frac{1}{26K\eta_g\beta}$, and $R \geq 1$, then:*

$$R = \mathcal{O}\left(\frac{\beta\sigma^2 F}{KN\epsilon^2} + \frac{\beta\hat{\zeta}_{1-p}^2 F}{N\epsilon^2} + \frac{\beta F}{\epsilon}\right), \tag{B.6}$$

Where $D := ||\boldsymbol{x}^0 - \boldsymbol{x}^*||^2$ and $F := f(\boldsymbol{x}^0) - f^*$.

In the special case $p = 1$ ($\zeta_{1-p}^2 = 0$, $\hat{\zeta}_{1-p}^2 = 0$), then FedPVR is identical to Scaffold and we recover their convergence guarantees. In the strongly convex case, the effect of the heterogeneity of the block of weights that are not variance-reduced (ζ_{1-p}^2) becomes negligible if $\tilde{\mathcal{O}}\left(\frac{\zeta_{1-p}^2}{\epsilon}\right)$ is sufficiently smaller than $\tilde{\mathcal{O}}\left(\frac{\sigma^2}{NK\epsilon}\right)$. In such case, our rate becomes $\frac{\sigma^2}{\mu NK\epsilon} + \frac{1}{\mu}$, which recovers the SCAFFOLD in the strongly convex without sampling and further matches the SGD (with mini-batch size K on each worker), proving that it is at least as fast as the SGD.

Remark 3 (heterogeneity-diversity)**.** *Theoretically, in the non-convex case, the heterogeneity of the block of the weights that are not variance-reduced* $\hat{\zeta}_{1-p} := \frac{1}{N}\sum_{i=1}^N \|(\mathbf{1} - p) \odot \nabla f_i(x)\|^2$ *may slow down the convergence. In the manuscript, we observe that the diversity of the block of weights that are not variance-reduced* $\xi^r := \frac{\sum_{i=1}^N \|(\mathbf{1}-p)\odot(y_{i,K}^r - x^{r-1})\|^2}{\|\sum_{i=1}^N (\mathbf{1}-p)\odot(y_{i,K}^r - x^{r-1})\|^2}$ *tends to increase the convergence speed. The above two observations do not necessarily disagree as we define* $\hat{\zeta}_{1-p}$ *and* ξ *differently. The theory has not yet captured the phenomenon of the diversity of the non-variance-reduced weights improving convergence.*

B.1. Algorithm and additional definitions

We write our algorithm using the following notions: $\{y_i\}$ represents the client model, x is the aggregated server model, c_i, and c are the client and server control variate. The server maintains a global control variate c and each client maintains its own control variate c_i. N is the total number of clients. All the clients participate each round.

To give the proof for the convergence rate and simplify the algorithm, we rewrite it as shown in Algorithm II. The algorithm is identical to Algorithm-1 in the main paper except that c_i and c are now vectors with the same length as the model parameters. However, we only update the block of weights where the corresponding value in p equals 1. This version of the algorithm may consume more bits during the communication between the client and server, but the convergence rate is identical to Algorithm-1 in the main manuscript.

Algorithm II Partial variance reduction (FedPVR)

> **server**: initialise the server model x, control variate $c = 0$, and global step size η_g
> **client**: initialise the control variate $c_i = 0$, and local step size η_l
> **general**: set a binary mask $p \in \{0, 1\}^d$

1: **procedure** MODEL UPDATING
2:     **for** $r = 1 \rightarrow R$ **do**
3:         **Communicate x and c to all clients $i \in [N]$**
4:         **for client $i = 1 \rightarrow N$ in parallel do**
5:             $y_i \leftarrow x$
6:             **for $k = 1 \rightarrow K$ do**
7:                 compute minibatch gradient $g_i(y_i)$
8:                 $y_i \leftarrow y_i - \eta_l(g_i(y_i) - c_i + c)$
9:             **end for**
10:             $c_i \leftarrow c_i - c + \frac{1}{K\eta_l}p \odot (x - y_i)$
11:             **Communicate y_i, c_i to the server**
12:         **end for**
13:         $x \leftarrow (1 - \eta_g)x + \frac{1}{N}\sum_{i=1}^N y_i$
14:         $c \leftarrow \frac{1}{N}\sum_{i=1}^N c_i$
15:     **end for**
16: **end procedure**

This algorithm is identical to Algorithm I in the manuscript in terms of convergence rate but may consume more bits during the communication between the clients and server.

Given the above algorithm, every client performs the following updates:

- Starting from the shared global parameters $y_{i,0}^r = x^{r-1}$, we update the local parameters:

$$y_{i,k}^r = y_{i,k-1}^r - \eta_l v_{i,k}^r, \quad \text{where} \quad v_{i,k}^r := g_i(y_{i,k-1}^r) - c_i^{r-1} + c^{r-1}. \tag{B.7}$$

- update the control variate

$$c_i^r = c_i^{r-1} - c^{r-1} + \frac{1}{K\eta_l}p \odot (x^{r-1} - y_{i,K}^r). \tag{B.8}$$

- compute the new global model and global control variate:

$$\boldsymbol{x}^r = \boldsymbol{x}^{r-1} + \frac{\eta_g}{N}\sum_{i=1}^{N}(\boldsymbol{y}_{i,K}^r - \boldsymbol{x}^{r-1}) \quad \text{and} \quad \boldsymbol{c}^r = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{c}_i^r. \tag{B.9}$$

Definition 2. *We define the client drift $\mathcal{E}_r$ to be the amount of the movement between a client model $\boldsymbol{y}_{i,k}^r$ and the starting server model $\boldsymbol{x}^{r-1}$*

$$\mathcal{E}_r := \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{E}||\boldsymbol{y}_{i,k}^r - \boldsymbol{x}^{r-1}||^2. \tag{B.10}$$

Definition 3. *We define C_r as:*

$$C_r := \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}||\mathbb{E}[\boldsymbol{c}_i^r] - \nabla f_i(\boldsymbol{x}^*)||^2. \tag{B.11}$$

Definition 4. *We define the effective learning rate $\tilde{\eta}$ as:*

$$\tilde{\eta} = K\eta_l\eta_g. \tag{B.12}$$

B.2. Proof for the convergence rate for convex functions

Lemma 8. *For updates B.7 to B.9, we can bound the variance of the server update in any round r and any $\tilde{\eta} := \eta_l\eta_g K \geq 0$ as follows*

$$\mathbb{E}[||\boldsymbol{x}^r - \boldsymbol{x}^{r-1}||^2] \leq 8\tilde{\eta}^2\beta^2\mathcal{E}_r + 8\tilde{\eta}^2\beta(\mathbb{E}[f(\boldsymbol{x}^{r-1}) - f(\boldsymbol{x}^*)]) + 16\tilde{\eta}^2 C_{r-1} + \frac{8\tilde{\eta}^2\sigma^2}{KN} + \frac{16\tilde{\eta}^2\sigma_p^2}{KN}. \tag{B.13}$$

Proof. We know that $\boldsymbol{y}_{i,k}^r = \boldsymbol{x}^{r-1} - \eta_l\sum_k \boldsymbol{v}_{i,k}^r$, where $\boldsymbol{v}_{i,k}^r = g_i(\boldsymbol{y}_{i,k-1}^r) - \boldsymbol{c}_i + \boldsymbol{c}$, and $\boldsymbol{x}^r = \boldsymbol{x}^{r-1} + \frac{\eta_g}{N}\sum_i(\boldsymbol{x}^{r-1} - \eta_l\sum_k \boldsymbol{v}_{i,k}^r - \boldsymbol{x}^{r-1})$, so given $\tilde{\eta} = \eta_l\eta_g K$

$$\mathbb{E}[||\boldsymbol{x}^r - \boldsymbol{x}^{r-1}||^2] = \mathbb{E}[||\frac{\eta_g\eta_l}{N}\sum_{i,k}\boldsymbol{v}_{i,k}^r||^2] = \mathbb{E}[||\frac{\tilde{\eta}}{KN}\sum_{i,k}g_i(\boldsymbol{y}_{i,k-1}^r) - \boldsymbol{c}_i^{r-1} + \boldsymbol{c}^{r-1}||^2]. \tag{B.14}$$

We drop the round index r and $r-1$ everywhere, we have:

$$\begin{aligned}
\mathbb{E}[||\Delta\boldsymbol{x}||^2] &\leq 4\tilde{\eta}^2\mathbb{E}||\frac{1}{KN}\sum_{i,k}g_i(\boldsymbol{y}_{i,k-1}) - \nabla f_i(\boldsymbol{x})||^2 + 4\tilde{\eta}^2\mathbb{E}||\frac{1}{N}\sum_i\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}^*)||^2 \\
&\quad + 4\tilde{\eta}^2\mathbb{E}||\frac{1}{N}\sum_i\boldsymbol{c}_i - \nabla f_i(\boldsymbol{x}^*)||^2 + 4\tilde{\eta}^2\mathbb{E}||\boldsymbol{c}||^2 \\
&\leq 8\tilde{\eta}^2\mathbb{E}||\frac{1}{NK}\sum_{i,k}\nabla f_i(\boldsymbol{y}_{i,k-1}) - \nabla f_i(\boldsymbol{x})||^2 + \frac{4\tilde{\eta}^2}{N}\sum_i\mathbb{E}||\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}^*)||^2 \\
&\quad + 4\tilde{\eta}^2\mathbb{E}||\frac{1}{N}\sum_i\boldsymbol{c}_i - \nabla f_i(\boldsymbol{x}^*)||^2 + 4\tilde{\eta}^2\mathbb{E}||\boldsymbol{c}||^2 + \frac{8\tilde{\eta}^2\sigma^2}{KN} \\
&\leq \frac{8\tilde{\eta}^2}{NK}\sum_{i,k}\mathbb{E}||\nabla f_i(\boldsymbol{y}_{i,k-1}) - \nabla f_i(\boldsymbol{x})||^2 + \frac{4\tilde{\eta}^2}{N}\sum_i\mathbb{E}||\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}^*)||^2 \\
&\quad + 8\tilde{\eta}^2\mathbb{E}||\frac{1}{N}\sum_i\boldsymbol{c}_i - \nabla f_i(\boldsymbol{x}^*)||^2 + \frac{8\tilde{\eta}^2\sigma^2}{KN} \\
&\leq 8\tilde{\eta}^2\beta^2\mathcal{E}_r + 8\tilde{\eta}^2\beta(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*)) + 16\tilde{\eta}^2\frac{1}{N}\sum_i\mathbb{E}||\mathbb{E}[\boldsymbol{c}_i] - \nabla f_i(\boldsymbol{x}^*)||^2 \\
&\quad + \frac{8\tilde{\eta}^2\sigma^2}{KN} + \frac{16\tilde{\eta}^2\sigma_p^2}{KN} \\
&\leq 8\tilde{\eta}^2\beta^2\mathcal{E}_r + 8\tilde{\eta}^2\beta(\mathbb{E}[f(\boldsymbol{x}) - f(\boldsymbol{x}^*)]) + 16\tilde{\eta}^2 C_{r-1} + \frac{8\tilde{\eta}^2\sigma^2}{KN} + \frac{16\tilde{\eta}^2\sigma_p^2}{KN}.
\end{aligned} \tag{B.15}$$

The second step uses Jensen inequality. The third step uses Lemma 3 for separating the mean and variance and Lemma 2. The variance of $\frac{1}{KN}\sum_{i,k} g_i(\boldsymbol{y}_{i,k})$ is bounded by $\frac{\sigma^2}{KN}$, the variance of $\boldsymbol{c}_i = \frac{1}{K}\sum_k \boldsymbol{p}\odot g_i(\boldsymbol{y}_{i,k})$ is bounded by $\frac{\sigma_p^2}{K}$. The fourth step uses Lemma. 2 and by definition $\boldsymbol{c} = \frac{1}{N}\sum_i \boldsymbol{c}_i$. the last second step uses the smoothness and convexity of the function. The last step uses the definition of C_r and completes the proof. $\qquad\square$

Lemma 9. *For updates (B.7) - (B.9) with control variate update B.8 and Assumption. A-1 to Assumption. A-3, the following holds true for any $\tilde{\eta} := \eta_l \eta_g K \in [0, \frac{1}{\beta}]$*

$$C_r \leq 4\beta_p^2 \mathcal{E}_r + 8\beta_p(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*)) + 2\zeta_{1-p}^2 \tag{B.16}$$

Proof. We first simplify the expression for $\boldsymbol{c}_i$.

$$\begin{aligned}
\boldsymbol{c}_i^r &= \boldsymbol{c}_i^{r-1} - \boldsymbol{c}^{r-1} + \frac{1}{K\eta_l}\boldsymbol{p}\odot(\boldsymbol{x}^{r-1} - \boldsymbol{y}_{i,K}^r)\\
&= \boldsymbol{c}_i^{r-1} - \boldsymbol{c}^{r-1} + \frac{1}{K\eta_l}\boldsymbol{p}\odot(\eta_l \sum_k (g_i(\boldsymbol{y}_{i,k}^r) - \boldsymbol{c}_i^{r-1} + \boldsymbol{c}^{r-1}))\\
&= (\mathbf{1}-\boldsymbol{p})\odot(\boldsymbol{c}_i^{r-1} - \boldsymbol{c}^{r-1}) + \frac{1}{K}\sum_k \boldsymbol{p}\odot g_i(\boldsymbol{y}_{i,k}^r)\\
&= \frac{1}{K}\sum_k \boldsymbol{p}\odot g_i(\boldsymbol{y}_{i,k-1}^r).
\end{aligned} \tag{B.17}$$

The zero element in $(\mathbf{1}-\boldsymbol{p})$ is corresponding non-zero element in $\boldsymbol{c}_i$, so the element-wise multiplication results in zero. Taking the expectation on both side, we obtain: $\mathbb{E}[\boldsymbol{c}_i] = \frac{1}{K}\sum_k \nabla f_i(\boldsymbol{y}_{i,k})$

$$\begin{aligned}
C_r &= \frac{1}{N}\sum_i \mathbb{E}\|\frac{1}{K}\sum_k \boldsymbol{p}\odot\nabla f_i(\boldsymbol{y}_{i,k}) - \nabla f_i(\boldsymbol{x}^*)\|^2\\
&\leq \frac{2}{NK}\sum_{i,k}\mathbb{E}\|\boldsymbol{p}\odot(\nabla f_i(\boldsymbol{y}_{i,k}) - \nabla f_i(\boldsymbol{x}^*))\|^2 + \frac{2}{N}\sum_i \mathbb{E}\|(\mathbf{1}-\boldsymbol{p})\odot\nabla f_i(\boldsymbol{x}^*)\|^2\\
&\leq \frac{4}{NK}\sum_{i,k}\mathbb{E}\|\boldsymbol{p}\odot(\nabla f_i(\boldsymbol{y}_{i,k}) - \nabla f_i(\boldsymbol{x}))\|^2 + \frac{4}{N}\sum_i \mathbb{E}\|\boldsymbol{p}\odot(\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}^*))\|^2\\
&\quad + \frac{2}{N}\sum_i \mathbb{E}\|(\mathbf{1}-\boldsymbol{p})\odot\nabla f_i(\boldsymbol{x}^*)\|^2\\
&\leq 4\beta_p^2 \mathcal{E}_r + 8\beta_p(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*)) + 2\zeta_{1-p}^2.
\end{aligned} \tag{B.18}$$

The first and second step uses Jensen inequality. The last step uses the smoothness and convexity of the function. The definition of the function heterogeneity ζ_{1-p}^2 completes the proof.

$\qquad\square$

Lemma 10. *Suppose f_i satisfies the Assumption. A-1 to Assumption. A-5, then for any global $\eta_g \geq 1$, we can bound the drift as:*

$$\mathcal{E}_r \leq \frac{36\tilde{\eta}^2}{\eta_g^2}C_{r-1} + \frac{18\tilde{\eta}^2\beta(\mathbb{E}[f(\boldsymbol{x}^{r-1})] - f(\boldsymbol{x}^*))}{\eta_g^2} + \frac{36\tilde{\eta}^2\sigma_p^2}{K\eta_g^2} + \frac{3\tilde{\eta}^2\sigma^2}{K\eta_g^2} . \tag{B.19}$$

Proof. If $K = 1$, the lemma is trivially true as the left-hand side $\mathcal{E}_r = 0$ and the right hand size are positive. For $K > 1$, we build the bound for $\mathcal{E}_r$ recursively with dropping the round index:

$$\mathcal{T}_1 = \frac{1}{N}\sum_i \mathbb{E}[||\boldsymbol{y}_{i,k} - \boldsymbol{x}||^2] = \frac{1}{N}\sum_i \mathbb{E}[||\boldsymbol{y}_{i,k-1} - \eta_l g(\boldsymbol{y}_{i,k-1}) + \eta_l \boldsymbol{c}_i - \eta_l \boldsymbol{c} - \boldsymbol{x}||^2]$$

$$\leq \frac{1}{N}\sum_i \mathbb{E}[||\boldsymbol{y}_{i,k-1} - \eta_l \nabla f_i(\boldsymbol{y}_{i,k-1}) + \eta_l \nabla f_i(\boldsymbol{x}) - \eta_l \nabla f_i(\boldsymbol{x}) - \boldsymbol{x} + \eta_l \boldsymbol{c}_i - \eta_l \boldsymbol{c}||^2] + \eta_l^2\sigma^2$$

$$\overset{A.17}{\leq} \underbrace{\frac{1+a}{N}\sum_i \mathbb{E}[||\boldsymbol{y}_{i,k-1} - \eta_l \nabla f_i(\boldsymbol{y}_{i,k-1}) + \eta_l \nabla f_i(\boldsymbol{x}) - \boldsymbol{x}||^2]}_{\mathcal{T}_2} \tag{B.20}$$

$$+ \underbrace{(1+a^{-1})\eta_l^2 \frac{1}{N}\sum_i \mathbb{E}[||\boldsymbol{c}_i - \nabla f_i(\boldsymbol{x}) - \boldsymbol{c}||^2]}_{\mathcal{T}_3} + \eta_l^2\sigma^2\,.$$

$$\mathcal{T}_2 \overset{A.21}{\leq} \frac{1+a}{N}\sum_i \mathbb{E}[||\boldsymbol{y}_{i,k-1} - \boldsymbol{x}||^2]\,. \tag{B.21}$$

$$\mathcal{T}_3 \leq \frac{3}{N}\sum_i \mathbb{E}||\boldsymbol{c}_i - \nabla f_i(\boldsymbol{x}^*)||^2 + 3\mathbb{E}||\boldsymbol{c}||^2 + \frac{3}{N}\sum_i \mathbb{E}||\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}^*)||^2$$

$$\leq \frac{6}{N}\sum_i \mathbb{E}||\boldsymbol{c}_i - \nabla f_i(\boldsymbol{x}^*)||^2 + \frac{3}{N}\sum_i \mathbb{E}||\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}^*)||^2$$

$$\leq \frac{12}{N}\sum_i \mathbb{E}||\mathbb{E}[\boldsymbol{c}_i] - \nabla f_i(\boldsymbol{x}^*)||^2 + \frac{3}{N}\sum_i \mathbb{E}||\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}^*)||^2 + \frac{12\sigma_p^2}{K} \tag{B.22}$$

$$\leq 12C_{r-1} + 6\beta(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*)) + \frac{12\sigma_p^2}{K}\,.$$

The first step uses Jensen inequality. The second steps uses the definition of $\boldsymbol{c} = \frac{1}{N}\sum_i \boldsymbol{c}_i$. The last step uses the smoothness and convexity of the function and the definition of C_{r-1}. Then using $a = \frac{1}{K-1}$, we can simplify $\mathcal{T}_1$ as:

$$\mathcal{T}_1 = \frac{1}{N}\sum_i \mathbb{E}||\boldsymbol{y}_{i,k} - \boldsymbol{x}||^2$$

$$\leq (1 + \frac{1}{K-1})\frac{1}{N}\sum_i \mathbb{E}||\boldsymbol{y}_{i,k-1} - \boldsymbol{x}||^2 + 12K\eta_l^2 C_{r-1} + 6K\eta_l^2\beta(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*)) \tag{B.23}$$

$$+ 12\eta_l^2\sigma_p^2 + \eta_l^2\sigma^2\,.$$

Unrolling the recursion B.23, we get the following for any $k \in \{1, 2, ..., K\}$

$$\mathcal{T}_1 \leq (12K\eta_l^2 C_{r-1} + 6K\eta_l^2\beta(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*)) + 12\eta_l^2\sigma_p^2 + \eta_l^2\sigma^2)\sum_{\tau=0}^{k}(1 + \frac{1}{K-1})^\tau$$

$$\leq (12K\eta_l^2 C_{r-1} + 6K\eta_l^2\beta(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*)) + 12\eta_l^2\sigma_p^2 + \eta_l^2\sigma^2)3K \tag{B.24}$$

$$\leq \frac{36\tilde{\eta}^2}{\eta_g^2}C_{r-1} + \frac{18\tilde{\eta}^2\beta(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*))}{\eta_g^2} + \frac{36\tilde{\eta}^2\sigma_p^2}{K\eta_g^2} + \frac{3\tilde{\eta}^2\sigma^2}{K\eta_g^2}\,.$$

Averaging over K yields the lemma statement. $\qquad\square$

Lemma 11. *Progress in one round*: *Assume assumptions A-1-A-2 are true. The following hold for any step size satisfying* $\tilde{\eta} \leq \min\{\frac{1}{80\beta}, \frac{26}{20\mu}\}, \eta_g \geq 1$ *and* $\tilde{\eta} := K\eta_l\eta_g$:

$$\mathbb{E}||\boldsymbol{x}^r - \boldsymbol{x}^*||^2 + 18\tilde{\eta}^2 C_r \leq (1 - \frac{\mu\eta}{2})(\mathbb{E}||\boldsymbol{x}^{r-1} - \boldsymbol{x}^*||^2 + 18\tilde{\eta}^2 C_{r-1}) - \tilde{\eta}(\mathbb{E}[f(\boldsymbol{x}^{r-1})] - f(\boldsymbol{x}^*))$$
$$+ 36\tilde{\eta}^2 \zeta_{1-p}^2 + \frac{\tilde{\eta}^2\sigma^2}{KN}(24 + \frac{117N}{80\eta_g^2}). \tag{B.25}$$

Proof. Starting from the server update equation

$$\Delta x = -\frac{\tilde{\eta}}{KN}\sum_{i,k}(g_i(\boldsymbol{y}_{i,k-1}) - \boldsymbol{c}_i + \boldsymbol{c}), \quad \mathbb{E}[\Delta x] = -\frac{\tilde{\eta}}{KN}\sum_{i,k}g_i(\boldsymbol{y}_{i,k-1}). \tag{B.26}$$

Using Lemma 8 that bounds $\mathbb{E}||\Delta x||^2$, we can write:

$$\mathbb{E}||\boldsymbol{x} + \Delta\boldsymbol{x} - \boldsymbol{x}^*||^2 = \mathbb{E}||\boldsymbol{x} - \boldsymbol{x}^*||^2 + \mathbb{E}||\Delta\boldsymbol{x}||^2 - \frac{2\tilde{\eta}}{KN}\mathbb{E}\sum_{i,k} < \boldsymbol{x} - \boldsymbol{x}^*, \nabla f_i(\boldsymbol{y}_{i,k-1}) >$$

$$\leq \mathbb{E}||\boldsymbol{x} - \boldsymbol{x}^*||^2 \underbrace{- \frac{2\tilde{\eta}}{KN}\sum_{i,k}\mathbb{E} < \boldsymbol{x} - \boldsymbol{x}^*, \nabla f_i(\boldsymbol{y}_{i,k-1}) >}_{\mathcal{T}_5} + 8\tilde{\eta}^2\beta^2\mathcal{E}_r \tag{B.27}$$

$$+ 8\tilde{\eta}^2\beta(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*)) + 16\tilde{\eta}^2 C_{r-1} + \frac{8\tilde{\eta}^2\sigma^2}{KN} + \frac{16\tilde{\eta}^2\sigma_p^2}{KN}.$$

Given Lemma. 5, let $h = f_i, \boldsymbol{z} = \boldsymbol{x}^*, \boldsymbol{y} = \boldsymbol{x}, \boldsymbol{x} = \boldsymbol{y}_{i,k-1}^r$, then

$$\mathcal{T}_5 = \frac{2\tilde{\eta}}{KN}\mathbb{E}\sum_{i,k}\langle\nabla f_i(\boldsymbol{y}_{i,k}), \boldsymbol{x}^* - \boldsymbol{x}\rangle$$

$$\leq \frac{2\tilde{\eta}}{KN}\mathbb{E}\sum_{i,k}f_i(\boldsymbol{x}^*) - f_i(\boldsymbol{x}) - \frac{\mu}{4}||\boldsymbol{x} - \boldsymbol{x}^*||^2 + \beta||\boldsymbol{y}_{i,k-1}^r - \boldsymbol{x}||^2 \tag{B.28}$$

$$= -2\tilde{\eta}(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*)) - \frac{\tilde{\eta}\mu}{2}\mathbb{E}||\boldsymbol{x} - \boldsymbol{x}^*||^2 + \frac{2\tilde{\eta}\beta}{KN}\sum_{i,k}\mathbb{E}||\boldsymbol{y}_{i,k-1}^r - \boldsymbol{x}||^2$$

$$= -2\tilde{\eta}(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*)) - \frac{\tilde{\eta}\mu}{2}\mathbb{E}||\boldsymbol{x} - \boldsymbol{x}^*||^2 + 2\tilde{\eta}\beta\mathcal{E}_r.$$

Plugging $\mathcal{T}_5$ back to the equation, we obtain:

$$\mathbb{E}||\boldsymbol{x}^r - \boldsymbol{x}^*||^2 \leq (1 - \frac{\tilde{\eta}\mu}{2})(\mathbb{E}[||\boldsymbol{x} - \boldsymbol{x}^*||^2]) + (8\tilde{\eta}^2\beta - 2\tilde{\eta})(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*))$$
$$+ (2\tilde{\eta}\beta + 8\tilde{\eta}^2\beta^2)\mathcal{E}_r + 16\tilde{\eta}^2 C_{r-1} + \frac{8\tilde{\eta}^2\sigma^2}{KN} + \frac{16\tilde{\eta}^2\sigma_p^2}{KN}. \tag{B.29}$$

According to Lemma 9,

$$18\tilde{\eta}^2 C_r \leq 72\beta_p^2\tilde{\eta}^2\mathcal{E}_r + 144\beta_p\tilde{\eta}^2(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*)) + 36\tilde{\eta}^2\zeta_{1-p}^2. \tag{B.30}$$

According to Lemma 10,

$$3\tilde{\eta}\beta\mathcal{E}_r \leq 3\tilde{\eta}\beta(\frac{36\tilde{\eta}^2}{\eta_g^2}C_{r-1} + \frac{18\tilde{\eta}^2\beta(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*))}{\eta_g^2} + \frac{36\tilde{\eta}^2\sigma_p^2}{K\eta_g^2} + \frac{3\tilde{\eta}^2\sigma^2}{K\eta_g^2})$$

$$\leq \frac{108\tilde{\eta}^2\tilde{\eta}\beta}{\eta_g^2}C_{r-1} + \frac{54\tilde{\eta}\tilde{\eta}^2\beta^2(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*))}{\eta_g^2} + \frac{108\tilde{\eta}^2\tilde{\eta}\beta\sigma_p^2}{K\eta_g^2} + \frac{9\tilde{\eta}^2\tilde{\eta}\beta\sigma^2}{K\eta_g^2} \tag{B.31}$$

$$\leq \frac{27\tilde{\eta}^2 C_{r-1}}{20} + \frac{7\tilde{\eta}(\mathbb{E}[f(\boldsymbol{x})] - f(\boldsymbol{x}^*))}{800\eta_g^2} + \frac{27\tilde{\eta}^2\sigma_p^2}{20K\eta_g^2} + \frac{9\tilde{\eta}^2\sigma^2}{80K\eta_g^2}.$$

The last step follows $\tilde{\eta} \leq \frac{1}{80\beta}$.

Add the above three equations together:

$$
\begin{aligned}
\mathbb{E}||\boldsymbol{x}^r - \boldsymbol{x}^*||^2 + 18\tilde{\eta}^2 C_r &\leq (1 - \frac{\tilde{\eta}\mu}{2})(\mathbb{E}||\boldsymbol{x}^{r-1} - \boldsymbol{x}^*||^2 + 18\tilde{\eta}^2 C_{r-1}) \\
&\quad + (8\tilde{\eta}^2\beta - 2\tilde{\eta} + 144\beta_p\tilde{\eta}^2 + \frac{7\tilde{\eta}}{800\eta_g^2})(\mathbb{E}[f(\boldsymbol{x}^{r-1})] - f(\boldsymbol{x}^*)) \\
&\quad + (8\tilde{\eta}^2\beta^2 + 72\beta_p^2\tilde{\eta}^2 - \tilde{\eta}\beta)\mathcal{E}_r + 36\tilde{\eta}^2\zeta_{1-p}^2 \\
&\quad + (\frac{\tilde{\eta}\mu}{2} - \frac{13}{20})\tilde{\eta}^2 C_{r-1} \\
&\quad + \frac{\tilde{\eta}^2\sigma^2}{KN}(8 + \frac{9N}{80\eta_g^2}) + \frac{\tilde{\eta}^2\sigma_p^2}{KN}(16 + \frac{27N}{20\eta_g^2}) \\
&\leq (1 - \frac{\tilde{\eta}\mu}{2})(\mathbb{E}||\boldsymbol{x}^{r-1} - \boldsymbol{x}^*||^2 + 18\tilde{\eta}^2 C_{r-1}) - \tilde{\eta}(\mathbb{E}[f(\boldsymbol{x}^{r-1})] - f(\boldsymbol{x}^*)) \\
&\quad + 36\tilde{\eta}^2\zeta_{1-p}^2 + \frac{\tilde{\eta}^2\sigma^2}{KN}(24 + \frac{117N}{80\eta_g^2}).
\end{aligned}
\tag{B.32}
$$

As $0 \leq \beta_p \leq \beta$, the lemma follows from noting that $\tilde{\eta} \leq \frac{1}{80\beta}$ implies $(8\tilde{\eta}^2\beta^2 + 72\beta_p^2\tilde{\eta}^2 - \tilde{\eta}\beta) \leq 0$, $(8\tilde{\eta}^2\beta - 2\tilde{\eta} + 144\beta_p\tilde{\eta}^2) \leq 0$ and $\tilde{\eta} \leq \frac{26}{20\mu}$ implies $(\frac{\tilde{\eta}\mu}{2} - \frac{13}{20}) \leq 0$. The last step uses the preposition that $\sigma_p^2 \leq \sigma^2$. $\square$

The final rate for the general convex case For the general convex case ($\mu = 0$), we proceed by unrolling the recursive bound in Lemma. 11:

$$
\frac{1}{R}\sum_{r=1}^{R}\mathbb{E}[f(\boldsymbol{x}^{r-1})] - f(\boldsymbol{x}^*) \leq \frac{||\boldsymbol{x}^0 - \boldsymbol{x}^*||^2}{\tilde{\eta}R} + \frac{18\tilde{\eta}C_0}{R} + 36\tilde{\eta}\zeta_{1-p}^2 + \frac{\tilde{\eta}\sigma^2}{KN}(24 + \frac{117N}{80\eta_g^2}). \tag{B.33}
$$

Using Lemma 6, let $D := ||\boldsymbol{x}^0 - \boldsymbol{x}^*||^2$, $F := f(\boldsymbol{x}^0) - f(\boldsymbol{x}^*)$, and $\tilde{\eta} \leq \frac{1}{80\beta}$:

$$
\mathbb{E}[f(\bar{\boldsymbol{x}}^R)] - f(\boldsymbol{x}^*) \leq \mathcal{O}\left(\frac{\sigma\sqrt{D}}{\sqrt{RKN}}\sqrt{1 + \frac{N}{\eta_g^2}} + \frac{\zeta_{1-p}\sqrt{D}}{\sqrt{R}} + \frac{\beta D}{R} + F\right). \tag{B.34}
$$

The final rate for the strongly convex case. For the strongly convex case ($\mu > 0$), we use Lemma 7 letting $r^t = ||\boldsymbol{x}^t - \boldsymbol{x}^*||^2$, $a = \frac{\mu}{2}$, $b = 1$, $c = \zeta_{1-p}^2 + \frac{\sigma^2}{KN}(1 + \frac{N}{\eta_g^2})$, and $\frac{1}{d} = \min\left(\frac{1}{80\beta}, \frac{26}{20\mu}\right)$, $R \geq \max\{\frac{20}{13}, \frac{160\beta}{\mu}\}$, $D = ||\boldsymbol{x}^0 - \boldsymbol{x}^*||^2$ we obtain:

$$
\mathbb{E}[f(\bar{\boldsymbol{x}}^R)] - f(\boldsymbol{x}^*) \leq \tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu NKR}(1 + \frac{N}{\eta_g^2}) + \frac{\zeta_{1-p}^2}{\mu R} + \mu D\exp\left(-\min\left\{\frac{13}{20}, \frac{\mu}{160\beta}\right\}R\right)\right). \tag{B.35}
$$

Remark 4. *Following [14], we can bound C_0 with using warm-up strategy by $f(\boldsymbol{x}^0) - f(\boldsymbol{x}^*)$ and σ^2, so to simplify the convergence rate, we simply represent C_0 with F in the general convex case. For the strongly convex case, as the exponential term has a bigger influence on the convergence rate, we simply omit C_0 in the coefficient of the exponential term.*

B.3. Proof for the convergence rate for non-convex functions

We follow the same procedure to derive the convergence rate for the non-convex functions. We first bound the variance of the server update in Lemma. 12 and then the client drift in Lemma. 13. We then combine the results from these two lemmas to get the bound for the progress in one round in Lemma. 14. Following that, we give the convergence rate. To proceed the proof, we make the following definitions:

Recall the definition for client drift $\mathcal{E}_r$ and $\boldsymbol{c}_i$:

$$
\mathcal{E}_r := \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{E}||\boldsymbol{y}_{i,k}^r - \boldsymbol{x}^{r-1}||^2
$$

$$c_i^r = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{p} \odot g_i(\boldsymbol{y}_{i,k}^r)$$

Definition 5. *We define Θ_r as:*

$$\Theta_r := \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\|\mathbb{E}[\boldsymbol{c}_i] - \boldsymbol{p} \odot \nabla f_i(\boldsymbol{x})\|^2. \tag{B.37}$$

Lemma 12. *For updates from B.7 to B.9, the following holds true for any $\tilde{\eta} := \eta_l \eta_g K$*

$$\mathbb{E}\|\boldsymbol{x}^r - \boldsymbol{x}^{r-1}\|^2 \le 8\tilde{\eta}^2 \beta^2 \mathcal{E}_r + 16\tilde{\eta}^2 \Theta_{r-1} + 8\tilde{\eta}^2 \mathbb{E}\|\nabla f(\boldsymbol{x})\|^2 + \frac{6\tilde{\eta}^2 \sigma^2}{NK} + \frac{12\tilde{\eta}^2 \sigma_p^2}{NK}. \tag{B.38}$$

Proof.

$$\mathbb{E}\|\Delta \boldsymbol{x}\|^2 = \mathbb{E}\|\frac{\tilde{\eta}}{NK} \sum_{i,k} g_i(\boldsymbol{y}_{i,k}^r) - \boldsymbol{c}_i + \boldsymbol{c}\|^2$$

$$\le 2\mathbb{E}\|\frac{\tilde{\eta}}{NK} \sum_{i,k} \nabla f_i(\boldsymbol{y}_{i,k}) - \mathbb{E}[\boldsymbol{c}_i] + \mathbb{E}[\boldsymbol{c}]\|^2 + 2(\frac{3\tilde{\eta}^2 \sigma^2}{KN} + \frac{6\tilde{\eta}^2 \sigma_p^2}{KN})$$

$$\le 2\mathbb{E}\|\frac{\tilde{\eta}}{NK} \sum_{i,k} \nabla f_i(\boldsymbol{y}_{i,k}) - \nabla f_i(\boldsymbol{x}) + \boldsymbol{p} \odot \nabla f_i(\boldsymbol{x}) - \mathbb{E}[\boldsymbol{c}_i]$$

$$+ \mathbb{E}[\boldsymbol{c}] - \boldsymbol{p} \odot \nabla f(\boldsymbol{x}) + (\mathbf{1} - \boldsymbol{p}) \odot \nabla f_i(\boldsymbol{x}) + \boldsymbol{p} \odot \nabla f(\boldsymbol{x})\|^2 + \frac{6\tilde{\eta}^2 \sigma^2}{NK} + \frac{12\tilde{\eta}^2 \sigma_p^2}{NK} \tag{B.39}$$

$$\le \frac{8\tilde{\eta}^2}{NK} \sum_{i,k} \mathbb{E}\|\nabla f_i(\boldsymbol{y}_{i,k}) - \nabla f_i(\boldsymbol{x})\|^2$$

$$+ \frac{16\tilde{\eta}^2}{N} \sum_{i} \mathbb{E}\|\mathbb{E}[\boldsymbol{c}_i] - \boldsymbol{p} \odot \nabla f_i(\boldsymbol{x})\|^2 + 8\tilde{\eta}^2 \mathbb{E}\|\nabla f(\boldsymbol{x})\|^2 + \frac{6\tilde{\eta}^2 \sigma^2}{NK} + \frac{12\tilde{\eta}^2 \sigma_p^2}{NK}$$

$$\le 8\tilde{\eta}^2 \beta^2 \mathcal{E}_r + 16\tilde{\eta}^2 \Theta_{r-1} + 8\tilde{\eta}^2 \mathbb{E}\|\nabla f(\boldsymbol{x})\|^2 + \frac{6\tilde{\eta}^2 \sigma^2}{NK} + \frac{12\tilde{\eta}^2 \sigma_p^2}{NK}.$$

The second step uses Lemma 3. The last second step uses Jensen inequality. Given the definition of $\mathcal{E}_r$ and Θ_r, we complete the proof.

$\square$

Lemma 13. *Suppose f_i satisfy Assumption. A-2, A-4, and Assumption. A-5, then, for any global $\eta_g \ge 1$, we can bound the drift as:*

$$\mathcal{E}_r \le \frac{60\tilde{\eta}^2 \Theta_{r-1}}{\eta_g^2} + \frac{15\tilde{\eta}^2 \mathbb{E}\|\nabla f(\boldsymbol{x})\|^2}{\eta_g^2} + \frac{15\tilde{\eta}^2 \hat{\zeta}_{1-p}^2}{\eta_g^2} + \frac{60\tilde{\eta}^2 \sigma_p^2}{K\eta_g^2} + \frac{3\tilde{\eta}^2 \sigma^2}{K\eta_g^2}, \tag{B.40a}$$

$$\Theta_r \le \beta_p^2 \mathcal{E}_r. \tag{B.40b}$$

Proof. We first observe that if $K = 1$, then $\mathcal{E}_r = 0$ since $\boldsymbol{y}_{i,0} = \boldsymbol{x}$ for all $i \in [N]$ and the right hand side are all positive. Thus, the lemma is trivially true if $K = 1$ and we will assume $K > 1$ for the following proof. Starting from the update rule for $i \in [N], k \in [K]$

$$\mathbb{E}\|\boldsymbol{y}_{i,k} - \boldsymbol{x}\|^2 = \mathbb{E}\|\boldsymbol{y}_{i,k-1} - \eta_l(\nabla f_i(\boldsymbol{y}_{i,k-1}) - \boldsymbol{c}_i + \boldsymbol{c}) - \boldsymbol{x}\|^2 + \eta_l^2 \sigma^2$$

$$\le (1 + \frac{1}{K-1})\mathbb{E}\|\boldsymbol{y}_{i,k-1} - \boldsymbol{x}\|^2 + K\eta_l^2 \underbrace{\mathbb{E}\|\nabla f_i(\boldsymbol{y}_{i,k}) - \boldsymbol{c}_i + \boldsymbol{c}\|^2}_{\mathcal{T}_6} + \eta_l^2 \sigma^2. \tag{B.41}$$

The first step uses the Lemma. 2 with $\alpha = K - 1$

$$
\begin{aligned}
\mathcal{T}_6 = & \mathbb{E}||\boldsymbol{p} \odot \nabla f_i(\boldsymbol{y}_{i,k-1}) - \boldsymbol{p} \odot \nabla f_i(\boldsymbol{x}) - \boldsymbol{c}_i + \boldsymbol{p} \odot \nabla f_i(\boldsymbol{x}) + \boldsymbol{c} - \boldsymbol{p} \odot \nabla f(\boldsymbol{x}) + \boldsymbol{p} \odot \nabla f(\boldsymbol{x}) \\
& + (\boldsymbol{1} - \boldsymbol{p}) \odot \nabla f_i(\boldsymbol{y}_{i,k-1})||^2 \\
\leq & 5\mathbb{E}||\boldsymbol{p} \odot (\nabla f_i(\boldsymbol{y}_{i,k-1}) - \nabla f_i(\boldsymbol{x}))||^2 + 5\mathbb{E}||\boldsymbol{c}_i - \boldsymbol{p} \odot \nabla f_i(\boldsymbol{x})||^2 + \\
& + 5\mathbb{E}||\boldsymbol{c} - \boldsymbol{p} \odot \nabla f(\boldsymbol{x})||^2 + 5\mathbb{E}||\boldsymbol{p} \odot \nabla f(\boldsymbol{x})||^2 + 5\mathbb{E}||(\boldsymbol{1} - \boldsymbol{p}) \odot \nabla f_i(\boldsymbol{y}_{i,k-1})||^2 \\
\leq & 5\mathbb{E}||\boldsymbol{p} \odot (\nabla f_i(\boldsymbol{y}_{i,k-1}) - \nabla f_i(\boldsymbol{x}))||^2 + 20\mathbb{E}||\mathbb{E}[\boldsymbol{c}_i] - \boldsymbol{p} \odot \nabla f_i(\boldsymbol{x})||^2 \\
& + 5\mathbb{E}||\boldsymbol{p} \odot \nabla f(\boldsymbol{x})||^2 + 5\mathbb{E}||(\boldsymbol{1} - \boldsymbol{p}) \odot \nabla f_i(\boldsymbol{y}_{i,k-1})||^2 + \frac{20\sigma_p^2}{K} \\
\leq & 5\beta_p^2 \mathbb{E}||\boldsymbol{y}_{i,k-1} - \boldsymbol{x}||^2 + 20\mathbb{E}||\mathbb{E}[\boldsymbol{c}_i] - \boldsymbol{p} \odot \nabla f_i(\boldsymbol{x})||^2 + 5\mathbb{E}||\nabla f(\boldsymbol{x})||^2 \\
& + 5\mathbb{E}||(\boldsymbol{1} - \boldsymbol{p}) \odot \nabla f_i(\boldsymbol{y}_{i,k-1})||^2 + \frac{20\sigma_p^2}{K} \, .
\end{aligned}
\tag{B.42}
$$

The second step uses Jensen inequality. The third step uses the Lemma 3. The last step uses the smoothness of the function.

$$
\begin{aligned}
\frac{1}{N} \sum_i \mathbb{E}||\boldsymbol{y}_{i,k} - \boldsymbol{x}||^2 \leq & (1 + \frac{1}{K-1} + 5K\eta_l^2\beta_p^2) \frac{1}{N} \sum_i \mathbb{E}||\boldsymbol{y}_{i,k-1} - \boldsymbol{x}||^2 \\
& + 20K\eta_l^2 \underbrace{\frac{1}{N} \sum_i \mathbb{E}||\mathbb{E}[\boldsymbol{c}_i] - \boldsymbol{p} \odot \nabla f_i(\boldsymbol{x})||^2}_{\Theta_{r-1}} \\
& + 5K\eta_l^2 \mathbb{E}||\nabla f(\boldsymbol{x})||^2 + 5K\eta_l^2 \hat{\zeta}_{1-p}^2 \\
& + 20\eta_l^2\sigma_p^2 + \eta_l^2\sigma^2 \, .
\end{aligned}
\tag{B.43}
$$

The above step uses the definition of $\hat{\zeta}_{1-p}^2$. We next bound Θ_r using the smoothness Assumption. A-2.

$$
\begin{aligned}
\Theta_{r-1} = & \frac{1}{N} \sum_i \mathbb{E}||\frac{1}{K} \sum_k \boldsymbol{p} \odot \nabla f_i(\boldsymbol{y}_{i,k}) - \boldsymbol{p} \odot \nabla f_i(\boldsymbol{x})||^2 \\
\leq & \frac{1}{NK} \sum_{i,k} \mathbb{E}||\boldsymbol{p} \odot (\nabla f_i(\boldsymbol{y}_{i,k}) - \nabla f_i(\boldsymbol{x}))||^2 \\
\leq & \beta_p^2 \mathcal{E}_{r-1} \, .
\end{aligned}
\tag{B.44}
$$

Use the definition of Θ_r, we can then bound $\frac{1}{N} \sum_i \mathbb{E}||\boldsymbol{y}_{i,k} - \boldsymbol{x}||^2$:

$$
\begin{aligned}
\frac{1}{N} \sum_i \mathbb{E}||\boldsymbol{y}_{i,k} - \boldsymbol{x}||^2 \leq & \left(1 + \frac{1}{K-1} + \frac{5\tilde{\eta}^2\beta_p^2}{K\eta_g^2}\right) \frac{1}{N} \sum_i \mathbb{E}||\boldsymbol{y}_{i,k-1} - \boldsymbol{x}||^2 + 20K\eta_l^2\Theta_{r-1} \\
& + 5K\eta_l^2 \mathbb{E}||\nabla f(\boldsymbol{x})||^2 + 5K\eta_l^2 \hat{\zeta}_{1-p}^2 + 20\eta_l^2\sigma_p^2 + \eta_l^2\sigma^2 \\
\leq & (20K\eta_l^2\Theta_{r-1} + 5K\eta_l^2 \mathbb{E}||\nabla f(\boldsymbol{x})||^2 + 5K\eta_l^2 \hat{\zeta}_{1-p}^2 \\
& + 20\eta_l^2\sigma_p^2 + \eta_l^2\sigma^2) \sum_{\tau=0}^{k-1} \left(1 + \frac{1}{K-1} + \frac{5\tilde{\eta}^2\beta_p^2}{K\eta_g^2}\right)^{\tau} \\
\leq & (20K\eta_l^2\Theta_{r-1} + 5K\eta_l^2 \mathbb{E}||\nabla f(\boldsymbol{x})||^2 + 5K\eta_l^2 \hat{\zeta}_{1-p}^2 + 20\eta_l^2\sigma_p^2 + \eta_l^2\sigma^2)3K \\
\leq & \frac{60\tilde{\eta}^2\Theta_{r-1}}{\eta_g^2} + \frac{15\tilde{\eta}^2 \mathbb{E}||\nabla f(\boldsymbol{x})||^2}{\eta_g^2} + \frac{15\tilde{\eta}^2 \hat{\zeta}_{1-p}^2}{\eta_g^2} + \frac{60\tilde{\eta}^2\sigma_p^2}{K\eta_g^2} + \frac{3\tilde{\eta}^2\sigma^2}{K\eta_g^2} \, .
\end{aligned}
\tag{B.45}
$$

Averaging over K yields the lemma statement. $\qquad\square$

Lemma 14. *Suppose the updates B.7 to B.9 satisfy the smooth assumption and bounded variance. For any effective step size* $\tilde{\eta} := K\eta_l\eta_g$ *satisfying* $\tilde{\eta} \leq \frac{1}{26\beta}$:

$$\mathbb{E}[f(\boldsymbol{x} + \Delta\boldsymbol{x})] + 9\beta\tilde{\eta}^2\Theta_r \leq \mathbb{E}[f(\boldsymbol{x})] + 9\beta\tilde{\eta}^2\Theta_{r-1} - \tilde{\eta}\mathbb{E}||\nabla f(\boldsymbol{x})||^2$$
$$+ \frac{\tilde{\eta}^2\beta\sigma^2}{KN}\left(\frac{63N}{26\eta_g^2} + 9\right) + \frac{15\tilde{\eta}^2\beta\hat{\zeta}_{1-p}^2}{26\eta_g^2}. \tag{B.46}$$

Proof.

$$\mathbb{E}_{r-1}[f(\boldsymbol{x} + \Delta\boldsymbol{x})] \leq f(\boldsymbol{x}) + \langle\nabla f(\boldsymbol{x}), \mathbb{E}_{r-1}[\Delta\boldsymbol{x}]\rangle + \frac{\beta}{2}\mathbb{E}_{r-1}||\Delta\boldsymbol{x}||^2. \tag{B.47}$$

Recall that:

$$\mathbb{E}[\Delta\boldsymbol{x}] = -\frac{\tilde{\eta}}{KN}\sum_{i,k}\nabla f_i(\boldsymbol{y}_{i,k}). \tag{B.48}$$

Therefore

$$\mathbb{E}[f(\boldsymbol{x} + \Delta\boldsymbol{x})] - f(\boldsymbol{x}) \leq -\frac{\tilde{\eta}}{KN}\sum_{i,k}\langle\nabla f(\boldsymbol{x}), \nabla f_i(\boldsymbol{y}_{i,k})\rangle$$

$$+ \frac{\beta}{2}(8\tilde{\eta}^2\beta^2\mathcal{E}_r + 16\tilde{\eta}^2\Theta_{r-1} + 8\tilde{\eta}^2\mathbb{E}||\nabla f(\boldsymbol{x})||^2 + \frac{6\tilde{\eta}^2\sigma^2}{KN} + \frac{12\tilde{\eta}^2\sigma_p^2}{KN})$$

$$\leq -\frac{\tilde{\eta}}{2}||\nabla f(\boldsymbol{x})||^2 + \frac{\tilde{\eta}}{2}\mathbb{E}||\frac{1}{NK}\sum_{i,k}\nabla f_i(\boldsymbol{y}_{i,k}) - \nabla f(\boldsymbol{x})||^2 \tag{B.49}$$

$$+ 4\tilde{\eta}^2\beta^3\mathcal{E}_r + 8\tilde{\eta}^2\beta\Theta_{r-1} + 4\tilde{\eta}^2\beta\mathbb{E}||\nabla f(\boldsymbol{x})||^2 + \frac{3\tilde{\eta}^2\beta\sigma^2}{KN} + \frac{6\tilde{\eta}^2\beta\sigma_p^2}{KN}$$

$$\leq (4\tilde{\eta}^2\beta - \frac{\tilde{\eta}}{2})\mathbb{E}||\nabla f(\boldsymbol{x})||^2 + (\frac{\tilde{\eta}\beta^2}{2} + 4\tilde{\eta}^2\beta^3)\mathcal{E}_r + 8\tilde{\eta}^2\beta\Theta_{r-1} + \frac{3\tilde{\eta}^2\beta\sigma^2}{KN} + \frac{6\tilde{\eta}^2\beta\sigma_p^2}{KN}.$$

The second step uses the inequality that $-ab = \frac{1}{2}((b-a)^2 - a^2) - \frac{1}{2}b^2 \leq \frac{1}{2}((b-a)^2 - a^2)$ for any $a, b \in \mathbb{R}$

$$\tilde{\eta}\beta^2\mathcal{E}_r \leq \tilde{\eta}\beta^2\left(\frac{60\tilde{\eta}^2\Theta_{r-1}}{\eta_g^2} + \frac{15\tilde{\eta}^2\mathbb{E}||\nabla f(\boldsymbol{x})||^2}{\eta_g^2} + \frac{15\tilde{\eta}^2\hat{\zeta}_{1-p}^2}{\eta_g^2} + \frac{60\tilde{\eta}^2\sigma_p^2}{K\eta_g^2} + \frac{3\tilde{\eta}^2\sigma^2}{K\eta_g^2}\right)$$

$$\leq \frac{15\tilde{\eta}\Theta_{r-1}}{169} + \frac{4\tilde{\eta}\mathbb{E}||\nabla f(\boldsymbol{x})||^2}{169\eta_g^2} + \frac{15\tilde{\eta}^2\beta\hat{\zeta}_{1-p}^2}{26\eta_g^2} + \frac{30\tilde{\eta}^2\beta\sigma_p^2}{13K\eta_g^2} + \frac{3\tilde{\eta}^2\beta\sigma^2}{26K\eta_g^2}. \tag{B.50}$$

The above step uses $\tilde{\eta} \leq \frac{1}{26\beta}$

$$9\beta\tilde{\eta}^2\Theta_r \leq 9\beta\tilde{\eta}^2\beta_p^2\mathcal{E}_r. \tag{B.51}$$

Adding the above three equations together:

$$
\begin{aligned}
\mathbb{E}[f(\boldsymbol{x} + \Delta\boldsymbol{x})] + 9\beta\tilde{\eta}^2\Theta_r \leq{} & \mathbb{E}[f(\boldsymbol{x})] + 9\beta\tilde{\eta}^2\Theta_{r-1} \\
& + (4\tilde{\eta}^2\beta + \frac{4\tilde{\eta}}{169\eta_g^2} - \frac{\tilde{\eta}}{2})\mathbb{E}||\nabla f(\boldsymbol{x})||^2 \\
& + (4\tilde{\eta}^2\beta^3 + 9\beta\beta_p^2\tilde{\eta}^2 - \frac{\tilde{\eta}\beta^2}{2})\mathcal{E}_r \\
& + (\frac{15\tilde{\eta}}{169} - \beta\tilde{\eta}^2)\Theta_{r-1} \\
& + \frac{\tilde{\eta}^2\beta\sigma^2}{KN}(\frac{3N}{26\eta_g^2} + 3) + \frac{\tilde{\eta}^2\beta\sigma_p^2}{KN}(\frac{30N}{13\eta_g^2} + 6) + \frac{15\tilde{\eta}^2\beta\hat{\zeta}_{1-p}^2}{26\eta_g^2} \\
\leq{} & \mathbb{E}[f(\boldsymbol{x})] + 9\beta\tilde{\eta}^2\Theta_{r-1} - \tilde{\eta}\mathbb{E}||\nabla f(\boldsymbol{x})||^2 \\
& + \frac{\tilde{\eta}^2\beta\sigma^2}{KN}(\frac{3N}{26\eta_g^2} + 3) + \frac{\tilde{\eta}^2\beta\sigma_p^2}{KN}(\frac{30N}{13\eta_g^2} + 6) + \frac{15\tilde{\eta}^2\beta\hat{\zeta}_{1-p}^2}{26\eta_g^2} \\
\leq{} & \mathbb{E}[f(\boldsymbol{x})] + 9\beta\tilde{\eta}^2\Theta_{r-1} - \tilde{\eta}\mathbb{E}||\nabla f(\boldsymbol{x})||^2 + \frac{\tilde{\eta}^2\beta\sigma^2}{KN}(\frac{63N}{26\eta_g^2} + 9) + \frac{15\tilde{\eta}^2\beta\hat{\zeta}_{1-p}^2}{26\eta_g^2}.
\end{aligned}
\tag{B.52}
$$

From the first equation, we see $\tilde{\eta} \leq \frac{1}{26\beta}$ implies $(4\tilde{\eta}^2\beta^3 + 9\tilde{\eta}^2\beta\beta_p^2 - \frac{\tilde{\eta}\beta^2}{2}) \leq 0$, $4\tilde{\eta}^2\beta + \frac{4\tilde{\eta}}{169\eta_g^2} - \frac{\tilde{\eta}}{2} \leq 0$, and $\frac{15\tilde{\eta}}{169} - \beta\tilde{\eta}^2 \leq 0$. The last step uses the inequality $\sigma_p^2 \leq \sigma^2$

$\square$

Convergence rate: Average the above equation over R rounds:

$$
\begin{aligned}
\frac{1}{R}\sum_{r=1}^R \mathbb{E}||\nabla f(\boldsymbol{x}^r)||^2 ={} & \frac{\mathbb{E}[f(\boldsymbol{x}^R)] - \mathbb{E}[f(\boldsymbol{x}^0)]}{\tilde{\eta}R} + 9\beta\tilde{\eta}\Theta_0 + \frac{\tilde{\eta}\beta\sigma^2}{KN}(\frac{63N}{26\eta_g^2} + 9) + \frac{15\tilde{\eta}\beta\hat{\zeta}_{1-p}^2}{26\eta_g^2} \\
\leq{} & \sigma\sqrt{\frac{\mathbb{E}[f(\boldsymbol{x}^R)] - \mathbb{E}[f(\boldsymbol{x}^0)]}{KNR}}\sqrt{\beta\left(\frac{63N}{\eta_g^2} + 9\right)} + \hat{\zeta}_{1-p}\sqrt{\frac{\mathbb{E}[f(\boldsymbol{x}^R)] - \mathbb{E}[f(\boldsymbol{x}^0)]}{R}}\sqrt{\frac{15\beta}{26\eta_g^2}} \\
& + \frac{26\beta(\mathbb{E}[f(\boldsymbol{x}^R)] - \mathbb{E}[f(\boldsymbol{x}^0)])}{R}.
\end{aligned}
\tag{B.53}
$$

Note that if we initialize $\boldsymbol{c}_i^0 = g_i(\boldsymbol{x}^0)$, $\Theta_0 = \frac{1}{N}\sum_{i=1}^N \mathbb{E}||\mathbb{E}[\boldsymbol{c}_i^0] - \nabla f_i(\boldsymbol{x}^0)||^2 = 0$. The last step follows from using a stepsize $\tilde{\eta} = \min\left(\frac{1}{26\beta}, \sqrt{\frac{\mathbb{E}[f(\boldsymbol{x}^R)] - \mathbb{E}[f(\boldsymbol{x}^0)]}{R}}\sqrt{\frac{1}{\frac{\beta\sigma^2}{KN}(\frac{63N}{\eta_g^2} + 9) + \frac{15\beta\hat{\zeta}_{1-p}^2}{26\eta_g^2}}}\right)$ and Lemma. 6.

Therefore, for non-convex functions:

$$
\mathbb{E}||\nabla f(\bar{\boldsymbol{x}}^R)||^2 \leq \mathcal{O}\left(\frac{\sigma\sqrt{F}}{\sqrt{KNR}}\sqrt{\beta\left(\frac{N}{\eta_g^2} + 1\right)} + \frac{\hat{\zeta}_{1-p}\sqrt{F}}{\sqrt{R}}\sqrt{\frac{\beta}{\eta_g^2}} + \frac{\beta F}{R}\right).
\tag{B.54}
$$

C. Extra experimental setup and results

C.1. Additional experimental setups

Data distribution We follow the procedure as described in [22] to simulate the data heterogeneity scenario. There are around 5000 images per client. We use $1\%(500)$ of the images as the validation dataset to tune the hyperparameters (learning rate and schedule). Fig. C.1 shows data distribution across clients using CIFAR10 and CIFAR100. When $\alpha = 0.1$, some clients may only have data from a single class.
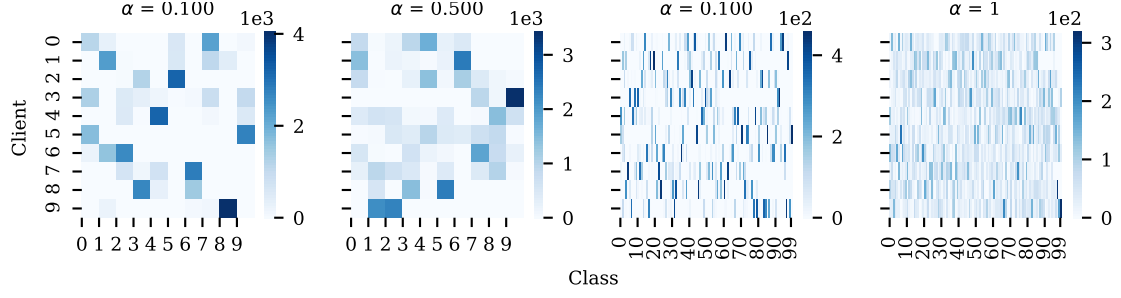
Figure C.1. Different levels of data heterogeneity. The left two figures are from CIFAR10 and the right two figures are from CIFAR100.

Conformal prediction Let d_i denotes a data point and l_i denotes the corresponding label. Conformal prediction aims to produce a predictive set $\mathcal{C}_\kappa(d_i)$ such that:

$$P(l_i \in \mathcal{C}_\kappa(d_i)) \geq 1 - \kappa \tag{C.1}$$

where $\kappa \in (0,1)$ specifies the desired converge level. To find the predictive set $\mathcal{C}$, we use a threshold τ on the predicted probability to indicate which predictions are included in the prediction set. Specifically, we assume that the server can have a small portion of a dataset in the same domain as the datasets in each client (e.g. the validation dataset). As we only suggest conformal prediction as an effective tool when the task is sensitive and producing a wrong prediction is dangerous (e.g. chemical hazards detection), we think it is reasonable to make such an assumption to guarantee the performance.

Architecture We use two types of neural networks in our paper: VGG-11 and ResNet8. We simply adopt the architectures that are used in [22]. Detailed information about these two architectures can be found at `https://github.com/epfml/federated-learning-public-code`.

Hyperparameters The learning rate and learning rate schedule are shown in the following two tables. Most of the experiments use a learning rate of 0.1 with a constant schedule, which means that each client always uses a learning rate of 0.1 along the communication rounds. To stabilize the training procedure, we apply momentum with a factor of 0.9 on the block of weights that are not variance-reduced. We use 10 clients with full participation. The number of local epochs per round is 10. For FedProx [31], we tune the temperature parameter μ from $\{0.1, 0.5, 1.0\}$. For FedDyn [1], the penalization parameter (α as used in [1]) is 0.001 for CIFAR10 and 0.01 for CIFAR100. We tune the centralised learning experiment with constant learning rate and momentum parameter from $\{0.9, 0.95\}$.

Table C.1. The learning rate and learning rate schedule for CIFAR10 experiments where c represents `constant` and m represents multi-step decay

| | VGG-11 | | | | | | ResNet8 | | | | | |
| | FedAvg | | SCAFFOLD | | FedPVR | | FedAvg | | SCAFFOLD | | FedPVR | |
	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.5$
LR	0.05	0.1	0.05	0.1	0.05	0.1	0.1	0.2	0.1	0.3	0.1	0.3
LR-schedule	c	c	m	c	c	c	c	c	c	c	c	c

Table C.2. The learning rate and learning rate schedule for CIFAR100 experiments where c represents `constant` and m represents `multi-step decay` and *cos* represents `cosine-decay`

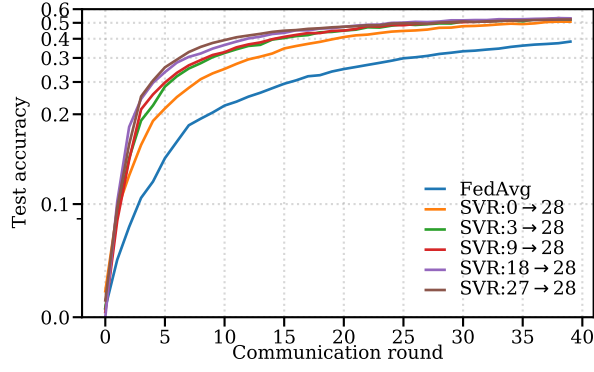| | VGG-11 | | | | | | ResNet8 | | | | | |
| | FedAvg | | SCAFFOLD | | FedPVR | | FedAvg | | SCAFFOLD | | FedPVR | |
	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 0.1$	$\alpha = 1.0$
LR	0.1	0.05	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.1	0.1
LR-schedule	c	c	c	c	c	c	c	c	cos	c	c	c

C.2. Additional experimental results

Figure C.2. Influence of using variance reduction on layers that start from different positions in a neural network on the learning speed. `SVR:0→28` applies variance reduction on the entire model, which corresponds to `SCAFFOLD`. `SVR:27→28` applies variance reduction from the layer index 27 to 28, which corresponds to our method. The later we apply variance reduction, the better performance speedup we obtain. However, no variance reduction (`FedAvg`) performs the worst here.
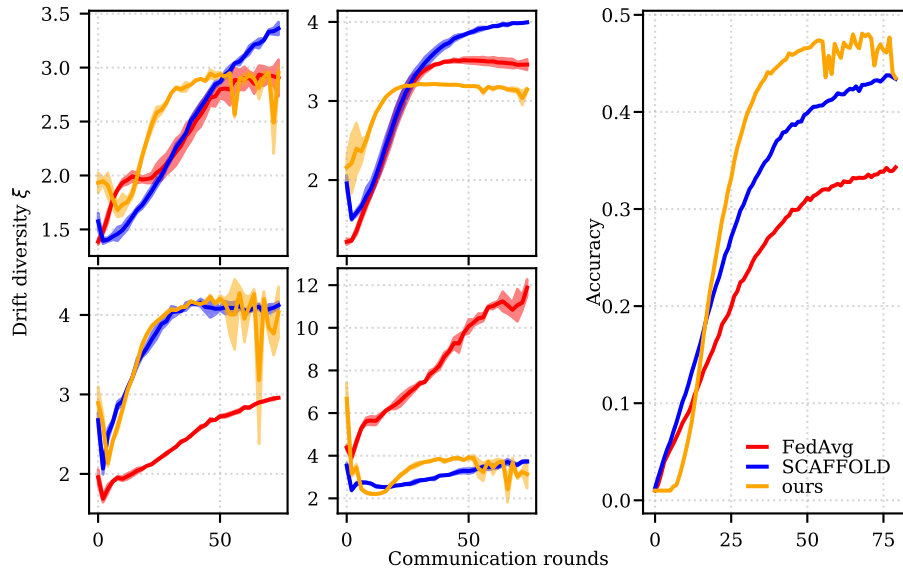

Figure C.3. Drift diversity and learning curve for VGG-11 on CIFAR100 with $\alpha = 0.1$. Compared to FedAvg, SCAFFOLD and our method can both improve the agreement between the classifiers. Compared to SCAFFOLD, our method results in a higher gradient diversity at the early stage of the communication, which tends to boost the learning speed as the curvature of the drift diversity seem to match the learning curve.
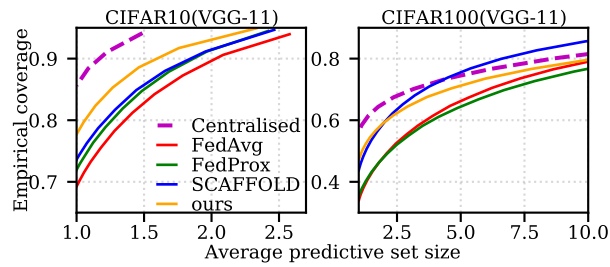

Figure C.4. Relation between average predictive size and empirical coverage using centralised empirical coverage instead of centralised Top-1 accuracy.