OPT2025: 17th Annual Workshop on Optimization for Machine Learning

Quasi-Newton Methods for Federated Learning with Error Feedback

Yanlin Wu YANLIN.WU@ALUMNI.MBZUAI.AC.AE
Dmitry Kamzolov KAMZOLOV.OPT@GMAIL.COM
Martin Takáč MARTIN.TAKAC@MBZUAI.AC.AE
Mohamed bin Zayed University of Artificial Intelligence, Masdar City, Abu Dhabi, UAE

Abstract

In this paper, we propose a new class of Quasi-Newton methods for federated learning by integrating them with the error feedback framework—specifically focusing on the EF21 mechanism, which offers stronger theoretical guarantees and improved practical performance compared to earlier approaches. EF21 overcomes several limitations of prior methods, such as dependence on strong assumptions and high communication overhead.

Quasi-Newton methods, particularly the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm, are renowned for their empirical efficiency. By leveraging this, our proposed EF21+L-BFGS algorithm achieves an $\mathcal{O}\left(\frac{1}{T}\right)$ convergence rate in the nonconvex setting and enjoys linear convergence under the Polyak–Łojasiewicz (PL) condition. Through both theoretical analysis and empirical evaluations, we demonstrate the effectiveness of our approach, showing faster convergence and improved model performance compared to existing methods.

1. INTRODUCTION

Federated Learning (FL), an emerging paradigm in machine learning, has gained significant attention due to its ability to collaboratively train models across distributed devices while preserving data privacy and security [15]. Early research laid the groundwork for global model updates without centralized access to data, marking a shift toward privacy-preserving learning frameworks. As FL continues to evolve, enhancing its efficiency and convergence behavior has become a critical area of research.

To tackle challenges such as communication overhead, privacy preservation, and model performance, various compression and sparsification techniques have been proposed. Among them, the *Error Feedback 21* (**EF21**) algorithm stands out for addressing the divergence issues caused by biased compressors. Unlike heuristic-based methods, EF21 provides rigorous convergence guarantees—achieving a fast $\mathcal{O}(1/T)$ rate in smooth nonconvex settings and even linear convergence under the Polyak–Łojasiewicz (PL) condition, despite the presence of biased compression [18].

In this work, we explore the integration of two powerful ideas—**EF21** and the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) method—within the context of federated learning [14].

1.1. Background

Consider a general federated learning (FL) setting with M clients, each holding a local dataset. The goal is to minimize the global empirical loss:

$$\min_{x\in\mathbb{R}^d} \left(f(x) := \tfrac{1}{M}\textstyle\sum_{i=1}^{M} f_i(x)\right), \tag{1}$$

where $x \in \mathbb{R}^d$ denotes the model parameters to be optimized, and $f_i(x)$ represents the local loss function for client i.

In FL, the model is trained locally, but updates are aggregated globally. This requires transmitting model parameters between the server and clients across multiple rounds until convergence, which can result in significant communication overhead [11]. This issue is exacerbated in the era of overparameterized models, where the dimensionality of x is large.

Compression techniques offer a way to mitigate this bottleneck by reducing the amount of information exchanged during communication. Compressing $x \in \mathbb{R}^d$ not only lowers communication costs but also improves privacy by limiting the exposure of sensitive client data [4]. Furthermore, as shown in [16], compression enhances scalability by reducing computational and communication loads, and it improves robustness to heterogeneous client capabilities.

While **unbiased compressors** are analytically convenient, **biased compressors** can often yield better empirical performance [19]. However, their use must be handled carefully, as they may lead to divergence [3].

The original **Error Feedback** (EF) mechanism was introduced heuristically to mitigate divergence caused by naive use of biased compressors [19]. More recently, the **EF21** algorithm proposed in [10, 18] builds upon this idea with a novel error feedback mechanism, offering both theoretical convergence guarantees and superior empirical performance. (See Section A for more details on related work.)

1.2. Motivation

Both **EF** and **EF21** utilize first-order methods with biased compression operators in federated learning, meaning only model parameters and gradients are taken into account. In contrast, second-order methods incorporate information about the curvature of the objective function and typically achieve faster convergence in centralized settings. In many cases, second-order methods require fewer iterations to reach the optimal solution, making them more communication-efficient—an important advantage when communication is a major bottleneck in FL.

The motivation behind this work lies in enhancing convergence speed and model performance by incorporating second-order information. However, storing and transmitting the full Hessian matrix is significantly more expensive than working with vectors. To address this, we adopt a Quasi-Newton method—**L-BFGS** [14]—which approximates the Hessian and its inverse using a sequence of vector operations.

Our contributions can be summarized as follows:

- We introduce a novel integration of Quasi-Newton methods, specifically L-BFGS, with the error feedback framework for federated learning.
- We provide theoretical insights into the convergence properties of the proposed methods, demonstrating their effectiveness in optimizing nonconvex and **Polyak–Łojasiewicz** (PL) objective functions in a distributed setting.
- Through empirical evaluations on benchmark datasets and practical federated learning scenarios, we show that our proposed methods outperform existing approaches in terms of convergence speed and model accuracy.

2. A method combining EF21 and L-BFGS

Here in our work, we utilize the second-order information of the parameters and try to implement it with original **EF21** method, thus a better model performance and convergence rate can be achieved.

2.1. Compressors

Compression of a vector $x \in \mathbb{R}^d$ refers to a mapping $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$. The compressed vector $\mathcal{C}(x)$ is cheaper to communicate than the original vector x. Compressors $\mathcal{C}$ can be broken down into two main categories: unbiased compression operators $\mathcal{U}$ and biased compression operators $\mathcal{B}$ whose properties are as follows: $\mathbf{E}[\mathcal{U}(x)] = x$; $\mathbf{E}[\|\mathcal{U}(x) - x\|^2] \leq w\|x\|^2, w \geq 0$; $\mathbf{E}[\|\mathcal{B}(x) - x\|^2] \leq (1-\alpha)\|x\|^2,\ 0 \leq \alpha \leq 1$. There are 2 classic examples, *Rand-K* and *Top-K*, belonging to unbiased compressor and biased compressor respectively: *Rand-K: pick K elements of x randomly* ; *Top-K: pick top K value elements of x*.

2.2. Review of EF21

Let us start with the **EF21** update in a certain distributed setting. On the $i\text{-}th$ client, its local gradient state can be updated through

$$g_i^t = g_i^{t-1} + c_i^{t-1} = g_i^{t-1} + \mathcal{C}(\nabla f_i(x^t) - g_i^{t-1})),$$

where the compressed vector $\mathcal{C}(\nabla f_i(x^{t+1}) - g_i^t))$ needs aggregating to form the global state update as

$$g^t = \textstyle\sum_{i=1}^M g_i^t = \sum_{i=1}^M (g_i^{t-1} + c_i^{t-1}) = g^{t-1} + \sum_{i=1}^M \mathcal{C}(\nabla f_i(x^t) - g_i^{t-1}).$$

Further, one-step update of the model parameters act on the server side as $x^{t+1} = x^t - \gamma g^t$.

2.3. Specification of our method - the Algorithm

On the one hand, the stepsize γ in **EF21** needs to be manually fine-tuned to ensure good performance. To address this, we introduce the L-BFGS technique, which automatically determines a suitable search direction p^t, and update the model via $x^{t+1} = x^t - p^t$.

In the t-th communication round, the server maintains a memory of m pairs $\{(s^k, y^k)\}_{k=t-m}^{t-1}$, where $s^k = x^{k+1} - x^k$ and $y^k = g^{k+1} - g^k$. The search direction p^t is computed using the two-loop recursion in Algorithm 1, following the standard L-BFGS procedure [14]. Convergence analysis is provided in Section B.

Since high sparsity in the vectors may lead to computational issues such as overflow, we adopt a *Permutation & Partition* trick to construct a relatively denser aggregated vector on the server side—while keeping the number of transmitted bits from clients unchanged. At the beginning of each communication round, we randomly permute and partition the indices of model parameters into M equal-sized, non-overlapping parts $\{\text{index}_i\}_{i=1}^M$, assigning index_i to the i-th client. Accordingly, the local update rule becomes:

$$g_i^t = g_i^{t-1} + c_i^{t-1} = g_i^{t-1} + \mathcal{C}_{\text{index}_i}(\nabla f_i(x^t) - g_i^{t-1}), \tag{2}$$

where $\mathcal{C}_{\text{index}_i}$ denotes the compression operator applied only to the coordinates specified by index_i.

For example, consider the Top-1 compressor introduced by [1]. If we apply compression to the full vector, we have: Top-1$([1, 2, 3, 4, 5]) = [0, 0, 0, 0, 5]$. If we apply compression only to part of the vector with index set $\{0, 3\}$, we get: Top-1$_{\{0,3\}}([1, 2, 3, 4, 5]) = [0, 0, 0, 4, 0]$.

Algorithm 1 L-BFGS Two-Loop Recursion

```
Input: $\{(s^k, y^k)\}_{k=t-m}^{t-1}, g_t$
Output: $p^t$
$p^t = g^t$
for $k = t-1, t-2, \ldots, t-m$ do
    $\alpha^k = \frac{s^k \cdot p}{s^k \cdot y^k}$
    $p^t = p^t - \alpha^k y^k$
end for
$p^t = \left(\frac{s^{t-1} \cdot y^{t-1}}{y^{t-1} \cdot y^{t-1}}\right) p$
for $k = t-m, t-m+1, \ldots, t-1$ do
    $\beta = \frac{y^k \cdot p}{s^k \cdot y^k}$
    $p^t = p^t + (\alpha^k - \beta) s^k$
end for
```

Furthermore, compared to purely first-order methods, our approach better captures curvature information in complex scenarios, mitigates the effects of ill-conditioning, and reduces the need for extensive hyperparameter tuning—ultimately enabling faster convergence and improved performance.

Algorithm 2 EF21+L-BFGS

```
Input: starting point $x^0 \in \mathbb{R}^d$; $g_i^0 = \mathcal{C}(\nabla f_i(x^0))$ for $i = 1, 2, ..., M$ (known by clients and the
server);
$g^0 = \frac{1}{M} \sum_{i=1}^{M} g_i^0$; memory size $m \in \mathbb{R}$
for $t = 0, 1, 2, ..., T-1$ do
    Server $x_1 = x_0 - \gamma g_0$;
    When $t \geq 1$, server computes $p^t$ based on the pairs sequence $\{(s^k, y^k)\}_{k=t-m}^{t-1}$ using Algorithm 1
    Choose stepsize $\alpha^t$ and update parameters, i.e., $x^{t+1} = x^t - \alpha^t p^t$.
    Broadcast $x^{t+1}$ to all clients.
    for all clients $i = 1, 2, ..., M$ in parallel do
        Compress $c_i^t = \mathcal{C}(\nabla f_i(x^{t+1}) - g_i^t)$ and send $c_i^t$ back to server
        Update local state $g_i^{t+1} = g_i^t + c_i^t$
    end for
    Server aggregates gradient $g^{t+1} = g^t + \frac{1}{M} \sum_{i=1}^{M} c_i^t$
    $s^t = x^{t+1} - x^t$, $y^t = g^{t+1} - g^t = \frac{1}{M} \sum_{i=1}^{M} c_i^t$.
    Update pairs sequence $\{(s^k, y^k)\}_{k=t-m+1}^{t}$ through dropping $(s^{t-m}, y^{t-m})$ and adding
    $(s^t, y^t)$
end for
```

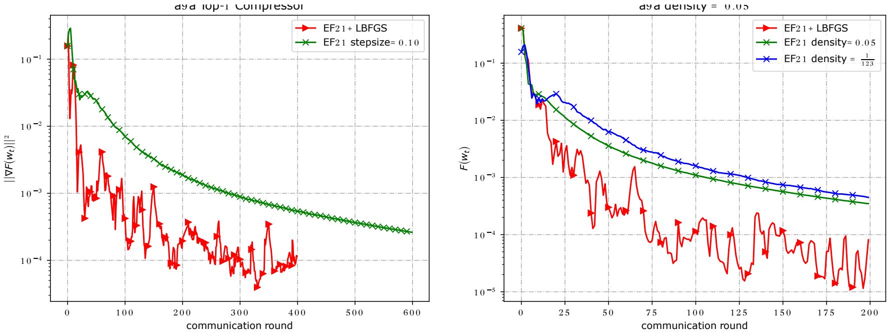

Figure 1: a9a: Convergence under *Top-1* (left) and *Top-6* (right) compression. EF21+L-BFGS is more robust under extreme compression.

3. Experiments

We evaluate the performance of EF21+L-BFGS and EF21 on the a9a and MNIST datasets from LibSVM [6] and [12], respectively. Each dataset is randomly split among $n = 4$ clients. More experiments (including deep learning experiment) and details are provided in Section C.

For EF21+L-BFGS, we adopt a backtracking Armijo line search [17] to determine the stepsize. For EF21, we also evaluate its performance under line search. To model communication constraints, we use the *Top-K* compressor where clients only transmit the top-K entries of their update vectors. Let density $= \frac{K}{d}$ denote the compression ratio, where d is the dimension of the model. We track the total bytes sent from clients to server to measure communication efficiency [11].

3.1. Logistic Regression on a9a

We consider the regularized logistic regression problem:

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + e^{-y_i w^T x_i}) + \frac{\lambda}{2}\|w\|^2,$$

where $x_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$ and λ is the regularization parameter.

3.1.1. EFFECT OF COMPRESSION DENSITY

We study the impact of compression using the extreme case of *Top-1* compression (each client sends only one value per communication round). Fig. 1 (left) shows that both EF21 and EF21+L-BFGS can still converge, albeit with high sparsity. As density increases (e.g., *Top-6*), convergence improves significantly (Fig. 1, right), and EF21+L-BFGS consistently outperforms EF21.

3.1.2. CHALLENGING CURVATURE REGIMES

We next evaluate robustness to poor curvature by modifying the loss landscape to induce more irregular Hessians. Fig. 2 shows that EF21 becomes sensitive to stepsize under ill-conditioned settings, with unstable convergence. In contrast, EF21+L-BFGS maintains stable progress due to its better curvature exploitation. Even when EF21 uses line search, it fails to match the performance of EF21+L-BFGS.

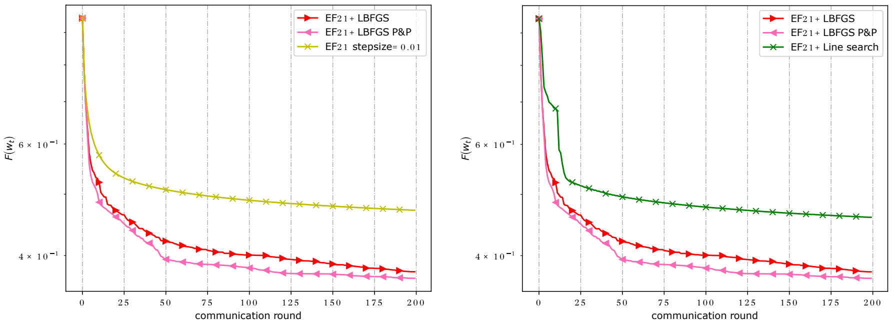

Figure 2: `a9a` with bad curvature: EF21+L-BFGS is more stable. EF21 suffers, even with line search.

4. CONCLUSIONS

In this paper, we presented novel Quasi-Newton methods tailored for federated learning by integrating them with the error feedback framework. By leveraging second-order information through Quasi-Newton methods, particularly L-BFGS method, we demonstrated significant improvements in convergence speed and model performance compared to existing approaches.

Theoretical analysis provided insights into the convergence properties of our proposed methods, highlighting their efficacy in optimizing nonconvex objective functions in a distributed setting. Empirical evaluations on benchmark datasets and practical federated learning scenarios further validated the superiority of our methods.

In conclusion, our work contributes to advancing the state-of-the-art in federated learning by offering efficient and effective optimization techniques. Future research directions may include exploring the applicability of our methods to more complex federated learning scenarios and investigating extensions to incorporate additional constraints or regularization techniques.

References

[1] 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.

[2] Albert S Berahas, Majid Jahani, Peter Richtárik, and Martin Takáč. Quasi-newton methods for machine learning: forget the past, just sample. *Optimization Methods and Software*, 37(5): 1668–1704, 2022.

[3] Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *arXiv preprint arXiv:2002.12410*, 2020.

[4] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[5] Charles G Broyden. Quasi-Newton methods and their application to function minimisation. *Mathematics of Computation*, 21:368–381, 1967. doi: 10.2307/2003239. URL http://www.jstor.org/stable/2003239.

[6] 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.

[7] Roger Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13: 317–322, 1 1970. ISSN 0010-4620. doi: 10.1093/comjnl/13.3.317. URL https://doi.org/10.1093/comjnl/13.3.317.

[8] Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24:23–26, 1970. doi: 10.2307/2004873. URL https://doi.org/10.2307/2004873.

[9] Eduard Gorbunov, Dmitry Kovalev, Dmitry Makarenko, and Peter Richtárik. Linearly converging error compensated sgd. *Advances in Neural Information Processing Systems*, 33: 20889–20900, 2020.

[10] Sarit Khirirat, Abdurakhmon Sadiev, Artem Riabinin, Eduard Gorbunov, and Peter Richtárik. Error feedback under (l_0, l_1)-smoothness: Normalization and momentum. *arXiv preprint arXiv:2410.16871*, 2024.

[11] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] Dong-Hui Li and Masao Fukushima. On the global convergence of the bfgs method for non-convex unconstrained optimization problems. *SIAM Journal on Optimization*, 11(4):1054–1064, 2001.

[14] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989. doi: 10.1007/BF01589116. URL https://doi.org/10.1007/BF01589116.

[15] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[16] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019.

[17] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer New York, NY, 1 edition, 1999. doi: 10.1007/b98874.

[18] Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. Ef21: A new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, 34: 4384–4396, 2021.

[19] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Fifteenth annual conference of the international speech communication association*, 2014.

[20] David F Shanno. Conditioning of Quasi-Newton methods for function minimization. *Mathematics of Computation*, 24:647–656, 1970. doi: 10.2307/2004840. URL https://doi.org/10.2307/2004840.

Appendix A. Related Work

A.1. Error Feedback in Federated Learning

Previous studies have explored error feedback mechanisms in federated learning, demonstrating their potential to enhance convergence rates. Initially proposed on a heuristic level by [19], this innovative approach has been further investigated, with subsequent works establishing its theoretical underpinnings. In strongly convex scenarios, in [3], it was provided analysis within a generalized distributed framework, achieving a linear convergence rate under the condition that $\nabla f_i(x^\star) = 0$ for all i. Additionally, in [9], the authors discuss the convex case, introducing an additional compressor. They achieve a desirable linear convergence rate through the implementation of **EC-GD-DIANA** and **EC-LSVRG-DIANA** whose drawback is the increased communication burden introduced by the use of the extra compressor. While these studies offer insights into error feedback mechanisms, they are subject to various limitations. Moreover, their findings are based on strong assumptions, such as bounded gradients ($||\nabla f_i(x)||^2 \leq G^2$) and bounded dissimilarity ($\frac{1}{n}||f_i(x) - f(x)||^2 \leq G^2$), which are seldom met in practice [18]. Building upon this foundation, [18] propose a novel error feedback mechanism that achieves a linear convergence rate with only standard assumptions of smoothness and lower boundness. Our work also adheres to these standard assumptions.

A.2. Quasi-Newton/L-BFGS Method

Quasi-Newton methods, specifically the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [5, 7, 8, 20], have garnered significant attention for their efficiency and robustness[17]. These methods iteratively update an approximation of the inverse Hessian matrix based on gradient information, offering advantages over first-order methods, such as gradient descent, in terms of convergence speed and robustness.Unlike Newton's method, which demands explicit computation of the Hessian matrix, quasi-Newton methods approximate it iteratively, circumventing the computational overhead associated with Hessian computations.

Two-loop recursion as demonstrated in Algoriithm1 is one of the most common implementations in L-BFGS[14], which stores the last m BFGS update pairs (s_t, y_t) and uses two loops to update the approximation of the inverse Hessian to perform the matrix operations of BFGS implicitly.

To ensure L-BFGS works in a nonconvex setting, [13] further adopt a cautious strategy which ensures the generated sequence of (inverse) Hessian $\{H^t\}_{t=0,1,2,\ldots}$ is bounded away from zero. At the t-th iteration, the update of the (inverse) Hessian approximation, denoted as H^t, only occurs when the set of curvature pairs satisfies the condition:

$$s^T y \geq \epsilon ||s||^2, \tag{3}$$

where $\epsilon \geq 0$ is a predetermined constant. If no curvature pairs satisfying this condition, then the new(inverse) Hessian approximation is set to $H^t = I$.

Appendix B. Theory

Assumption 1 *Every f_i has L_i lipschitz gradient, i.e.,*

$$||\nabla f_i(x) - \nabla f_i(y)|| \leq L_i ||x - y||$$

for all $x, y \in \mathbb{R}^d$ and $f^{\inf} \overset{def}{=} \inf_{x \in \mathbb{R}^d} f(x) \geq -\infty$

Assumption 2 *f_i is twice continuously differentiable.*

Under Assumption 1,$f = \sum_{i=1}^n f_i$ also has Lipschitz gradient with $\tilde{L} \overset{def}{=} \left(\frac{1}{n}\sum_{i=1}^n L_i^2\right)^{\frac{1}{2}} \geq \frac{1}{n}\sum_{i=1}^n L_i$.

We adopt the cautious strategy as described in [13], which, as a result, can bound the (inverse) Hessian from both below and above.

Lemma 3 *Under Assumption 1, 2, the inverse Hessian approximation H^t is generated using modified two-loop recursion L-BFGS method with the condition, where H^t is updated when the curvature pairs satisfying the condition $s^T y \geq \epsilon ||s||^2$ where $\epsilon \geq 0$ is a predetermined constant. Otherwise H^t is set to be identity matrix. Then, there exist constants $0 < \mu_1 \leq \mu_2$ such that*

$$\mu_1 I \preceq H^t \preceq \mu_2 I \tag{4}$$

This lemma is a direct result from work[2], ensuring the eigenvalues of the (inverse) Hessian approximations generated by the L-BFGS method are bounded above and away from zero, which turns out to stabilize the L-BFGS method in distributed setting especially under big sparsify due to the use of compressor. On the other hand, such lower bound ensures our method can also work in nonconvex setting.

Theorem 4 *Under Assumption1 and stepsize α in algorithm satisfying*

$$0 \leq \alpha \leq \left(\frac{L}{1-\tilde{\mu}} + \tilde{L}\sqrt{\frac{\beta}{\theta(1-\tilde{\mu})}}\right)^{-1}, \tag{5}$$

where $\tilde{\mu} = [\max\{(1 - \frac{1}{\mu_2}), (\frac{1}{\mu_1} - 1)\}]^2$. For iteration $T \geq 1$

$$\sum_{t=0}^{T-1} \frac{1}{T}\mathbb{E}[||\nabla f(x^t)||^2] \leq \frac{2(f(x^0) - f^{inf})}{\alpha T} + \frac{\mathbb{E}[G^0]}{\theta T} \tag{6}$$

Theorem 4 shows that the introduction of LBFGS method into **EF21** inherit the good theoretical convergence rate of **EF21** which has an ideal $\mathcal{O}(\frac{1}{T})$ convergence rate.

Further, under **PL** condition, our method enjoys a linear convergence rate.

Assumption 5 *There exists $\mu \geq 0$ such that $f(x) - f(x^\star) \leq \frac{1}{2\mu}||\nabla f(x)||^2$ for all $x \in \mathbb{R}^d$, where $x^\star = \operatorname{argmin} f$.*

Theorem 6 *Under the Assumption 5,2, 1 and stepsize α in algorithm satisfying*

$$0 \leq \alpha \leq \min\left\{\left(\frac{L}{1-\tilde{\mu}} + \tilde{L}\sqrt{\frac{\beta}{\theta(1-\tilde{\mu})}}\right)^{-1}, \frac{\theta}{2\mu}\right\}$$

Denote $\Psi^t = f(x^t) - f(x^\star) + \frac{\alpha}{\theta}G^t$, we have for any $T \geq 0$:

$$\mathbb{E}[\Psi^T] \leq (1 - \alpha\mu)^T \mathbb{E}[\Psi^0]$$

Table 1: Datasets details

Dataset	# of clients	# of datapoints	# of features
Mnist	4	60000	28×28
a9a	4	32561	123

Appendix C. Detailed Numerical Experiments

C.1. Datasets, experiment setting and details

In the experiment, we utilize two main datasets: ’a9a’ from LibSVM [6] and ’MNIST’ from [12]. These datasets are randomly partitioned into $n = 4$ equal-sized cohorts and assigned to $n = 4$ nodes, respectively. For further details, please refer to Table I. All experiments are conducted within a Python 3.9 environment.

For EF21+LBFGS, we use a backtracking Armijo line search [17] to determine the suitable stepsize. In the experiment, we also implement the line search with original EF21 to find suitable step length.

For compressor, we use the idea of *Top-K* compressor, nodes only send *top K* values of the vector. Considering the choice of K is also dependent on n, the dimension of the original vector. We define density for a *Top-K* compressor as $density = \frac{K}{n} \in [0, 1]$ to better summarize such relation. When density $= 1$, there is no compression at all and nodes need to transmit the original vector. When density $= 0$, nodes compress everything and nothing will be sent back to server for aggregation.

As upload speeds are typically much slower than download speeds , the transmission process from nodes to the server is considered much more challenging than its reverse [11]. We use the number of bytes transferred from clients to server as a measure of how much communication burden is incurred.

C.2. Experiment with logistic regression

Considering the binary logistic regression problem with a regularizer as follows:

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{(-y_i w^T x_i)}) + \frac{\lambda}{2} \|w\|^2 \tag{7}$$

where $w \in \mathbb{R}^d$ denotes model parameters and $y_i \in \{-1, 1\}, x_i$ are training data. λ is regularizer parameter. We presented results on $a9a$ which is a binary classification dataset and compared the performance of our proposed EF21+L-BFGS and classical EF21 method.

C.2.1. TUNING DENSITY

Firstly, we choose the Top-1 compressor as an extreme case to test the robustness of our proposed EF21+LBFGS. With the use of Top-1 compressor, each node sends only one number out of the entire vector to the server for aggregation in each communication process, resulting in significant sparsity. The horizontal axis represents the number of bytes transferred from clients to server. We can observe from Fig.1 that both methods converge to the optimal value steadily. Furthermore, Fig.2 shows that the increasing the density enables capturing more information from the original vector.

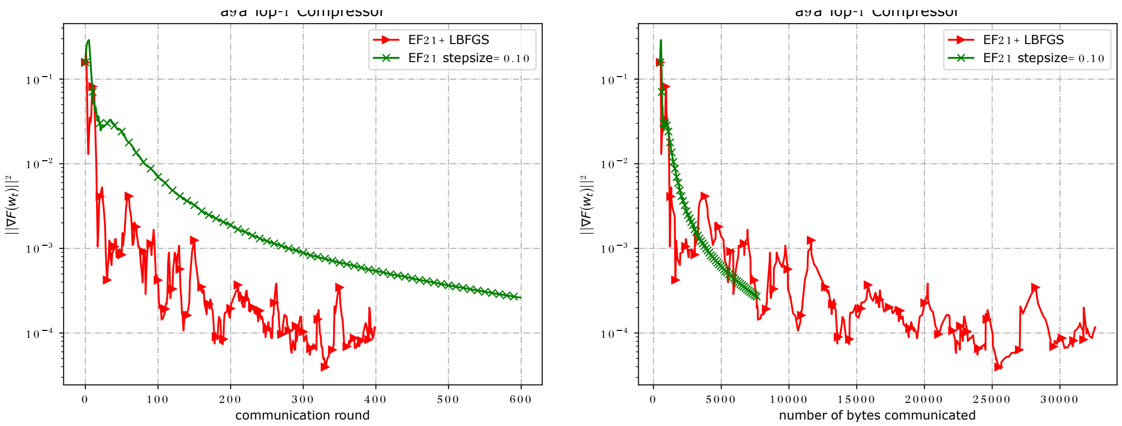

Figure 3: $a9a$ with Top-1 compressor: Due to the application of Line search procedure in EF21+LBFGS, nodes need to transmit the function value to server for several times to determine the appropriate stepsize automatically. That's why it needs to send more bytes.

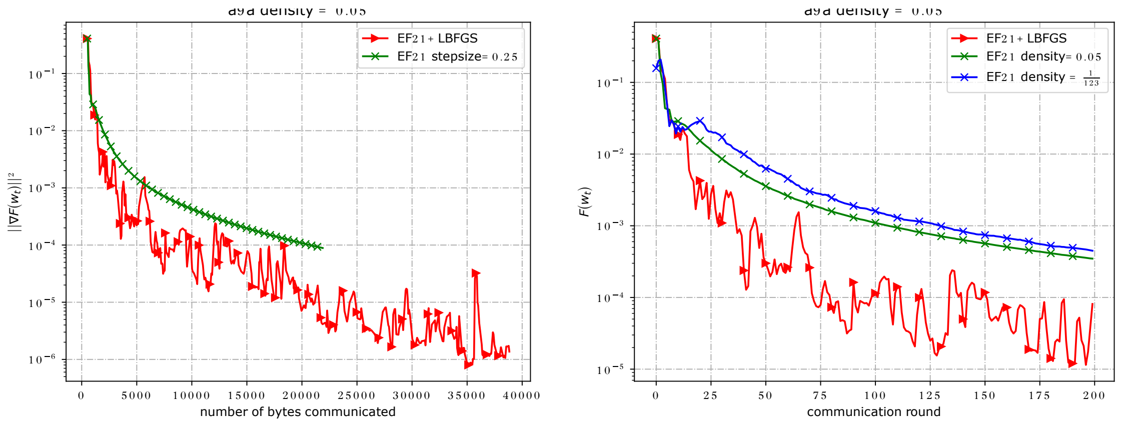

Figure 4: $a9a$ with Top-6 compressor: Compared with Top-1 compressor, increasing density means more information is communicated thus both EF21 and EF21+LBFGS converge faster, and less rounds are needed to achieve the same performance. EF21+LBFGS outperforms EF21

C.2.2. BAD CURVATURE CONDITION

L-BFGS is known for its ability to effectively utilize curvature information from the loss function. In this section, we conduct experiments on the $a9a$ dataset and explore the performance of EF21 and EF+L-BFGS when the curvature information is worsened to different extents. Density is fixed to be 0.05 in this part.

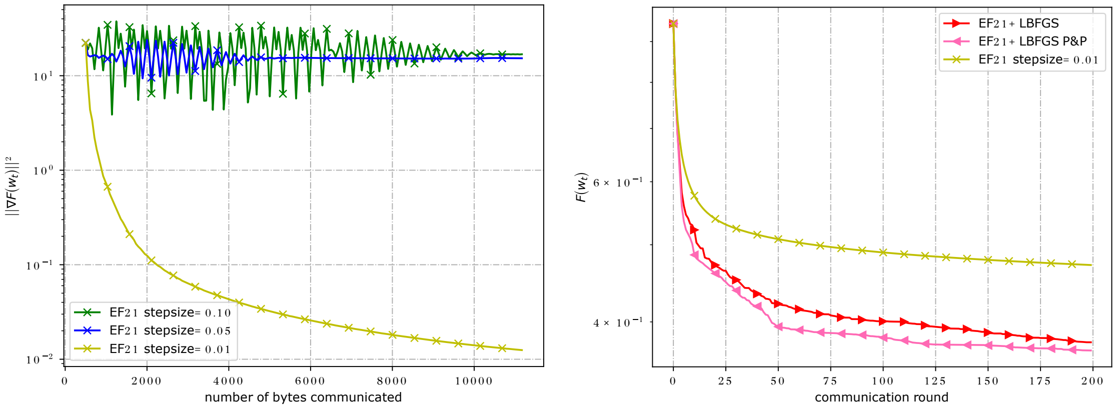

Figure 5: $a9a$ with bad curvature: When the Hessian of the loss becomes more complicated, which means the eigenvalues vary a lot and can hardly cluster. EF21 becomes more sensitive to the stepsize, which may even lead to fluctuation a lot.

Fig.3 shows that EF21 is sensitive to the changes in stepsize, we further introduce line-search method in EF21 case as well so that the stepsize can be tuned automatically during training.

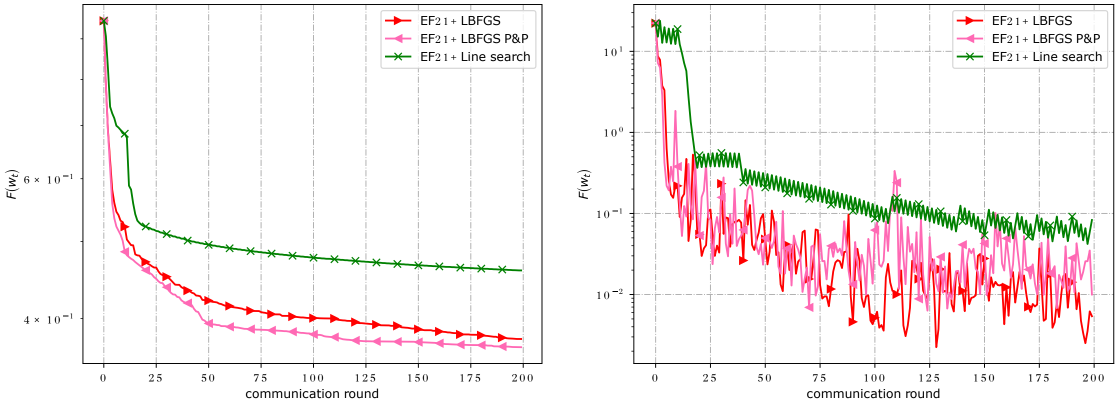

Figure 6: $a9a$ with bad curvature: EF21 with line search cannot obtain the optimal solution also.

C.3. Deep learning experiments

In this section, we conduct several deep-learning experiments for multi-class image classification. In particular, we compare our EF21+LBFGS method to EF21 by running LeNet model on the MNIST[12]dataset. Results demonstrate the performance of both EF21+LBFGS and EF21 in non-convex regime.

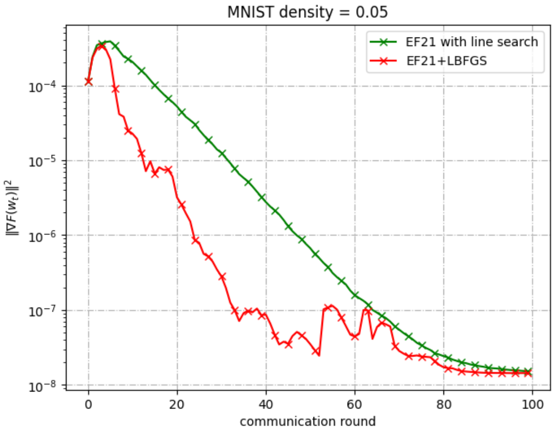

Figure 7: 'MNIST' trained with LeNet: line search is used to determine the steplength in EF21 method.

Appendix D. Notations, Lemmas and Proofs

Biased Compressor: A compressor: $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is biased if there exists $0 \leq \alpha \leq 1$

$$\mathbb{E}[\|\mathcal{C}(x) - x\|^2] \leq (1-\alpha)\|x\|^2 \tag{8}$$

Top-k Compressor: Top-k is defined as:

$$\text{Top-}k(x) = m \cdot x$$

where m is a mask s.t. $m_i = 1$ if $i \in \mathcal{T}_k(x)$, 0 otherwise. $\mathcal{T}_k(x)$ is a set of top-k coordinates in magnitude of vector x

Lemma 7 *Top-k is a biased compressor, for any vector $x \in \mathbb{R}^d$, Top-$k \in \mathbb{B}(k/d)$*

Proof $\mathbb{E}[||\text{Top-}k(x) - x||^2] = \mathbb{E}[||m \cdot x - x||^2] = ||m \cdot x - x||^2 = \sum_{i\in[d]\setminus\mathcal{T}_k(x)} |x_i|^2 = ||x||^2 - \sum_{i\in\mathcal{T}_k(x)} |x_i|^2$

Considering that $\mathcal{T}_k(x)$ is a set of Top-k coordinates in magnitude of the vector x, we have: $\frac{d}{k}\sum_{i\in\mathcal{T}_k(x)} |x_i|^2 \geq ||x||^2$

$\mathbf{E}[||\text{Top-}k(x) - x||^2] = ||x||^2 - \sum_{i\in\mathcal{T}_k(x)} |x_i|^2 \leq ||x||^2 - \frac{k}{d}||x||^2 = (1-\frac{k}{d})||x||^2$

It's concluded that Top-$k \in \mathbb{B}(k/d)$. ■

Lemma 8 *Permutation&Partition with Top-k is a biased compressor.*

Proof Suppose we have a vector $x \in \mathbb{R}^d$ and n clients in the federated setting. Server permutates $\Delta(x)$ standing for the coordinate set of x and partitions it into n non-overlapping parts $\{\Delta(x)^1, \Delta(x)^2, ..., \Delta(x)^n\}$, where x^i denotes the permutated&partitioned vector on i-th client with respect to $\Delta(x)^i$. Top-k compressor then would be applied to x^i.

The compressed vector on i-th client can be expressed as Top-$k(x^i)$.

On the clients' prospective, which part of x should be compressed is randomly sent from server. This progress can be considered as applying rand-$\frac{d}{n}$ to the original x.

$$\begin{aligned}
\mathbb{E}[||\text{Top-}k(x^i) - x||^2] &= \mathbb{E}[||\text{Top-}k(x^i) - x^i + x^i - x||^2] \leq \mathbb{E}[||\text{Top-}k(x^i) - x^i||^2] + \mathbb{E}[||x^i - x||^2] \\
&\leq (1-\tfrac{k}{d})\mathbb{E}||x^i||^2 + \mathbb{E}[||x^i - x||^2] \\
&= (1-\tfrac{k}{d})||x^i||^2 + \mathbb{E}[||x^i||^2] + \mathbb{E}[||x||^2] - 2\mathbb{E}[x^i]\mathbb{E}[x] \\
&\leq (1-\tfrac{k}{d})\mathbb{E}||x^i||^2 + \tfrac{1}{n}||x||^2 + ||x||^2 - \tfrac{2}{n}||x||^2 \\
&\leq (1-\tfrac{k}{d}) \times \tfrac{1}{n}||x||^2 + (1-\tfrac{1}{n})||x||^2 = (1-\tfrac{k}{nd})||x||^2
\end{aligned}$$

It's concluded that P&P with Top-$k \in \mathbb{B}(k/nd)$. ■

Lemma 9 *Let $\mathcal{C} \in \mathbb{B}(\alpha)$ for $0 \leq \alpha \leq 1$. Define $G_i^t = ||g_i^t - \nabla f_i(x^t)||^2$, where $g_i^{t+1} = g_i^t + c_i^t = g_i^t + \mathcal{C}(\nabla f_i(x^{t+1}) - g_i^t))$, $G^t \overset{def}{=} \frac{1}{n}\sum_{i=1}^n G_i^t = \frac{1}{n}\sum_{i=1}^n \mathbb{E}[||g_i^t - \nabla f_i(x^t)||^2]$ and $W^t = \{g_1^t, g_2^t, ..., g_n^t, x^t, x^{t+1}\}$. For any $s > 0$ we have*

$$\mathbb{E}[G_i^{t+1}|W^t] \leq (1-\theta)||g_i^t - \nabla f_i(x^t)||^2 + \beta||\nabla f_i(x^{t+1}) - \nabla f_i(x^t)||^2 \tag{9}$$

where $(1-\theta) = (1-\alpha)(1+s)$, $\beta = (1-\alpha)(1+s^{-1})$

$$\begin{aligned}\mathbb{E}[G_i^{t+1}|W^t] =&\mathbb{E}[||g_i^{t+1} - \nabla f_i(x^{t+1})||^2|W^t] = \mathbb{E}[||g_i^t + \mathcal{C}(\nabla f_i(x^{t+1}) - g_i^t)) - \nabla f_i(x^{t+1})||^2|W^t] \\ \leq&(1-\alpha)||g_i^t - \nabla f_i(x^{t+1})||^2 \\ \leq&(1-\alpha)(1+s)||g_i^t - \nabla f_i(x^t)||^2 + (1-\alpha)(1+s^{-1})||\nabla f_i(x^{t+1}) - \nabla f_i(x^t)||^2\end{aligned}$$

here we set $(1-\theta) = (1-\alpha)(1+s)$, $\beta = (1-\alpha)(1+s^{-1})$, inequality can be rewritten as:

$$\mathbb{E}[G_i^{t+1}|W^t] \leq (1-\theta)||g_i^t - \nabla f_i(x^t)||^2 + \beta||\nabla f_i(x^{t+1}) - \nabla f_i(x^t)||^2$$

Further we have

$$\begin{aligned}\mathbb{E}[G^{t+1}|W^t] =&\tfrac{1}{n}\sum_{i=1}^{n}\mathbb{E}[||g_i^{t+1} - \nabla f_i(x^{t+1})||^2|W^t] \\ \leq&\tfrac{1}{n}\sum_{i=1}^{n}[(1-\theta)||g_i^t - \nabla f_i(x^t)||^2 + \beta||\nabla f_i(x^{t+1}) - \nabla f_i(x^t)||^2] \\ =&(1-\theta)\tfrac{1}{n}\sum_{i=1}^{n}[||g_i^t - \nabla f_i(x^t)||^2 + \tfrac{1}{n}\sum_{i=1}^{n}\beta||\nabla f_i(x^{t+1}) - \nabla f_i(x^t)||^2 \\ =&(1-\theta)G^t + \tfrac{1}{n}\sum_{i=1}^{n}\beta||\nabla f_i(x^{t+1}) - \nabla f_i(x^t)||^2 \leq (1-\theta)G^t + \beta(\tfrac{1}{n}\sum_{i=1}^{n}L_i^2)||x^{t+1} - x^t||^2\end{aligned}$$

Using Tower property of expectation, $\tilde{L} \overset{def}{=} \sqrt{\frac{1}{n}\sum_{i=1}^{n} L_i^2}$

$$\mathbb{E}[G^{t+1}] = \mathbb{E}[\mathbb{E}[G^{t+1}|W^t]] \leq (1-\theta)\mathbb{E}[G^t] + \beta\tilde{L}^2\mathbb{E}[||x^{t+1} - x^t||^2] \tag{10}$$

Condition: At the $t-th$ iteration, we update the (inverse) Hessian approximation H^t using only the set of curvature pairs that satisfy: $s^T y \geq \epsilon||s||^2$ where $\epsilon \geq 0$ is a predetermined constant. If no curvature pairs satisfying this condition, then the new(inverse) Hessian approximation is set to $H^t = I$.

Lemma 10 *Under Assumption2, 1, the inverse Hessian approximation H^t is generated using modified two-loop recursion L-BFGS method with the condition, where H^t is updated when the curvature pairs satisfying the condition, otherwise H^t is set to be identity matrix. Then, there exist constants $0 < \mu_1 \leq \mu_2$ such that*

$$\mu_1 I \preceq H^t \preceq \mu_2 I$$

Lemma 11 *Let $a, b \geq 0$. If $0 \leq \alpha \leq \frac{1}{\sqrt{a}+b}$, then $a\alpha^2 + b\alpha \leq 1$. Further, we set $a = \frac{\tilde{L}^2\beta}{\theta(1-\tilde{\mu})}$,$b = \frac{L}{1-\tilde{\mu}}$, if $0 \leq \alpha \leq \frac{1}{\sqrt{a}+b}$,we have*

$$\tfrac{\beta\tilde{L}^2}{\theta}\alpha^2 + L\alpha \leq 1 - \tilde{\mu} \tag{11}$$

Theorem 12 *Under Assumption1 and stepsize α in algorithm satisfying*

$$0 \leq \alpha \leq \left(\tfrac{L}{1-\tilde{\mu}} + \tilde{L}\sqrt{\tfrac{\beta}{\theta(1-\tilde{\mu})}}\right)^{-1}$$

where $\tilde{\mu} = [\max\{(1-\frac{1}{\mu_2}), (\frac{1}{\mu_1} - 1)\}]^2$. For iteration $T \geq 1$

$$\sum_{t=0}^{T-1}\tfrac{1}{T}\mathbb{E}[||\nabla f(x^t)||^2] \leq \tfrac{2(f(x^0) - f^{inf})}{\alpha T} + \tfrac{\mathbb{E}[G^0]}{T\theta} \tag{12}$$

Proof $x^{t+1} = x^t - \alpha H^t g^t, \mu_1 I \preceq H^t \preceq \mu_2 I$

$$\begin{aligned}
f(x^{t+1}) =& f(x^t - \alpha H^t g^t) \leq f(x^t) + \langle \nabla f(x^t), -\alpha H^t g^t \rangle + \tfrac{L}{2}|| - \alpha H^t g^t||^2 \\
=& f(x^t) - \alpha \langle \nabla f(x^t), H^t g^t \rangle + \tfrac{L}{2}||\alpha H^t g^t||^2 \\
=& f(x^t) - \tfrac{\alpha}{2}(||\nabla f(x^t)||^2 + ||H^t g^t||^2 - ||\nabla f(x^t) - H^t g^t||^2) + \tfrac{L}{2}||\alpha H^t g^t||^2 \\
=& f(x^t) - \tfrac{\alpha}{2}||\nabla f(x^t)||^2 + \left(\tfrac{L}{2} - \tfrac{1}{2\alpha}\right)||\alpha H^t g^t||^2 + \tfrac{\alpha}{2}||\nabla f(x^t) - H^t g^t||^2 \qquad (13)
\end{aligned}$$

$$\begin{aligned}
||\nabla f(x^t) - H^t g^t||^2 &= ||\nabla f(x^t) - g^t + g^t - H^t g^t||^2 \\
&\leq ||\nabla f(x^t) - g^t||^2 + ||(H^{t^{-1}} - I) H^t g^t||^2 \\
&\leq ||\nabla f(x^t) - g^t||^2 + \tilde{\mu}||H^t g^t||^2
\end{aligned}$$

where $\tilde{\mu} = [\max\{(1 - \frac{1}{\mu_2}), (\frac{1}{\mu_1} - 1)\}]^2$

$$\begin{aligned}
f(x^{t+1}) \leq& f(x^t) - \tfrac{\alpha}{2}||\nabla f(x^t)||^2 + \left(\tfrac{L}{2} - \tfrac{1}{2\alpha}\right)||\alpha H^t g^t||^2 + \tfrac{\alpha}{2}(||\nabla f(x^t) - g^t||^2 + \tilde{\mu}||g^t||^2) \\
=& f(x^t) - \tfrac{\alpha}{2}||\nabla f(x^t)||^2 + \tfrac{\alpha}{2}||\nabla f(x^t) - g^t||^2 + \left(\tfrac{L}{2} - \tfrac{1}{2\alpha}\right)||\alpha H^t g^t||^2 + \tfrac{\alpha\tilde{\mu}}{2}||H^t g^t||^2 \\
=& f(x^t) - \tfrac{\alpha}{2}||\nabla f(x^t)||^2 + \tfrac{\alpha}{2}||\nabla f(x^t) - g^t||^2 + \left(\tfrac{L\alpha^2}{2} - \tfrac{\alpha}{2} + \tfrac{\alpha\tilde{\mu}}{2}\right)||H^t g^t||^2
\end{aligned}$$

Using Jensen's inequality,

$$f(x^{t+1}) \leq f(x^t) - \tfrac{\alpha}{2}||\nabla f(x^t)||^2 + \tfrac{\alpha}{2}G^t + \left(\tfrac{L\alpha^2}{2} - \tfrac{\alpha}{2} + \tfrac{\alpha\tilde{\mu}}{2}\right)||H^t g^t||^2 \tag{14}$$

$$\mathbb{E}[f(x^{t+1}) - f^{inf}] \leq \mathbb{E}[f(x^t) - f^{inf}] - \tfrac{\alpha}{2}\mathbb{E}[||\nabla f(x^t)||^2] + \tfrac{\alpha}{2}\mathbb{E}[G^t] + \left(\tfrac{L\alpha^2}{2} - \tfrac{\alpha}{2} + \tfrac{\alpha\tilde{\mu}}{2}\right)\mathbb{E}[||H^t g^t||^2] \tag{15}$$

Denote $\delta^t = \mathbb{E}[f(x^t) - f^{inf}]$, $s^t = \mathbb{E}[G^t], r^t = \mathbb{E}[||H^t g^t||^2]$

$$\delta^{t+1} \leq \delta^t - \tfrac{\alpha}{2}\mathbb{E}[||\nabla f(x^t)||^2] + \tfrac{\alpha}{2}s^t - \left(\tfrac{\alpha}{2} - \tfrac{L\alpha^2}{2} - \tfrac{\alpha\tilde{\mu}}{2}\right) r^t \tag{16}$$

update rule: $x^{t+1} = x^t - \alpha H^t g^t$

$$\begin{aligned}
\mathbb{E}[G^{t+1}] = \mathbb{E}[\mathbb{E}[G^{t+1}|W^t]] \leq (1-\theta)\mathbb{E}[G^t] + \beta\tilde{L}^2\mathbb{E}[||x^{t+1} - x^t||^2] \\
s^{t+1} \leq (1-\theta)s^t + \beta\tilde{L}^2\mathbb{E}[|| - \alpha H^t g^t||^2] \leq (1-\theta)s^t + \beta\tilde{L}^2\alpha^2 r^t \qquad (17)
\end{aligned}$$

Then by adding (16) with a $\frac{\alpha}{2\theta}$ (17),we have:

$$\begin{aligned}
\delta^{t+1} + \tfrac{\alpha}{2\theta}s^{t+1} &\leq \delta^t - \tfrac{\alpha}{2}\mathbb{E}[||\nabla f(x^t)||^2] + \tfrac{\alpha}{2}s^t - \left(\tfrac{\alpha}{2} - \tfrac{L\alpha^2}{2} - \tfrac{\alpha\tilde{\mu}}{2}\right) r^t + \tfrac{\alpha}{2\theta}[(1-\theta)s^t + \beta\tilde{L}^2\alpha^2 r^t] \\
&\leq \delta^t + \tfrac{\alpha}{2\theta}s^t - \tfrac{\alpha}{2}\mathbb{E}[||\nabla f(x^t)||^2] - \left(\tfrac{\alpha}{2} - \tfrac{L\alpha^2}{2} - \tfrac{\alpha\tilde{\mu}}{2} - \tfrac{\alpha}{2\theta}\beta\tilde{L}^2\alpha^2\right) r^t \\
&\overset{(11)}{\leq} \delta^t + \tfrac{\alpha}{2\theta}s^t - \tfrac{\alpha}{2}\mathbb{E}[||\nabla f(x^t)||^2]
\end{aligned}$$

Unroll the recurrence,

$$0 \leq \delta^T + \tfrac{\alpha}{2\theta}s^T \leq \delta^0 + \tfrac{\alpha}{2\theta}s^0 - \sum_{t=0}^{T-1} \tfrac{\alpha}{2}\mathbb{E}[||\nabla f(x^t)||^2] \tag{18}$$

$$\sum_{t=0}^{T-1} \tfrac{1}{T}\mathbb{E}[||\nabla f(x^t)||^2] \leq \tfrac{2}{\alpha T}(\delta^0 + \tfrac{\alpha}{2\theta}s^0) \tag{19}$$

■

Theorem 13 *Under thr Assunption 5,2, 1 and stepsize α in algorithm satisfying*

$$0 \le \alpha \le \min\left\{ \left(\frac{L}{1-\tilde{\mu}} + \tilde{L}\sqrt{\frac{\beta}{\theta(1-\tilde{\mu})}} \right)^{-1}, \frac{\theta}{2\mu} \right\}$$

Denote $\Psi^t = f(x^t) - f(x^\star) + \frac{\alpha}{\theta}G^t$, we have for any $T \ge 0$:

$$\mathbb{E}[\Psi^T] \le (1-\alpha\mu)^T \mathbb{E}[\Psi^0] \tag{20}$$

Proof We proceed from (14) which says

$$f(x^{t+1}) \le f(x^t) - \frac{\alpha}{2}||\nabla f(x^t)||^2 + \frac{\alpha}{2}G^t + \left(\frac{L\alpha^2}{2} - \frac{\alpha}{2} + \frac{\alpha\tilde{\mu}}{2} \right) ||H^t g^t||^2$$

and substract $f(x^\star)$ from both sides:

$$\begin{aligned} \mathbb{E}[f(x^{t+1}) - f(x^\star)] \le & \mathbb{E}[f(x^t) - f(x^\star)] - \frac{\alpha}{2}\mathbb{E}[||\nabla f(x^t)||^2] + \frac{\alpha}{2}\mathbb{E}[G^t] + \left(\frac{L\alpha^2}{2} - \frac{\alpha}{2} + \frac{\alpha\tilde{\mu}}{2} \right) \mathbb{E}[||H^t g^t||^2] \\ \overset{(5)}{\le} & (1-\alpha\mu)\mathbb{E}[f(x^{t+1}) - f(x^\star)] + \frac{\alpha}{2}\mathbb{E}[G^t] + \left(\frac{L\alpha^2}{2} - \frac{\alpha}{2} + \frac{\alpha\tilde{\mu}}{2} \right) \mathbb{E}[||H^t g^t||^2] \end{aligned} \tag{21}$$

Denote $\delta^t = \mathbb{E}[f(x^t) - f(x^\star)]$, $s^t = \mathbb{E}[G^t], r^t = \mathbb{E}[||H^t g^t||^2]$. Then by adding (21) with a $\frac{\alpha}{\theta}$ (17),we have:

$$\begin{aligned} \delta^{t+1} + \frac{\alpha}{\theta}s^{t+1} \le & (1-\alpha\mu)\delta^t + \frac{\alpha}{2}s^t - \left(\frac{\alpha}{2} - \frac{L\alpha^2}{2} - \frac{\alpha\tilde{\mu}}{2} \right) r^t + \frac{\alpha}{\theta}[(1-\theta)s^t + \beta\tilde{L}^2\alpha^2 r^t] \\ = & (1-\alpha\mu)\delta^t + \frac{\alpha}{\theta}(1-\frac{\theta}{2})s^t - \left(\frac{\alpha}{2} - \frac{L\alpha^2}{2} - \frac{\alpha\tilde{\mu}}{2} - \frac{\alpha}{2\theta}\beta\tilde{L}^2\alpha^2 \right) r^t \\ \le & (1-\alpha\mu)\delta^t + \frac{\alpha}{\theta}(1-\frac{\theta}{2})s^t \le (1-\alpha\mu)\delta^t + \frac{\alpha}{\theta}(1-\alpha\mu)s^t \end{aligned}$$

Thus, $\delta^{t+1} + \frac{\alpha}{\theta}s^{t+1} \le (1-\alpha\mu)(\delta^t + \frac{\alpha}{\theta}s^t)$.

Unroll the recurrence, for any $T \ge 0$, we have $\mathbb{E}[\Psi^T] \le (1-\alpha\mu)^T \mathbb{E}[\Psi^0]$. ■