FLECS-CGD: A Federated Learning Second-Order Framework via Compression and Sketching with Compressed Gradient Differences

Artem Agafonov AGAFONOV.AD@PHYSTECH.EDU
Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
Moscow Institute of Physics and Technology, Dolgoprudny, Russia

Brahim Erraji BRAHIM.ERRAJI@MBZUAI.AC.AE
Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE

Martin Takáč TAKAC.MT@GMAIL.COM
Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE

Abstract

In the recent paper FLECS (Agafonov et al, FLECS: A Federated Learning Second-Order Framework via Compression and Sketching), the second-order framework FLECS was proposed for the Federated Learning problem. This method utilize compression of sketched Hessians to make communication costs low. However, the main bottleneck of FLECS is gradient communication without compression. In this paper, we propose the modification of FLECS with compressed gradient differences, which we call FLECS-CGD (FLECS with **C**ompressed **G**radient **D**ifferences) and make it applicable for stochastic optimization. Convergence guarantees are provided in strongly convex and nonconvex cases. Experiments show the practical benefit of proposed approach.

1. Introduction

In this paper, we focus on the stochastic federated learning problem, where the objective function is the empirical loss of overall n workers:

$$\min_{w\in\mathbb{R}^d}\left\{F(w) := \frac{1}{n}\sum_{i=1}^{n} f_i(w)\right\}, \tag{1}$$

where $f_i(w) = \mathbb{E}_{\xi\sim\mathcal{D}_i}\left[f(w,\xi)\right]$, with F being a general loss function (parametrized by $w \in \mathbb{R}^d$ and ξ) associated with the data stored on the i-th machine, n is number of machines. The distributions $\mathcal{D}_1, \dots, \mathcal{D}_n$ may differ on each machine, so means the functions $f_1, \dots, f_n$ can have completely different minimizers. In particular, $\mathbb{E}_{\xi\sim\mathcal{D}_i}\left[\nabla f_i(w^*,\xi)\right] \neq 0$, where w^* is an optimal solution of (1).

Problems of this nature are ubiquitous and rise naturally whenever multiple computing nodes can be connected [12, 18, 22]. For example, such problems arise in distributed machine learning, robotics, resource allocation, optimal transport, and other applications [10, 16, 17, 19–21, 26, 27, 31–33, 36–38].

The most popular approaches for the problem (1) might be first-order algorithms. Early work in that field includes [7, 8, 23]. To reduce communication burden methods with gradient compression were proposed [3, 6, 13, 24]. Other techniques like the usage of momentum [25, 40], variance reduction [15, 28, 29], and adaptive learning rates [35] were recently proposed in the literature.

Second-order methods are also proposed for the Federated Learning setup. Generally, these methods can be divided into two groups based on heterogeneous/homogeneous data setting assumptions. Algorithms in the first group [1, 4, 9, 11, 39] usually utilize statistical similarity, which means that the local function f_i approximates the global objective F well. Methods FedNL [34] and FLECS [2] work in truly heterogeneous setup, which makes them more practical. However, FedNL seems impractical for large-scale problems because of high memory requirements on devices. Indeed, in FedNL it is assumed that each device (e.g. mobile phone) should store $d \times d$ Hessian approximation locally, which is impossible for large d. FLECS tackles this problem by using a sketching technique and switching memory costs from machines to the server. Therefore, FLECS does not require storing Hessians locally. In both FedNL and FLECS, compression is applied only to the (sketched) difference of Hessian and its approximation. The goal of this work is to add gradient compression to FLECS. That allows reducing communication complexity. Moreover, compared to [2], we show that FLECS and CG-FLECS work in the general stochastic distributed optimization problem.

Contribution We briefly describe our contributions below. First of all, we make FLECS appliable to the the stochastic federated learning. Secondly, we propose FLECS-CGD – the modification of FLECS with gradient compression. This improves communication complexity from $O(cmd + 32d + 32m^2)$ (float32) to $O(cmd + cd + 32m^2)$, where m is a user-defined memory size and c is a number of bits per one value after compression (typically $c \ll 32$). Thirdly, we provide theoretical convergence guarantee in non-convex and strongly convex cases. Finally, our numerical experiments show practical benefit of the proposed approach.

Organization The rest of the paper is organized as follows. In Section 2, we introduce main notations and definitions. Then, In Section 3 we present our framework FLECS with compressed gradients for stochastic distributed optimization problem (1). Section 4 is dedicated to the convergence analysis of the proposed method (all proofs can be found in appendix). Finally, numerical experiments are provided in Section 5.

2. Preliminaries

Definition 1 *A differentiable function $F : \mathbb{R}^d \to \mathbb{R}^d$ is called μ-strongly convex for $\mu > 0$, if for all $x,\ y \in \mathbb{R}^d$:* $\quad F(x) + \langle \nabla F(x), y - x\rangle + \frac{\mu}{2}\|x - y\|^2 \leq F(y)$.

Definition 2 *A differentiable function $F : \mathbb{R}^d \to \mathbb{R}^d$ is called L-smooth for $L > 0$, if for all $x,\ y \in \mathbb{R}^d$:* $\quad F(y) \leq F(x) + \langle \nabla F(x), y - x\rangle + \frac{L}{2}\|x - y\|^2$.

Definition 3 *By $\mathcal{U}(\omega)$ ($\omega > 0$) we define the class of unbiased compression operators $Q :\ \mathbb{R}^d \to \mathbb{R}^d$ satisfying*

$$\mathbb{E}_Q\left[Q(x)\right] = x,\ \ \mathbb{E}_Q\left[\|Q(x)\|^2\right] \leq (\omega + 1)\|x\|^2, \tag{2}$$

for all $x \in \mathbb{R}^d$.

3. FLECS-CGD : FLECS with Gradient Compression

In this section, we describe the main steps of the proposed method.FLECS-CGD is a modification of FLECS with gradient compression. FLECS-CGD is listed as Algorithm 1. Detailed information about FLECS can be found in the original article [2].

The algorithm is initialized with user-defined memory size $m \ll d$, n vectors $h_0^i \in \mathbb{R}^d$ and n matrices $B_0^i \in \mathbb{R}^{d\times d}$. Each h_0^i (B_0^i) represents an approximation to the local gradient (Hessian) for the i-th worker.

In the beginning of iteration k each worker receives $B_k^i S_k,\ w_k$ from the server. Then, i-th worker samples $S_k \in \mathbb{R}^{d\times m}$, g_k^i such that $\mathbb{E}\left[g_k^i | w_k\right] = \nabla f_i(w_k)$, $H_k^i S_k$ such that $\mathbb{E}\left[H_k^i S_k | w_k, S_k\right] =$

$\nabla^2 f_k^i(w_k)S_k$. Note that S_k is the same for all machines and the server; we guarantee it by setting the random seed to be equal to the iteration number k.

Next, to utilize error-feedback technique each worker calculates

$$c_k^i = Q(g_k^i - h_k^i) \in \mathbb{R}^d,\ M_k^i = S_k^T Y_k^i \in \mathbb{R}^{m\times m},\ C_k^i = \mathcal{C}(Y_k^i - B_k^i S_k) \in \mathbb{R}^{d\times m}. \tag{3}$$

Then the i-th worker sends compressed differences $c_k^i,\ C_k^i$ and M_k^i to the server.

The server receives $c_k^i,\ C_k^i,\ M_k^i$ from all $i = 1,\dots,n$ workers. Firstly, $\widetilde{g}_k^i = c_k^i + h_k^i$ and $\widetilde{Y}_k^i = C_k^i + B_k^i S_k$ are computed. Then, the server computes new Hessian approximation $B_{k+1}^i,\ i = 1,\dots,n$ via Truncated L-SR1 update (Algorithm 2) or Direct update (Algorithm 3).

At the very end of k-th iteration the server forms

$$\begin{aligned} \widetilde{g}_k &= \tfrac{1}{n}\textstyle\sum_{i=1}^n \widetilde{g}_k^i, & \widetilde{Y}_k &:= \tfrac{1}{n}\textstyle\sum \widetilde{Y}_k^i = \tfrac{1}{n}\sum_{i=1}^n (C_k^i + H_k^i S_k), \\ M_k &:= \tfrac{1}{n}\textstyle\sum_{i=1}^n M_k^i = S_k \nabla^2 F(w_k) S_k^T, & B_{k+1} &:= \tfrac{1}{n}\textstyle\sum_{i=1}^n B_{k+1}^i. \end{aligned}$$

Finally, the main node calculates new iterate w_{k+1} via update rule $w_{k+1} = w_k + \alpha_k p_k$ where $\alpha_k > 0$ is the step-size. Search direction p_k can be computed via truncated inverse Hessian approximation step (Algorithm 4) or via FedSONIA (Algorithm 5) step.

Communication complexity Omitting both gradient and matrix compressions communication complexities per node of FLECS and FLECS-CGD are the same. Both algorithms need to send d dimensional vector, one $m \times m$ matrix and one $d \times m$ matrix. However, when using compression, the situation is different. Assuming that float data type is used, FLECS-CG reduces communication complexity of FLECS $O(cmd + 32d + 32m^2)$ to $O(cmd + cd + 32m^2)$, where c is number of bits per digit. It is important for the practical case of small memory sizes m. Indeed, if we set $m = 1$, then FLECS-CG communication complexity is $O(cd)$ which is much smaller than FLECS's $O(32d)$.

Step complexity [2] The the worker step's complexity consists of m Hessian-vector products and matrix multiplication $(O(md^2))$. The total complexity of either Hessian approximation (Algorithms 2, 3) update is $O(nmd^2)$. So the server step's complexity depends on options for the search direction: $O(d^3 + nmd^2)$ for Truncated Inverse Hessian approximation (Algorithm 4) and $O(nmd^2)$ for FedSONIA (Algorithm 5) .

4. FLECS-CGD : Convergence Analysis

In this section, we present the convergence theory for FLECS-CGD . All proofs can be found in Appendix B. Let w_0 be an initial point and w^* be a solution: $w^\star = \arg\min_{w\in\mathbb{R}^d} F(w)$, and $F^\star = F(w^\star)$.

Before proving the convergence of FLECS-CGD for different classes of functions, we will cite a few key assumptions and lemmas.

Assumption 1 *The function F is twice continuously differentiable.*

First, we focus on strongly convex case and present assumptions for this setting.

Assumption 2 *Each function $f_i(w)$ is μ-strongly convex and L-smooth $\mu I \preceq \nabla^2 f_i(w) \preceq LI$.*

Assumption 3 *Each g_k^i in Algorithm 1 has bounded variance $\mathbb{E}\left[\|g_k^i - \nabla f_i(w_k)\|\right] \le \sigma_i^2$, $\forall k \ge 0,\ i = 1,\dots,n$ for constants $\sigma_i < \infty$, $\sigma^2 := \frac{1}{n}\sum_{i=1}^n \sigma_i^2$.*

The following theorem establishes global linear convergence of FLECS-CGD under strong convexity.

Algorithm 1 FLECS-CGD

Require: w_0 – starting point, m – memory size, B_0^i – initial Hessian approximations for each worker $i = 1 \ldots n$ on the server, $0 < \omega < \Omega$ – truncation constants.

for $k = 0, 1, \ldots$ **do**

  On i**-th machine:**

  collect $B_k^i S_k,\ w_k$ from the server;

  sample $S_k \in \mathbb{R}^{d \times m}$, g_k^i such that $\mathbb{E}\left[g_k^i | w_k\right] = \nabla f_i(w_k)$, $H_k^i S_k$ such that $\mathbb{E}\left[H_k^i S_k | w_k, S_k\right] = \nabla^2 f_i(w_k) S_k$;

  let $Y_k^i := H_k^i S_k$, and compute $M_k^i := S_k^T Y_k^i$;

  send $c_k^i = Q(g_k^i - h_k^i),\ M_k^i,\ C_k^i = \mathcal{C}(Y_k^i - B_k^i S_k)$ to the server.

  select stepsize γ_k and update $h_{k+1}^i := h_k^i + \gamma_k c_k^i$

  On the server:

  sample S_k;

  collect $C_k^i,\ M_k^i,\ c_k^i\ i = 1 \ldots n$ from workers;

  compute $\widetilde{g}_k^i = c_k^i + h_k^i,\ \widetilde{Y}_k^i = C_k^i + B_k^i S_k$;

  compute B_{k+1}^i via Algorithm 2 or select learning rate β_k and compute B_{k+1}^i via Algorithm 3;

  form $\widetilde{g}_k, \widetilde{Y}_k, M_k, B_{k+1}$ as average over workers of $\widetilde{g}_k^i, \widetilde{Y}_k^i, M_k^i, B_{k+1}^i,\ i = 1, \ldots, n$;

  compute search direction p_k via Algorithm 4 or 5;

  select stepsize α_k and set $w_{k+1} = w_k + \alpha_k p_k$;

  sample $S_{k+1} \in \mathbb{R}^{d \times m}$;

  send $w_k, B_{k+1} S_{k+1}$ to all workers.

end for

Theorem 4 *Suppose that Assumption 1, 2, 3 holds. Let* $Q \in \mathcal{U}(\omega)$. *Let* $\{w_k\}$ *be the iterates generated by Algorithm 1, where* $0 < \alpha_k = \alpha \le \frac{5\mu\mu_1}{2L^2\mu_2^2\left(1+\frac{\omega}{n}\right)}$ *and* $0 < \gamma_k = \gamma \le \frac{1}{\omega+1}$. *Define the Lyapunov function* $\Psi_{k+1} = (F(w_{k+1}) - F(w_*)) + \frac{cL\mu_2^2\alpha^2}{n}\sum\limits_{i=1}^{n} \mathbb{E}_Q\left[\|h_{k+1}^i - h_*^i\|^2\right]$ *for* $0 < c = \min\left\{\frac{1-\frac{\alpha\mu\mu_1}{2}-\frac{\omega}{n}}{1-\gamma}; \frac{\mu}{2\gamma L}\right\}$. *Then for all* $k \ge 0$*:*

$$\mathbb{E}_Q\left[\Psi_k\right] \le \left(1 - \frac{\alpha\mu\mu_1}{2}\right)^{k+1}\Psi_0 + \left(\frac{\omega+1}{2n} + \gamma c\right)\frac{2L\mu_2^2\alpha}{\mu\mu_1}\sigma^2. \tag{4}$$

Now we present the assumptions for nonconvex case.

Assumption 4 *The function* F *is* L*-smooth.*

Assumption 5 *(Bounded data dissimilarity). There exists constant* $\zeta \ge 0$ *such that* $\forall x \in \mathbb{R}^d$ $\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x) - \nabla F(x)\|_2^2 \le \zeta^2$. *In particular,* $\zeta = 0$*, implies that all datasets stored in the* n *devices are drawn from the same data distribution* $\mathcal{D}$.

The following result shows that FLECS-CGD converges in the nonconvex case.

Theorem 5 *Suppose that Assumption 4, 5 holds. Let* $Q \in \mathcal{U}(\omega)$*, Let* $S = \{w_0, w_1, \ldots, w_{k-1}\}$ *be generated using Algorithm 1, and* $\bar{w}$ *be sampled uniformly at random from* S*, for* $\alpha \le \sqrt{\frac{n}{2Lw(w+1)\mu_2^2}}$

and $\gamma_k \leq \frac{1+\sqrt{1-\frac{2L\alpha^2 w(w+1)\mu_2^2}{n}}}{2(w+1)}$, *and a parameter* c *such as* $c < \frac{\mu_1}{L\alpha\gamma_k} - \frac{\mu_2^2}{2\gamma_k}$ *we have:*

$$\mathbb{E}_Q\left[\|\nabla F(\bar{w})\|_2^2\right] \leq 2\frac{\bar{\kappa}^0}{k\alpha\left(2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k\right)} + \frac{4cL\alpha}{2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k}\zeta^2 + \frac{\mu_2^2 + 2c}{2\mu_1 - \left(L\alpha\mu_2^2\right) - 2cL\alpha}L\sigma^2 \tag{5}$$

with $\bar{\kappa}^k = F(w_k) - F^* + c\frac{L\alpha^2}{2}\frac{1}{n}\sum_{i=1}^n \|h_k^i - h_*^i\|_2^2$

Corollary 6 *Set* $\gamma_k = \gamma$, $\alpha = \frac{2\mu_1 - 1}{L(\mu_2^2+2c\gamma)\sqrt{K}}$ *and* $h_0 = 0$, *after* K *iterations of algorithm 1, in the nonconvex setting, the error* ϵ *is at worst* $\frac{2}{\sqrt{K}}\frac{L(\mu_2^2+2c\gamma)}{(2\mu_1-1)}\bar{\kappa}^0 + \frac{1}{\sqrt{K}}\frac{4c(2\mu_1-1)}{\mu_2^2+2c\gamma}\zeta^2 + \frac{1}{\sqrt{K}}\frac{\left(\mu_2^2+2c\gamma\right)(2\mu_1-1)}{\mu_2^2+2c\gamma}\sigma^2$.

5. Experiments

We analyse the practical benefit of the proposed approach on regularized logistic regression problem for binary classification

$$\min_{w\in\mathbb{R}^d}\left\{\frac{1}{n}\sum_{i=1}^n \frac{1}{r}\sum_{j=1}^r \log(1+\exp(-b_{ij}a_{ij}^T w)) + \frac{\mu}{2}\|w\|^2\right\},$$

$\{a_{ij}, b_{ij}\}_{j\in[m]}$ are data points on i-th device. We use three datasets from the LIBSVM library [5]: gisette-scale (5000 features) and real-sim (20958 features), and a9a (123 features) (Appendix B).

FLECS-CGD vs FLECS. In this experiment we illustrate that gradient compression improves the convergence of FLECS in terms of communicated gradients per node. We use random dithering compressor for sketched Hessian and gradients with 128 levels and ∞ norm. Hyperparameters of both methods are set the same: initial Hessian approximation $B_k^i = 0,\ \omega = 10^{-5},\ \Omega = 10^8,\ \alpha = 1,\ \beta = 1,$ for FedSONIA update we set $\rho = \frac{1}{\Omega}$. For FLECS-CGD $\gamma = 1$. We choose memory sizes $m = 1,\ 2,\ 4,\ 8$. Random dithering is used as compressor with $s = 64$ levels and $p = \infty$-norm. Both FLECS-CGD and FLECS show their best performance with $m = 1$. Because of additional gradient compression, FLECS-CGD outperforms FLECS in this low memory-size setup.

According to the paper [2] FLECS outperforms FedNL, DIANA and ADIANA. In the experiments (Figure 5), we showed that gradient-compression improves FLECS convergence since it reduces communication complexity. Additional experiments can be found in Appendix B.

References

[1] Artem Agafonov, Pavel Dvurechensky, Gesualdo Scutari, Alexander Gasnikov, Dmitry Kamzolov, Aleksandr Lukashevich, and Amir Daneshmand. An accelerated second-order method for distributed stochastic optimization. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2407–2413, 2021. doi: 10.1109/CDC45484.2021.9683400.

[2] Artem Agafonov, Dmitry Kamzolov, Rachael Tappenden, Alexander Gasnikov, and Martin Takáč. Flecs: A federated learning second-order framework via compression and sketching. *arXiv preprint arXiv:2206.02009*, 2022.

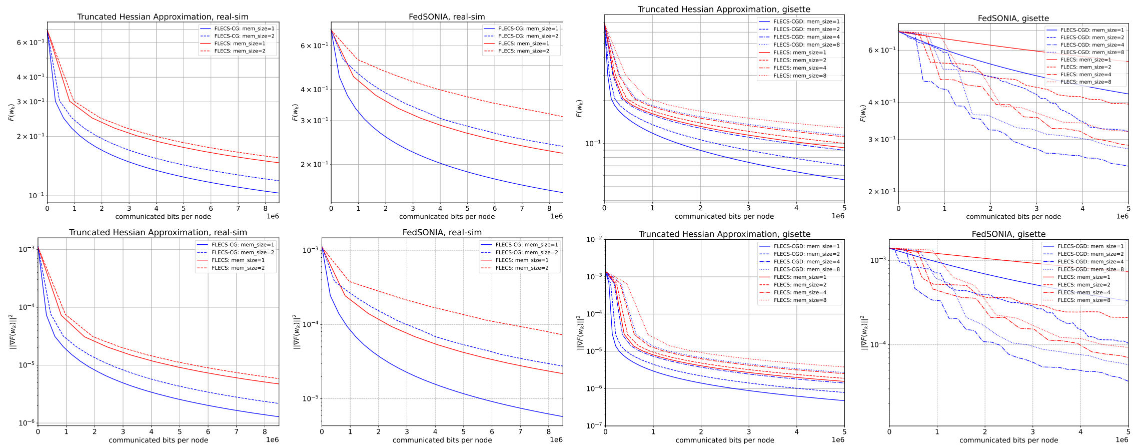

Figure 1: Comparison of objective function $F(w_k)$ and the squared norm of gradient $\|\nabla F(w_k)\|^2$ for FLECS and FLECS-CGD .

[3] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in Neural Information Processing Systems*, 30, 2017.

[4] Brian Bullins, Kshitij Patel, Ohad Shamir, Nathan Srebro, and Blake E Woodworth. A stochastic newton algorithm for distributed convex optimization. *Advances in Neural Information Processing Systems*, 34, 2021.

[5] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[6] Yicheng Chen, Rick S Blum, Martin Takac, and Brian M Sadler. Distributed learning with sparsified gradient differences. *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[7] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtarik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[8] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. URL https://arxiv.org/abs/1610.05492.

[9] Amir Daneshmand, Gesualdo Scutari, Pavel Dvurechensky, and Alexander Gasnikov. Newton method over networks is fast up to the statistical precision, 2021.

[10] Pavel Dvurechenskii, Darina Dvinskikh, Alexander Gasnikov, Cesar Uribe, and Angelia Nedich. Decentralize and randomize: Faster algorithm for wasserstein barycenters. In *Advances in Neural Information Processing Systems*, pages 10760–10770, 2018.

[11] Pavel Dvurechensky, Dmitry Kamzolov, Aleksandr Lukashevich, Soomin Lee, Erik Ordentlich, César A Uribe, and Alexander Gasnikov. Hyperfast second-order local solvers for efficient statistically preconditioned distributed optimization. *arXiv preprint arXiv:2102.08246*, 2021.

[12] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction, 2018. URL https://arxiv.org/abs/1811.03604.

[13] Samuel Horváth, Chen-Yu Ho, Ludovit Horvath, Atal Narayan Sahu, Marco Canini, and Peter Richtárik. Natural compression for distributed deep learning. *arXiv preprint arXiv:1905.10988*, 2019.

[14] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv preprint arXiv:1904.05115*, 2019.

[15] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik. Stochastic distributed learning with gradient quantization and variance reduction, 2019.

[16] Martin Jaggi, Virginia Smith, Martin Takác, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan. Communication-efficient distributed dual coordinate ascent. *Advances in neural information processing systems*, 27, 2014.

[17] Tim Kraska, Ameet Talwalkar, John C Duchi, Rean Griffith, Michael J Franklin, and Michael I Jordan. Mlbase: A distributed machine-learning system. In *CIDR*, volume 1, pages 2–1, 2013.

[18] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020. ISSN 0360-8352. doi: https://doi.org/10.1016/j.cie.2020.106854. URL https://www.sciencedirect.com/science/article/pii/S0360835220305532.

[19] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 583–598, 2014.

[20] Chenxin Ma, Jakub Konečnỳ, Martin Jaggi, Virginia Smith, Michael I Jordan, Peter Richtárik, and Martin Takáč. Distributed optimization with arbitrary local solvers. *optimization Methods and Software*, 32(4):813–848, 2017.

[21] Jakub Marecek, Peter Richtárik, and Martin Takac. Distributed block coordinate descent for minimizing partially separable functions. *Numerical Analysis and Optimization 2014, Springer Proceedings in Mathematics and Statistics*, 2014.

[22] 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. URL http://arxiv.org/abs/1602.05629.

[23] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[24] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

[25] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, , and Peter Richtárik. Distributed learning with compressed gradient differences, 2019.

[26] Hossein K Mousavi, Mohammadreza Nazari, Martin Takáč, and Nader Motee. Multi-agent image classification via reinforcement learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5020–5027. IEEE, 2019.

[27] Angelia Nedić, Alex Olshevsky, and César A Uribe. Fast convergence rates for distributed non-bayesian learning. *IEEE Transactions on Automatic Control*, 62(11):5538–5553, 2017.

[28] Lam Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *In 34th International Conference on Machine Learning, ICML 2017*, 2017.

[29] Lam M Nguyen, Katya Scheinberg, and Martin Takáč. Inexact sarah algorithm for stochastic optimization. *Optimization Methods and Software*, 36(1):237–258, 2021.

[30] Santiago Paternain, Aryan Mokhtari, and Alejandro Ribeiro. A newton-based method for nonconvex optimization with fast evasion of saddle points. *SIAM Journal on Optimization*, 29 (1):343–368, 2019.

[31] M.G. Rabbat and R.D. Nowak. Decentralized source localization and tracking wireless sensor networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 921–924, 2004.

[32] Sundhar Srinivasan Ram, Venugopal V Veeravalli, and Angelia Nedic. Distributed non-autonomous power control through distributed convex optimization. In *IEEE INFOCOM 2009*, pages 3001–3005. IEEE, 2009.

[33] Peter Richtárik and Martin Takác. Distributed coordinate descent method for learning with big data. *Journal of Machine Learning Research*, 17:1–25, 2016.

[34] Mher Safaryan, Rustem Islamov, Xun Qian, and Peter Richtárik. Fednl: Making newton-type methods applicable to federated learning. *arXiv preprint arXiv:2106.02969*, 2021.

[35] Zheng Shi, Nicolas Loizou, Peter Richtárik, and Martin Takáč. Ai-sarah: Adaptive and implicit stochastic recursive gradient methods. *arXiv preprint arXiv:2102.09700*, 2021.

[36] Virginia Smith, Simone Forte, Ma Chenxin, Martin Takáč, Michael I Jordan, and Martin Jaggi. Cocoa: A general framework for communication-efficient distributed optimization. *Journal of Machine Learning Research*, 18:230, 2018.

[37] César A. Uribe, Darina Dvinskikh, Pavel Dvurechensky, Alexander Gasnikov, and Angelia Nedić. Distributed computation of Wasserstein barycenters over networks. In *2018 IEEE 57th Annual Conference on Decision and Control (CDC)*, 2018. Accepted, arXiv:1803.02933.

[38] Lin Xiao and Stephen Boyd. Optimal scaling of a gradient method for distributed resource allocation. *Journal of Optimization Theory and Applications*, 129(3):469–488, 2006.

[39] Yuchen Zhang and Lin Xiao. *Communication-Efficient Distributed Optimization of Self-concordant Empirical Loss*, pages 289–341. Springer International Publishing, Cham, 2018. ISBN 978-3-319-97478-1. doi: 10.1007/978-3-319-97478-1_11.

[40] Xun Qian Zhize Li, Dmitry Kovalev and Peter Richtárik. Acceleration for compressed gradient descent in distributed and federated optimization. *International Conference on Machine Learning*, 37, 2020.

Algorithm 2 Truncated L-SR1 update[2]

Require: $\widetilde{Y}_k^i \in \mathbb{R}^{d\times m},\ M_k^i \in \mathbb{R}^{m\times m},\ B_k^i \in \mathbb{R}^{d\times d},\ S_k \in \mathbb{R}^{d\times m}$ for $i = 1,\dots,n,\ \ \omega > 0$ – truncation constant.

On the server:
for $i = 1,\dots,n$ **do**
  compute $(M_k^i - (S_k^i)^T\widetilde{Y}_k^i) = U_k^i L_k^i (U_k^i)^T$;
  truncate $(L_k^i)^{-1}$ to form $[(L_k^i)^{-1}]_\omega$;
  compute B_{k+1}^i via

$$B_{k+1} = B_k + (\widetilde{Y}_k - B_k^i S_k)U_k^i[(L_k^i)^{-1}]_\omega (U_k^i)^T(\widetilde{Y}_k - B_k^i S_k)^T. \tag{6}$$

end for

Appendix A. FLECS-CGD

A.1. Hessian Approximation Update

Algorithm 3 Direct update [2]

Require: $\widetilde{Y}_k^i \in \mathbb{R}^{d\times m},\ M_k^i \in \mathbb{R}^{m\times m},\ B_k^i \in \mathbb{R}^{d\times d}\ \forall i$

On the server: $0 < \beta_k \leq 1$ – learning rate.
for $i = 1,\dots,n$ **do**
  compute $\widetilde{B}_k^i = \widetilde{Y}_k^i (M_k^i)^\dagger (\widetilde{Y}_k^i)^T$;
  select learning rate β_k
  compute $B_{k+1}^i = (1-\beta_k)B_k^i + \beta_k \widetilde{B}_k^i$.
end for

A.2. Iterate update

Definition 7 *Let B_k, V_k, Λ_k be matrices such that $B_k = V_k\Lambda_k V_k^T$, and let $0 < \omega \leq \Omega$. The truncated inverse Hessian approximation of B_k is $\left(|B_k|_\omega^\Omega\right)^{-1} := V_k(|\Lambda_k|_\omega^\Omega)^{-1}V_k^T$, where* $(|\Lambda_k|_\omega^\Omega)_{ii} = \min\left\{\max\left\{|\Lambda|_{ii}, \omega\right\}, \Omega\right\}.$

Definition 7 was proposed in [30] and was used to provide a convergence guarantee for their Nonconvex Newton method (to a local minimum). Firstly, an eigen-decomposition of B_k is computed, but with every eigenvalue replaced by its absolute value. Secondly, a thresholding step is applied, so that any eigenvalue (in absolute value) that is smaller (resp. greater) than a user defined threshold ω (resp. Ω) is replaced by ω (resp. Ω).

Appendix B. Additional Experiments

Comparison between FLECS and FLECS-CGD is provided on Figure 2.
Comparison between FLECS-CGD 's iterate updates (Algorithms 4, 5) is provided on Figure 3.

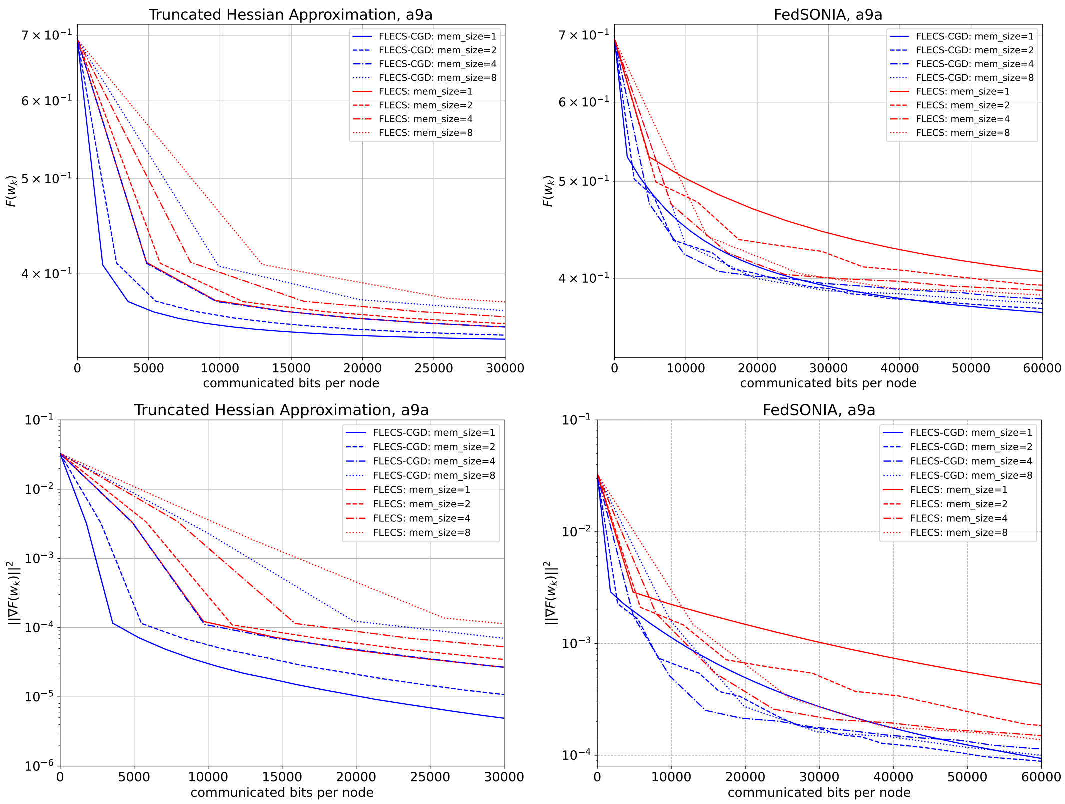

Figure 2: Comparison of objective function $F(w_k)$ and the squared norm of gradient $\|\nabla F(w_k)\|^2$ for FLECS and FLECS-CGD .

Algorithm 4 Truncated inverse Hessian approximation [2]

Require: $\nabla F(w_k) \in \mathbb{R}^d,\ \widetilde{Y}_k \in \mathbb{R}^{d\times m},\ M_k \in \mathbb{R}^{m\times m},\ B_{k+1} \in \mathbb{R}^{d\times d}$, $\Omega > \omega > 0$ – truncation constants.

1: **On the server:**
2: compute spectral decomposition $B_{k+1} = V_k \Lambda_k V_k^T$;
3: truncate Λ_k to form $|\Lambda_k|_\omega^\Omega$ via Definition 7;
4: compute search direction p_k via $p_k = \left(|B_{k+1}|_\omega^\Omega\right)^{-1} \nabla F(w_k)$;
5: **return** p_k.

Algorithm 5 FedSONIA [2]

Require: $\nabla F(w_k) \in \mathbb{R}^d,\ \widetilde{Y}_k \in \mathbb{R}^{d\times m},\ M_k \in \mathbb{R}^{m\times m}$, $\Omega > \omega > 0$ – truncation constants.

1: **On the server:**
2: compute $B_k^{\text{SONIA}} := \widetilde{Y}_k (M_k)^\dagger \widetilde{Y}_k^T$;
3: compute QR factorization of $\widetilde{Y}_k (= Q_k R_k)$;
4: compute spectral decomposition of $R_k (M_k)^\dagger R_k^T (= V_k \Lambda_k V_k^T)$;
5: construct $\widetilde{V}_k := Q_k V_k$;
6: truncate Λ_k to form $|\Lambda_k|_\omega^\Omega$ via Definition 7;
7: Set ρ_k and decompose gradient via $\nabla F(w_k) = g_k + g_k^\perp$;
8: Compute search direction p_k via $p_k := -\left(|B_{k+1}^{\text{SONIA}}|_\omega^\Omega\right)^{-1} g_k - \rho_k g_k^\perp, ;$
9: **return** p_k.

Appendix C. Proofs

C.1. Basic identities and propertites

Let $x, y \in \mathbb{R}^d$ and $\alpha \in [0, 1]$:

$$\|\alpha x + (1-\alpha) y\|_2^2 = \alpha \|x\|_2^2 + (1-\alpha)\|y\|_2^2 - \alpha(1-\alpha)\|x - y\|_2^2 \tag{7}$$

Let g be a random vector, and $h \in \mathbb{R}^d$:

$$\mathbb{E}\left[\|g - \mathbb{E}[g]\|_2^2\right] = \mathbb{E}\left[\|g - h\|_2^2\right] - \|\mathbb{E}[g] - h\|_2^2 \tag{8}$$

For any independent random variables $X_1, X_2, \ldots, X_n \in \mathbb{R}^d$

$$\mathbb{E}\left[\|\frac{1}{n}\sum_{i=1}^n (X_i - \mathbb{E}[X_i])\|^2\right] = \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|X_i - \mathbb{E}[X_i]\|^2\right] \tag{9}$$

Lemma 8 *[2] The search direction p_k in FLECS is equivalent to $p_k = -\mathcal{A}_k \widetilde{g}_k$*

Lemma 9 *[2] If Assumption 1 holds, there exist constants $0 < \mu_1 \le \mu_2$ such that the inverse truncated Hessian approximations $\{\mathcal{A}_k\}$ generated by FLECS satisfy*

$$\mu_1 I \preceq \mathcal{A}_k \preceq \mu_2 I, \qquad \textit{for all } k = 0, 1, \ldots \tag{10}$$

for some constants $\mu_1,\ \mu_2$.

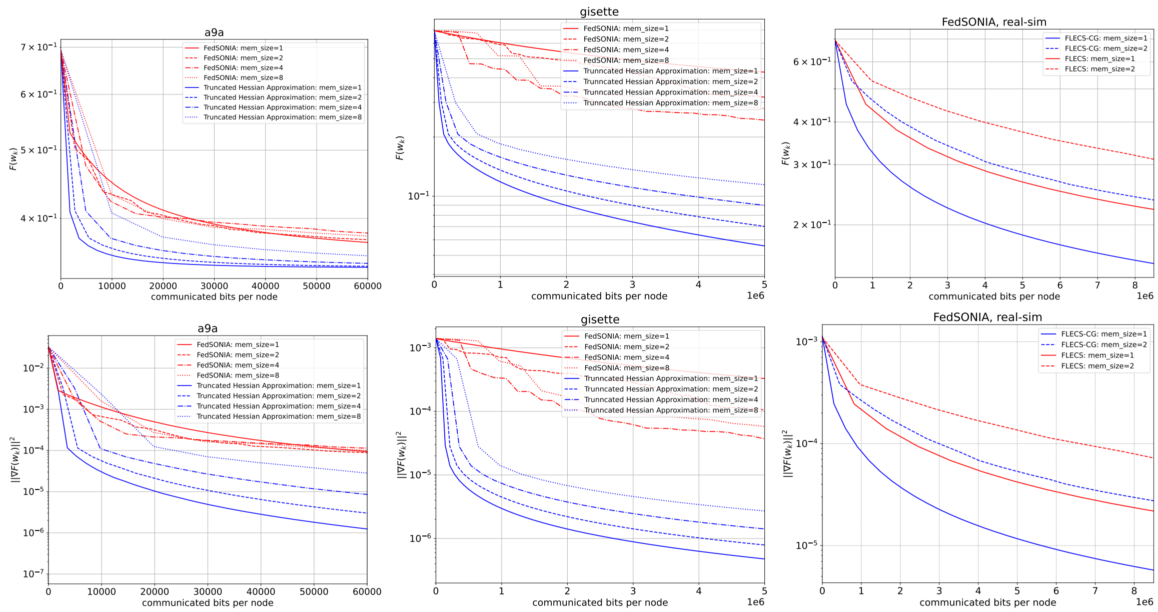

Figure 3: Comparison of objective function $F(w_k)$ and the squared norm of gradient $\|\nabla F(w_k)\|^2$ for different iterate updates in FLECS-CGD .

C.2. Proof of Theorem 4

Assumption 1 *The function F is twice continuously differentiable.*

Assumption 2 *Each function $f_i(w)$ is μ-strongly convex and L-smooth*

$$\mu I \preceq \nabla^2 f_i(w) \preceq LI. \tag{11}$$

Assumption 3 *Each g_k^i in Algorithm 1 has bounded variance*

$$\mathbb{E}\left[\|g_k^i - \nabla f_i(w_k)\|\right] \le \sigma_i^2, \quad \forall k \ge 0,\ i = 1, \dots, n \tag{12}$$

for constants $\sigma_i < \infty$, $\sigma^2 := \frac{1}{n}\sum\limits_{i=1}^{n} \sigma_i^2$.

Lemma 10 *[14] For all iterations $k \ge 0$ of Algorithm 1 we have:*

$$\mathbb{E}\left[\widetilde{g}_k\right] = g_k := \frac{1}{n}\sum_{i=1}^{n} g_k^i, \quad \mathbb{E}_Q\left[\|\widetilde{g}_k - \nabla F(w_k)\|^2\right] \le \frac{\omega}{n^2}\sum_{i=1}^{n}\|\nabla f_i(w_k) - h_k^i\|^2, \quad \mathbb{E}\left[g_k\right] = \nabla f(w_k); \tag{13}$$

and

$$\mathbb{E}\left[\|\widetilde{g}_k - h_*^i\|^2\right] \le \frac{2\omega}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] + \left(\frac{2\omega}{n} + 1\right)\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|\nabla f_i(w_k) - h_*^i\|^2\right] + (1+\omega)\frac{\sigma^2}{n}, \tag{14}$$

where $h_*^i = \nabla f_i(w^*)$.

Proof We prove the first equality in (13):

$$\mathbb{E}_Q\left[\widetilde{g}_k\right] = \mathbb{E}_Q\left[\frac{1}{n}\sum_{i=1}^{n}\widetilde{g}_k^i\right] = \mathbb{E}_Q\left[\frac{1}{n}\sum_{i=1}^{n}(Q(g_k^i - h_k^i) + h_k^i)\right] \overset{(2)}{=} \frac{1}{n}\sum_{i=1}^{n} g_k^i = g_k.$$

Now, we prove the second inequality in (13):

$$\begin{aligned}\mathbb{E}_Q\left[\|\widetilde{g}_k - g_k\|^2\right] &= \mathbb{E}_Q\left[\|\frac{1}{n}\sum_{i=1}^{n}(\widetilde{g}_k^i - \nabla f_i(w_k))\|^2\right] \\ &= \mathbb{E}_Q\left[\|\frac{1}{n}\sum_{i=1}^{n}(Q(g_k^i - h_k^i) - (g_k^i - h_k^i))\|^2\right] \\ &\overset{(2),(9)}{\leq} \mathbb{E}_Q\left[\frac{1}{n^2}\sum_{i=1}^{n}\omega\|g_k^i - h_k^i\|^2\right]. \end{aligned} \tag{15}$$

The last equality in (13) follows follows from the assumption that each g_k^i is an unbiased estimate of $\nabla f_i(w_k)$. Let $h_* = \nabla F(w^*) = 0$

$$\begin{aligned}\mathbb{E}\left[\|\widetilde{g}_k - h_*\|^2\right] &\overset{(8)}{=} \mathbb{E}\left[\|\widetilde{g}_k - g_k\|^2\right] + \mathbb{E}\left[\|g_k - h_*\|^2\right] \overset{(8)}{=} \\ \mathbb{E}\left[\|\widetilde{g}_k - g_k\|^2\right] &+ \mathbb{E}\left[\|g_k - \nabla F(w_k)\|^2\right] + \mathbb{E}\left[\|\nabla F(w_k) - h_*\|^2\right]\end{aligned} \tag{16}$$

Then,

$$\mathbb{E}_Q\left[\|\widetilde{g}_k - g_k\|^2\right] \overset{(15)}{\leq} \frac{\omega}{n^2}\sum_{i=1}^{n}\mathbb{E}_Q\left[\|g_k^i - h_k^i\|^2\right]. \tag{17}$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[\|\widetilde{g}_k - h_*\|^2\right] &\overset{(16),(17)}{\leq} \frac{w}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\|g_k^i - h_k^i\|^2\right] + \mathbb{E}\left[\|g_k - \nabla F(w_k)\|^2\right] + \mathbb{E}\left[\|\nabla F(w_k) - h_*\|^2\right] \\ &\leq \frac{w}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\|g_k^i - h_k^i\|^2\right] + \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|g_k^i - h_*^i\|^2\right] + \frac{\sigma^2}{n},\end{aligned} \tag{18}$$

where the last inequality is valid due to Jensen inequality. Next,

$$\begin{aligned}\mathbb{E}\left[\|g_k^i - h_k^i\|^2\right] &\leq \mathbb{E}\left[\|\nabla f_i(w_k) - h_k^i\|^2\right] + \mathbb{E}\left[\|\nabla f_i(w_k) - g_k^i\|^2\right] \\ &\leq \mathbb{E}\left[\|(\nabla f_i(w_k) - h_*^i) + (h_*^i - h_k^i)\|^2\right] + \sigma_i^2 \\ &\overset{\|\sum_{i=1}^{t} a_i\|^2 \leq t\sum_{i=1}^{t}\|a_i\|^2}{\leq} 2\mathbb{E}\left[\|\nabla f_i(w_k) - h_*^i\|^2\right] + 2\mathbb{E}\left[\|\nabla f_i(w^*) - h_k^i\|^2\right] + \sigma_i^2.\end{aligned} \tag{19}$$

Therefore,

$$\mathbb{E}\left[\|\widetilde{g}_k - h_*\|^2\right] \overset{(18),(19)}{\leq} \frac{2\omega}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\|\nabla f_i(w_k) - h_*^i\|^2 + \|h_*^i - h_k^i\|^2\right] + \frac{\omega}{n^2}\sigma_i^2$$

$$+\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|\nabla f_i(w_k) - h_*^i\|^2\right] + \frac{\sigma^2}{n}$$

$$\leq \frac{2\omega}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\|h_*^i - h_k^i\|^2\right] + \left(\frac{2\omega}{n}+1\right)\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|\nabla f_i(w_k) - h_*^i\|^2\right] + (\omega+1)\frac{\sigma^2}{n}.$$

■

Lemma 11 *[14] Let $\gamma_k(\omega+1) \leq 1$ for any $k \geq 0$. Then for all iterations $k \geq 0$ of Algorithm 1 and all workers $i = 1 \dots n$ we have:*

$$\mathbb{E}_Q\left[\|h_{k+1}^i - h_*^i\|^2\right] \leq (1-\gamma_k)\|h_k^i - h_*^i\|^2 + \gamma_k\|\nabla f_i(w_k) - h_*^i\|^2 + \gamma_k\sigma_i^2. \tag{20}$$

Proof Since $h_{k+1}^i = h_k^i + \gamma_k Q(g_k^i - h_k^i)$

$$\mathbb{E}_Q\left[\|h_{k+1}^i - h_*^i\|^2\right] = \mathbb{E}_Q\left[\|\gamma_k Q(g_k^i - h_k^i) + (h_k^i - h_*^i)\|^2\right]$$

$$= \|h_k^i - h_*^i\|^2 + 2\mathbb{E}_Q\left[\langle\gamma_k Q(g_k^i - h_k^i), h_k^i - h_*^i\rangle\right] + \mathbb{E}_Q\left[\|\gamma_k Q(g_k^i - h_k^i)\|^2\right]$$

$$\overset{(2)}{\leq} \|h_k^i - h_*^i\|^2 + 2\gamma_k\langle g_k^i - h_k^i, h_k^i - h_*^i\rangle + \gamma_k^2(\omega+1)\|g_k^i - h_k^i\|^2$$

$$\overset{\gamma_k(\omega+1)\leq 1}{\leq} \|h_k^i - h_*^i\|^2 + 2\gamma_k\langle g_k^i - h_k^i, h_k^i - h_*^i\rangle + \gamma_k\|g_k^i - h_k^i\|^2$$

$$= \|h_k^i - h_*^i\|^2 + 2\gamma_k\langle g_k^i - h_k^i, h_k^i - h_*^i\rangle + \gamma_k\langle g_k^i - h_k^i, g_k^i - h_k^i\rangle$$

$$= \|h_k^i - h_*^i\|^2 + \gamma_k\langle g_k^i - h_k^i, 2h_k^i - 2h_*^i + g_k^i - h_k^i\rangle$$

$$= \|h_k^i - h_*^i\|^2 + \gamma_k\langle g_k^i - h_k^i, h_k^i + g_k^i - 2h_*^i\rangle$$

$$= \|h_k^i - h_*^i\|^2 + \gamma_k\|g_k^i - h_*^i\|^2 - \gamma_k\|h_*^i - h_k^i\|^2$$

$$= (1-\gamma_k)\|h_k^i - h_*^i\|^2 + \gamma_k\|g_k^i - h_*^i\|^2$$

$$\leq (1-\gamma_k)\|h_k^i - h_*^i\|^2 + \gamma_k\|\nabla f_i(w_k) - h_*^i\|^2 + \gamma_k\sigma_i^2,$$

where in the last equality is due to fact that for any vectors a, b we have $\|a-b\|^2 = \langle a-b, a+b\rangle$.

■

Theorem 4 *Suppose that Assumption 1, 2, 3 holds. Let $Q \in \mathcal{U}(\omega)$. Let $\{w_k\}$ be the iterates generated by Algorithm 1, where $0 < \alpha_k = \alpha \leq \frac{5\mu\mu_1}{2L^2\mu_2^2\left(1+\frac{\omega}{n}\right)}$ and $0 < \gamma_k = \gamma \leq \frac{1}{\omega+1}$. Define the Lyapunov function*

$$\Psi_{k+1} = (F(w_{k+1}) - F(w_*)) + \frac{cL\mu_2^2\alpha^2}{n}\sum_{i=1}^{n}\mathbb{E}_Q\left[\|h_{k+1}^i - h_*^i\|^2\right]$$

for $0 < c = \min\left\{\frac{1-\frac{\alpha\mu\mu_1}{2}-\frac{\omega}{n}}{1-\gamma}; \frac{\mu}{2\gamma L}\right\}$. Then for all $k \geq 0$:

$$\mathbb{E}_Q\left[\Psi_k\right] \leq \left(1 - \frac{\alpha\mu\mu_1}{2}\right)^{k+1}\Psi_0 + \left(\frac{\omega+1}{2n} + \gamma c\right)\frac{2L\mu_2^2\alpha}{\mu\mu_1}\sigma^2. \tag{21}$$

Proof

$$\mathbb{E}\left[F(w_{k+1})\right] \leq \mathbb{E}\left[F(w_k) - \nabla F(w_k)^T(-\alpha A_k \widetilde{g}_k) + \frac{L}{2}\|\alpha A_k \widetilde{g}_k\|^2\right]$$

$$= \mathbb{E}\left[F(w_k)\right] - \alpha \mathbb{E}\left[\nabla F(w_k)^T A_k \nabla F(w_k)\right] + \frac{L\mu_2^2\alpha^2}{2}\mathbb{E}\left[\|\widetilde{g}_k\|^2\right]$$

$$\overset{\text{Lem. 9}}{\leq} \mathbb{E}\left[F(w_k)\right] - \alpha\mu_1 \mathbb{E}\left[\|\nabla F(w_k)\|^2\right] + \frac{L\mu_2^2\alpha^2}{2}\mathbb{E}\left[\|\widetilde{g}_k - h_*\|^2\right]$$

$$\overset{(14)}{\leq} \mathbb{E}\left[F(w_k)\right] - \alpha\mu_1 \mathbb{E}\left[\|\nabla F(w_k)\|^2\right]$$

$$+\frac{L\mu_2^2\alpha^2}{2}\left(\frac{2\omega}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] + \left(\frac{2\omega}{n}+1\right)\sum_{i=1}^n \mathbb{E}\left[\|\nabla f_i(w_k) - h_*^i\|^2\right] + (\omega+1)\frac{\sigma^2}{n}\right)$$

$$\leq \mathbb{E}\left[F(w_k)\right] - \alpha\mu_1 \mathbb{E}\left[\|\nabla F(w_k)\|^2\right]$$

$$+\frac{L\alpha^2\mu_2^2}{2}\mathbb{E}\left[\|\nabla F(w_k)\|^2\right] - \frac{L\alpha^2\mu_2^2}{2}\mathbb{E}\left[\|\nabla F(w_k)\|^2\right]$$

$$+\frac{L\mu_2^2\alpha^2}{2}\left(\frac{2\omega}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] + \left(\frac{2\omega}{n}+1\right)\sum_{i=1}^n \mathbb{E}\left[\|\nabla f_i(w_k) - h_*^i\|^2\right] + (\omega+1)\frac{\sigma^2}{n}\right).$$

By strong convexity of F we have $2\mu(F(w_k) - F(w_*)) \leq \|F(w_k)\|^2$. By L-Lipschitz continuity of each f_i we have $f_i(w_*) + \langle \nabla f_i(w_*), w_k - w_* \rangle + \frac{1}{2L}\|\nabla f_i(w_k) - \nabla f_i(w_*)\|^2 \leq f_i(w_k)$. Therefore,

$$\mathbb{E}\left[F(w_k) - F(w_*)\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[(f_i(w_k) - f_i(w_*))\right]$$

$$\geq \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\left(\langle \nabla f_i(w_*), w_k - w_* \rangle + \frac{1}{2L}\|\nabla f_i(w_k) - \nabla f_i(w_*)\|^2\right)\right]$$

$$= \mathbb{E}\left[\left(\langle \frac{1}{n}\sum_{i=1}^n \nabla f_i(w_*), w_k - w_* \rangle + \frac{1}{2L}\frac{1}{n}\sum_{i=1}^n \|\nabla f_i(w_k) - \nabla f_i(w_*)\|^2\right)\right]$$

$$= \frac{1}{2L}\frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[\|\nabla f_i(w_k) - \nabla f_i(w_*)\|^2\right].$$

Then,

$$\mathbb{E}\left[F(w_{k+1})\right] \leq \mathbb{E}\left[F(w_k)\right] - 2\alpha\mu\mu_1\mathbb{E}\left[(F(w_k) - F(w_*))\right] + L^2\alpha^2\mu_2^2\mathbb{E}\left[(F(w_k) - F(w_*))\right]$$
$$-\frac{L\alpha^2\mu_2^2}{2}\mathbb{E}\left[\|\nabla F(w_k)\|^2\right] + \frac{L\mu_2^2\alpha^2}{2}\left(\frac{2\omega}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] + \left(\frac{2\omega}{n}+1\right)2L\mathbb{E}\left[(F(w_k) - F(w_*))\right] + (\omega+1)\frac{\sigma^2}{n}\right)$$
$$= \mathbb{E}\left[F(w_k)\right] - (2\alpha\mu\mu_1 - 2L^2\mu_2^2\alpha^2(\omega+1))\mathbb{E}\left[(F(w_k) - F(w_*))\right] - \frac{L\alpha^2\mu_2^2}{4}\mathbb{E}\left[\|\nabla F(w_k)\|^2\right]$$
$$+\frac{L\mu_2^2\alpha^2\omega}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] + (\omega+1)\frac{L\mu_2^2\alpha^2\sigma^2}{2n}$$
$$\leq \mathbb{E}\left[F(w_k)\right] - \frac{\alpha\mu\mu_1}{2}\mathbb{E}\left[(F(w_k) - F(w_*))\right] - \frac{L\alpha^2\mu_2^2}{4}\mathbb{E}\left[\|\nabla F(w_k)\|^2\right]$$
$$+\frac{L\mu_2^2\alpha^2\omega}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] + (\omega+1)\frac{L\mu_2^2\alpha^2\sigma^2}{2n}$$

Where the last inequality holds due to the choice of learning rate $0 < \alpha \leq \frac{5\mu\mu_1}{2L^2\mu_2^2\left(1+\frac{\omega}{n}\right)}$.
By subtracting $F(w_*)$ from the LHS and the RHS, we have

$$\mathbb{E}\left[F(w_{k+1}) - F(w_*)\right] \leq \left(1 - \frac{\alpha\mu\mu_1}{2}\right)\mathbb{E}\left[(F(w_k) - F(w_*))\right] - \frac{L\alpha^2\mu_2^2}{4}\mathbb{E}\left[\|\nabla F(w_k)\|^2\right] \tag{22}$$
$$+\frac{L\mu_2^2\alpha^2\omega}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] + (\omega+1)\frac{L\mu_2^2\alpha^2\sigma^2}{2n}. \tag{23}$$

Let us define Lyapunov function Ψ_{k+1} as

$$\Psi_{k+1} = \mathbb{E}\left[(F(w_{k+1}) - F(w_*))\right] + \frac{cL\mu_2^2\alpha^2}{n}\sum_{i=1}^n \mathbb{E}\left[\|h_{k+1}^i - h_*^i\|^2\right]. \tag{24}$$

Then

$$\Psi_{k+1} \overset{(23)}{\leq} \left(1 - \frac{\alpha\mu\mu_1}{2}\right)\mathbb{E}\left[(F(w_k) - F(w_*))\right] - \frac{L\alpha^2\mu_2^2}{4}\mathbb{E}\left[\|\nabla F(w_k)\|^2\right]$$
$$+\frac{L\mu_2^2\alpha^2\omega}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] + \frac{cL\mu_2^2\alpha^2}{n}\sum_{i=1}^n \mathbb{E}\left[\|h_{k+1}^i - h_*^i\|^2\right] + (\omega+1)\frac{L\mu_2^2\alpha^2\sigma^2}{2n}$$
$$\overset{(20)}{\leq} \left(1 - \frac{\alpha\mu\mu_1}{2}\right)\mathbb{E}\left[(F(w_k) - F(w_*))\right] - \frac{L\alpha^2\mu_2^2}{4}\mathbb{E}\left[\|\nabla F(w_k)\|^2\right] + \frac{L\mu_2^2\alpha^2\omega}{n^2}\sum_{i=1}^n \mathbb{E}\left[\|h_k^i - h_*^i\|^2\right]$$
$$+\frac{cL\mu_2^2\alpha^2}{n}\sum_{i=1}^n\left((1-\gamma)\mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] + (\omega+1)\frac{L\mu_2^2\alpha^2\sigma^2}{2n} + \gamma\mathbb{E}\left[\|\nabla f_i(w_k) - h_*^i\|^2\right] + \gamma\sigma_i^2\right) \tag{25}$$

Then by L-Lipschitz continuity of each f_i and μ strong convexity of F we have

$$
\begin{aligned}
&-\frac{L\alpha^2\mu_2^2}{4}\mathbb{E}\left[\|\nabla F(w_k)\|^2\right] + \frac{cL\mu_2^2\alpha^2\gamma}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|\nabla f_i(w_k) - h_*^i\|^2\right] \\
&\le -\frac{\mu L\alpha^2\mu_2^2}{2}\mathbb{E}\left[(F(w_k) - F(w_*))\right] + 2c\gamma L^2\alpha^2\mu_2^2\mathbb{E}\left[(F(w_k) - F(w_*))\right] \\
&\le \left(2c\gamma L^2\alpha^2\mu_2^2 - \frac{\mu L\alpha^2\mu_2^2}{2}\right)\mathbb{E}\left[(F(w_k) - F(w_*))\right] \le 0, \qquad (26)
\end{aligned}
$$

where the last inequality is due to the choice of c and γ as $\gamma \le \frac{\mu}{2cL}$.

By assumption on c, (25) and (26) we have

$$
\begin{aligned}
\Psi_{k+1} &\le \left(1 - \frac{\alpha\mu\mu_1}{2}\right)\mathbb{E}\left[(F(w_k) - F(w_*))\right] + (1-\gamma)\frac{cL\mu_2^2\alpha^2}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] \\
&\quad + \frac{L\mu_2^2\alpha^2\omega}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] + \left(\frac{\omega+1}{2n} + \gamma c\right)L\mu_2^2\alpha^2\sigma^2 \\
&\le \left(1 - \frac{\alpha\mu\mu_1}{2}\right)\mathbb{E}\left[(F(w_k) - F(w_*))\right] + \left(1 - \frac{\alpha\mu\mu_1}{2}\right)\frac{cL\mu_2^2\alpha^2}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|h_k^i - h_*^i\|^2\right] + \left(\frac{\omega+1}{2n} + \gamma c\right)L\mu_2^2\alpha^2\sigma^2.
\end{aligned}
$$

Finally,

$$
\begin{aligned}
\Psi_{k+1} &\le \left(1 - \frac{\alpha\mu\mu_1}{2}\right)\Psi_k + \left(\frac{\omega+1}{2n} + \gamma c\right)L\mu_2^2\alpha^2\sigma^2 \\
&\le \left(1 - \frac{\alpha\mu\mu_1}{2}\right)^{k+1}\Psi_0 + \left(\frac{\omega+1}{2n} + \gamma c\right)L\mu_2^2\alpha^2\sigma^2\sum_{t=0}k\left(1 - \frac{\alpha\mu\mu_1}{2}\right)^t \\
&\le \left(1 - \frac{\alpha\mu\mu_1}{2}\right)^{k+1}\Psi_0 + \left(\frac{\omega+1}{2n} + \gamma c\right)\frac{2L\mu_2^2\alpha}{\mu\mu_1}\sigma^2,
\end{aligned}
$$

where the last inequality is due to estimate $\sum_{t=0}^{k}(1 - \frac{\alpha\mu\mu_1}{2})^t \le \frac{2}{\alpha\mu\mu_1}$. ■

C.3. Proof of Theorem 5

Lemma 12 *[25] Let $x^* \in X^*$, such that X^* is the set of solutions for (1), and define $h_*^i = \nabla f_i(x^*)$, we have for each worker $i \in [n]$, the first and second moments of h_{k+1}^i are equal to:*

$$
\mathbb{E}_Q\left[h_{k+1}^i\right] = (1-\gamma_k)h_k^i + \gamma_k g_k^i \qquad (27)
$$

$$
\mathbb{E}_Q\left[\|h_{k+1}^i - h_*^i\|_2^2\right] \le (1-\gamma_k)\|h_k^i - h_*^i\|_2^2 + \gamma_k\|g_k^i - h_*^i\|_2^2 + \left(\gamma_k^2\omega - \gamma_k(1-\gamma_k)\right)\|g_k^i - h_k^i\|_2^2 \qquad (28)
$$

Proof Since $h_{k+1}^i = h_k^i + \gamma_k c_k^i$

$$
\begin{aligned}
\mathbb{E}_Q\left[h_{k+1}^i\right] &= h_k^i + \gamma_k\mathbb{E}_Q\left[c_i^k\right] \\
&= h_k^i + \gamma_k(g_k^i - h_k^i).
\end{aligned}
$$

Secondly:

$$
\begin{aligned}
\mathbb{E}_Q\left[\|h_{k+1}^i - h_*^i\|_2^2\right] &\overset{(8)}{=} \left\|\mathbb{E}_Q\left[h_{k+1}^i\right] - h_*^i\right\|_2^2 + \mathbb{E}_Q\left[\left\|h_{k+1}^i - \mathbb{E}_Q\left[h_{k+1}^i\right]\right\|_2^2\right] \\
&\overset{(27)}{=} \left\|(1-\gamma_k)h_k^i + \gamma_k g_k^i - h_*^i\right\|_2^2 + \gamma_k^2 \mathbb{E}_Q\left[\left\|c_i^k - \mathbb{E}_Q\left[c_i^k\right]\right\|_2^2\right] \\
&= \left\|(1-\gamma_k)\left[h_k^i - h_*^i\right] + \gamma_k\left[g_k^i - h_*^i\right]\right\|_2^2 + \gamma_k^2 \mathbb{E}_Q\left[\left\|c_i^k - \mathbb{E}_Q\left[c_i^k\right]\right\|_2^2\right] \\
&\overset{(2)}{\leq} \left\|(1-\gamma_k)\left[h_k^i - h_*^i\right] + \gamma_k\left[g_k^i - h_*^i\right]\right\|_2^2 + \gamma_k^2 \omega \|g_k^i - h_k^i\|_2^2 \\
&\overset{(7)}{=} (1-\gamma_k)\|h_k^i - h_*^i\|_2^2 + \gamma_k\|g_k^i - h_*^i\|_2^2 - \gamma_k(1-\gamma_k)\|g_k^i - h_k^i\|_2^2 \\
&+ \gamma_k^2 \omega \|g_k^i - h_k^i\|_2^2 \\
&= (1-\gamma_k)\|h_k^i - h_*^i\|_2^2 + \gamma_k\|g_k^i - h_*^i\|_2^2 + \left(\gamma_k^2\omega - \gamma_k(1-\gamma_k)\right)\|g_k^i - h_k^i\|_2^2.
\end{aligned}
$$

■

Assumption 4 *The function F is L-smooth.*

Assumption 5 *(Bounded data dissimilarity). There exists constant $\zeta \geq 0$ such that $\forall x \in \mathbb{R}^d$*

$$
\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x) - \nabla F(x)\|_2^2 \leq \zeta^2 \tag{29}
$$

In particular, $\zeta = 0$, implies that all datasets stored in the n devices are drawn from the same data distribution $\mathcal{D}$.

Theorem 5 *Let $S = \{w_0, w_1, \ldots, w_{k-1}\}$ be generated using Algorithm 1, and $\bar{w}$ be sampled uniformly at random from S, for $\alpha \leq \sqrt{\frac{n}{2Lw(w+1)\mu_2^2}}$ and $\gamma_k \leq \frac{1+\sqrt{1-\frac{2L\alpha^2 w(w+1)\mu_2^2}{n}}}{2(w+1)}$, and a parameter c such as $c < \frac{\mu_1}{L\alpha\gamma_k} - \frac{\mu_2^2}{2\gamma_k}$ we have:*

$$
\begin{aligned}
\mathbb{E}_Q\left[\|\nabla F(\bar{w})\|_2^2\right] \leq 2\frac{\bar{\kappa}^0}{k\alpha\left(2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k\right)} + \frac{4cL\alpha}{2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k}\zeta^2 \\
+ \frac{\mu_2^2 + 2c}{2\mu_1 - \left(L\alpha\mu_2^2\right) - 2cL\alpha}L\sigma^2
\end{aligned}
\tag{30}
$$

with $\bar{\kappa}^k = F(w_k) - F^* + c\frac{L\alpha^2}{2}\frac{1}{n}\sum_{i=1}^{n}\|h_k^i - h_*^i\|_2^2$

Proof We have $w_{k+1} = w_k - \alpha_k A_k \widetilde{g}_k$, therefore:

$$\begin{aligned}
\mathbb{E}\left[F(w_{k+1})\right] &= \mathbb{E}\left[F(w_k - \alpha_k A_k \widetilde{g}_k)\right] \\
&\overset{(2)}{\leq} \mathbb{E}\left[F(w_k)\right] - \alpha \mathbb{E}\left[\langle \nabla F(w_k), A_k \widetilde{g}_k \rangle\right] + \frac{L\alpha^2}{2} \mathbb{E}\left[\|A_k \widetilde{g}_k\|^2\right] \\
&\overset{(10),(13)}{\leq} \mathbb{E}\left[F(w_k)\right] - \alpha \mu_1 \mathbb{E}\left[\|\nabla F(w_k)\|^2\right] + \frac{L\alpha^2 \mu_2^2}{2} \mathbb{E}\left[\|\widetilde{g}_k\|^2\right] \\
&= \mathbb{E}\left[F(w_k)\right] - \alpha \mu_1 \mathbb{E}\left[\|\nabla F(w_k)\|^2\right] + \frac{L\alpha^2 \mu_2^2}{2} \left[\mathbb{E}\left[\|\widetilde{g}_k - g_k\|^2\right] + \mathbb{E}\left[\|g_k\|^2\right]\right] \\
&\overset{(13)}{\leq} \mathbb{E}\left[F(w_k)\right] - \alpha \mu_1 \mathbb{E}\left[\|\nabla F(w_k)\|^2\right] + \frac{L\alpha^2 \mu_2^2}{2} \left[\frac{\omega}{n^2} \sum_{i=1}^{n} \|g_k^i - h_k^i\|^2 + \mathbb{E}\left[\|\nabla F(w_k)\|^2\right] + \sigma^2\right] \\
&= \mathbb{E}\left[F(w_k)\right] + \alpha \left(\frac{L\alpha \mu_2^2}{2} - \mu_1\right) \mathbb{E}\left[\|\nabla F(w_k)\|^2\right] + \frac{L\alpha^2 \mu_2^2}{2} \frac{\omega}{n^2} \sum_{i=1}^{n} \|g_k^i - h_k^i\|^2 + \frac{L\alpha^2 \mu_2^2}{2} \sigma^2
\end{aligned}$$

Define $\bar{\kappa}^k = F(w_k) - F^* + c\frac{L\alpha^2}{2}\frac{1}{n}\sum_{i=1}^{n} \|h_k^i - h_*^i\|_2^2$, where $h_*^i = \nabla f_i(w^*)$.

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\|h_{k+1}^i - h_*^i\|_2^2 | w_k\right] &\overset{(28)}{\leq} \frac{1}{n}\sum_{i=1}^{n} \left[(1-\gamma_k)\|h_k^i - h_*^i\|_2^2 + \gamma_k \|g_k^i - h_*^i\|_2^2 + \left(\gamma_k^2 \omega - \gamma_k(1-\gamma_k)\right)\|g_k^i - h_k^i\|_2^2\right] \\
&= \frac{1-\gamma_k}{n}\sum_{i=1}^{n} \|h_k^i - h_*^i\|_2^2 + \frac{\gamma_k}{n}\sum_{i=1}^{n} \|g_k^i - h_*^i\|^2 + \frac{\left(\gamma_k^2 \omega - \gamma_k(1-\gamma_k)\right)}{n}\sum_{i=1}^{n} \|g_k^i - h_k^i\|_2^2 \\
&\overset{\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2}{\leq} \frac{1-\gamma_k}{n}\sum_{i=1}^{n} \|h_k^i - h_*^i\|_2^2 + \frac{2\gamma_k}{n}\sum_{i=1}^{n} \|g_k^i\|_2^2 + \\
&\frac{2\gamma_k}{n}\sum_{i=1}^{n} \|h_*^i - \underbrace{\nabla F(w^*)}_{=0}\|_2^2 + \frac{\left(\gamma_k^2 \omega - \gamma_k(1-\gamma_k)\right)}{n}\sum_{i=1}^{n} \|g_k^i - h_k^i\|_2^2 \\
&\leq \frac{1-\gamma_k}{n}\sum_{i=1}^{n} \|h_k^i - h_*^i\|_2^2 + \frac{2\gamma_k}{n}\sum_{i=1}^{n} \|\nabla f_i(w_k)\|_2^2 + \frac{2\gamma_k}{n}\sum_{i=1}^{n} \sigma_i^2 \\
&+ \frac{2\gamma_k}{n}\sum_{i=1}^{n} \|h_*^i - \underbrace{\nabla F(w^*)}_{=0}\|_2^2 + \frac{\left(\gamma_k^2 \omega - \gamma_k(1-\gamma_k)\right)}{n}\sum_{i=1}^{n} \|g_k^i - h_k^i\|_2^2 \\
&\overset{(5)+(8)}{\leq} \frac{1-\gamma_k}{n}\sum_{i=1}^{n} \|h_k^i - h_*^i\|_2^2 + 2\gamma_k \|\nabla F(w_k)\|_2^2 + 4\gamma_k \zeta^2 \\
&+ \frac{\left(\gamma_k^2 \omega - \gamma_k(1-\gamma_k)\right)}{n}\sum_{i=1}^{n} \|g_k^i - h_k^i\|_2^2 + 2\gamma_k \sigma^2
\end{aligned}$$

Therefore:

$$\begin{aligned}
\mathbb{E}\left[\bar{\kappa}^{k+1}\right] &= \mathbb{E}\left[F(w_{k+1})\right] - F^* + c\frac{L\alpha^2}{2}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|h_{k+1}^i - h_*^i\|_2^2\right] \\
&\leq \mathbb{E}\left[F(w_k)\right] + \alpha\left(\frac{L\alpha\mu_2^2}{2} - \mu_1\right)\mathbb{E}\left[\|\nabla F(w_k)\|^2\right] + \frac{L\alpha^2\mu_2^2}{2}\frac{\omega}{n^2}\sum_{i=1}^{n}\|g_k^i - h_k^i\|^2 + \frac{L\alpha^2\mu_2^2}{2}\sigma^2 - F^* \\
&+ c\frac{L\alpha^2}{2}\left[\frac{1-\gamma_k}{n}\sum_{i=1}^{n}\|h_k^i - h_*^i\|_2^2 + 2\gamma_k\|\nabla F(w_k)\|_2^2\right. \\
&\left. + 4\gamma_k\zeta^2 + \frac{\left(\gamma_k^2\omega - \gamma_k(1-\gamma_k)\right)}{n}\sum_{i=1}^{n}\|g_k^i - h_k^i\|_2^2 + 2\gamma_k\sigma^2\right] \\
&= \mathbb{E}\left[F(w_k)\right] - F^* + c\frac{L\alpha^2(1-\gamma_k)}{2n}\sum_{i=1}^{n}\|h_k^i - h_i^*\|_2^2 - \alpha\left(\mu_1 - \frac{L\alpha\mu_2^2}{2} - cL\alpha\gamma_k\right)\|\nabla F(w_k)\|_2^2 \\
&+ 2cL\alpha^2\gamma_k\zeta^2 + \left(\frac{\mu_2^2}{2} + c\gamma_k\right)L\alpha^2\sigma^2 + \left(\underbrace{\frac{L\alpha^2\mu_2^2}{2}\frac{\omega}{n^2} + \frac{\left(\gamma_k^2\omega - \gamma_k(1-\gamma_k)\right)}{n}}_{:=T(\gamma_k,\alpha)}\right)\sum_{i=1}^{n}\|g_k^i - h_k^i\|_2^2
\end{aligned}$$

A key moment in the proof is to notice that $T(\gamma_k, \alpha) \leq 0$ for our choice of γ_k and α. In fact, we have:

$$\begin{aligned}
T(\gamma_k, \alpha) \leq 0 &\Leftrightarrow \frac{1}{n}\left(\frac{L\alpha^2\mu_2^2}{2}\frac{\omega}{n} + \left(\gamma_k^2\omega - \gamma_k(1-\gamma_k)\right)\right) \leq 0 \\
&\Leftarrow \begin{cases} \alpha \leq \sqrt{\frac{n}{2Lw(w+1)\mu_2^2}} \\ \gamma_k \leq \frac{\sqrt{1 - \frac{2L\alpha^2 w(w+1)\mu_2^2}{n}} + 1}{2(w+1)} \end{cases}
\end{aligned}$$

Therefore, we have:

$$\begin{aligned}
\mathbb{E}\left[\bar{\kappa}^{k+1}\right] &\leq \mathbb{E}\left[F(w_k)\right] - f^* + c\frac{L\alpha^2}{2n}\sum_{i=1}^{n}\|h_k^i - h_i^*\|_2^2 - \alpha\left(\underbrace{\mu_1 - \frac{L\alpha\mu_2^2}{2} - cL\alpha\gamma_k}_{>0 \text{ by our condition on } c}\right)\|\nabla F(w_k)\|_2^2 \\
&+ 2cL\alpha^2\gamma_k\zeta^2 + \left(\frac{\mu_2^2}{2} + c\gamma_k\right)L\alpha^2\sigma^2 \\
\mathbb{E}\left[\bar{\kappa}^{k+1}\right] &\leq \mathbb{E}\left[\bar{\kappa}^{k}\right] - \alpha\left(\mu_1 - \frac{L\alpha\mu_2^2}{2} - cL\alpha\gamma_k\right)\|\nabla F(w_k)\|_2^2 \\
&+ 2cL\alpha^2\gamma_k\zeta^2 + \left(\frac{\mu_2^2}{2} + c\gamma_k\right)L\alpha^2\sigma^2
\end{aligned}$$

Therefore:

$$\mathbb{E}\left[\|\nabla F(w_k)\|_2^2\right] \leq 2\frac{\mathbb{E}_Q\left[\bar{\kappa}^k\right] - \mathbb{E}_Q\left[\bar{\kappa}^{k+1}\right]}{\alpha\left(2\mu_1 - \left(L\alpha\mu_2^2\right) - 2cL\alpha\gamma_k\right)} + \frac{4cL\alpha^2\gamma_k}{\alpha\left(2\mu_1 - \left(L\alpha\mu_2^2\right) - 2cL\alpha\gamma_k\right)}\zeta^2 + \frac{\mu_2^2 + 2c\gamma_k}{\alpha\left(2\mu_1 - \left(L\alpha\mu_2^2\right) - 2cL\alpha\gamma_k\right)}L\alpha^2\sigma^2$$

Summing from 0 and $k-1$, simplifying the telescopic terms yiels:

$$\sum_{j=0}^{k-1}\mathbb{E}\left[\|\nabla F(w_j)\|_2^2\right] \leq 2\frac{\bar{\kappa}^0 - \mathbb{E}_Q\left[\bar{\kappa}^k\right]}{\alpha\left(2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k\right)} + k\frac{4cL\alpha^2}{\alpha\left(2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k\right)}\zeta^2 + k\frac{\mu_2^2 + 2c\gamma_k}{\alpha\left(2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k\right)}L\alpha^2\sigma^2$$

Finally:

$$\frac{1}{k}\sum_{j=0}^{k-1}\mathbb{E}\left[\|\nabla F(w_j)\|_2^2\right] \leq 2\frac{\bar{\kappa}^0 - \mathbb{E}_Q\left[\bar{\kappa}^k\right]}{k\alpha\left(2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k\right)} + \frac{4cL\alpha}{2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k}\zeta^2 + \frac{\mu_2^2 + 2c\gamma_k}{2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k}L\alpha\sigma^2$$

We can drop $\mathbb{E}_Q\left[\bar{\kappa}^k\right]$ because it is positive and that concludes the proof. ■

Corollary 6 *Set $\gamma_k = \gamma$, $\alpha = \frac{2\mu_1 - 1}{L(\mu_2^2+2c\gamma)\sqrt{K}}$ and $h_0 = 0$, after K iterations of algorithm 1, in the nonconvex setting, the error ϵ is at worst $\frac{2}{\sqrt{K}}\frac{L(\mu_2^2+2c\gamma)}{(2\mu_1-1)}\bar{\kappa}^0 + \frac{1}{\sqrt{K}}\frac{4c(2\mu_1-1)}{\mu_2^2+2c\gamma}\zeta^2 + \frac{1}{\sqrt{K}}\frac{\left(\mu_2^2+2c\gamma\right)(2\mu_1-1)}{\mu_2^2+2c\gamma}\sigma^2$.*

Proof It's easy to see that by our choice of γ_k, α and h_0
we have $2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k \geq 1$
Therefore, after the K steps, the error ϵ is upper bounded by:

$$2\frac{\bar{\kappa}^0}{k\alpha\left(2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k\right)} + \frac{4cL\alpha}{2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k}\zeta^2 + \frac{\mu_2^2 + 2c\gamma_k}{2\mu_1 - L\alpha\mu_2^2 - 2cL\alpha\gamma_k}L\alpha\sigma^2 \leq \frac{2}{\sqrt{K}}\frac{L(\mu_2^2 + 2c\gamma)}{(2\mu_1 - 1)}\bar{\kappa}^0 + \frac{1}{\sqrt{K}}\frac{4c\left(2\mu_1 - 1\right)}{\mu_2^2 + 2c\gamma}\zeta^2 + \frac{1}{\sqrt{K}}\frac{\left(\mu_2^2 + 2c\gamma\right)\left(2\mu_1 - 1\right)}{\mu_2^2 + 2c\gamma}\sigma^2$$

■