Parameter-Agnostic Error Feedback Enhanced With Hessian-Corrected Momentum

Abdurakhmon Sadiev ABDURAKHMON.SADIEV@KAUST.EDU.SA
Yury Demidovich
Igor Sokolov
Grigory Malinovsky
Sarit Khirirat
Peter Richtárik
KAUST, Saudi Arabia

Abstract

Advanced machine learning models often rely on massive datasets distributed across many nodes. To reduce communication overhead in large-scale stochastic optimization, compression is widely used, though it may introduce noise and harm convergence. Error feedback mitigates this by accumulating and reusing compression error, while Hessian-vector products provide variance reduction and improve complexity. Building on these ideas, we design a distributed algorithm for finding ε-stationary points of nonconvex L-smooth functions that leverages error feedback, normalization, and second-order momentum. Unlike prior methods requiring problem parameters to tune stepsizes, our algorithm is parameter-agnostic: it uses $\mathcal{O}(1)$ batch size and a time-varying learning rate independent of L and the functional gap. The method achieves $\mathcal{O}(\varepsilon^{-3})$ communication complexity.

1. Introduction

Distributed optimization has gained significant attention in Machine Learning (ML) due to the growing scale of modern problems, such as training deep neural networks with billions of parameters on massive datasets [15, 47]. To keep training feasible, tasks like stochastic gradient computation are parallelized via distributed methods [33, 52, 90]. These methods are particularly relevant in Federated Learning (FL), where data is naturally distributed and must remain private [42, 49, 55].

A central challenge in distributed training is communication efficiency. Compression techniques [3, 38, 86] reduce communication by applying a compressor to transmitted gradients. However, aggressive compression can harm training or even cause divergence. Error feedback methods address this by compensating lost information, e.g., EF14 [5, 32, 74, 77, 87], EF21 and its variants [26, 27, 29, 34, 45, 71].

Normalization [35, 89, 90] further stabilizes error feedback in nonconvex optimization and reduces parameter sensitivity. Yet normalized updates may amplify errors; large batches mitigate this but are costly. Cutkosky and Mehta [19] showed momentum can remove the need for large batches when optimizing nonconvex functions.

Finding global optima of nonconvex functions is NP-hard [57], so analysis focuses on critical points. SGD finds an ε-approximate critical point in $\mathcal{O}(\varepsilon^{-4})$ stochastic gradients [30]. Despite heuristics such as adaptive methods and learning-rate schedules [46, 54, 70], no asymptotic improvement over this rate exists, which is optimal for first-order methods [9]. To go beyond, limited second-order information can be exploited.

Tran and Cutkosky [83] proposed SGDHess, which uses Hessian-vector products to correct momentum bias and achieves the optimal $\mathcal{O}(\varepsilon^{-3})$ complexity. Similarly, Arjevani et al. [8] gave lower bounds showing pth-order methods ($p \geq 2$) cannot beat this rate. While Newton's method is powerful, its $\mathcal{O}(d^3)$ cost is prohibitive for deep learning [14]. By contrast, Hessian-vector products can be computed as efficiently as gradients [62].

Recently, He et al. [37] proposed NEOLITHIC, a nearly optimal first-order method with compression, but higher-order information was not considered.

These developments raise a key question:

Can one design a method that combines communication compression, error feedback, normalization, and practical higher-order momentum for nonconvex distributed optimization, with convergence guarantees?

In this paper, we answer this question positively.

2. Preliminaries

Problem formulation. We consider the distributed nonconvex stochastic optimization problem:

$$\min_{x\in\mathbb{R}^d} \quad f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \quad f_i(x) := \mathbb{E}_{\xi_i\sim\mathcal{D}_i}\left[f_i(x;\xi_i)\right], \quad \text{for} \quad i = 1,2,\dots,n. \tag{1}$$

Here, n is the number of clients, $x \in \mathbb{R}^d$ represents the parameters of the model we aim to train, and $f_i(x)$ is the loss of model parameterized by the vector x on the data $\mathcal{D}_i$ privately known by client i.

The goal is to find an ε-approximate stationary point, i.e. a point x such that $\mathbb{E}\left[\|\nabla f(x)\|\right] \leq \varepsilon$. The expectation is taken with respect to the randomness of the stochastic gradient oracle and the internal randomness of the algorithm.

Assumptions. We impose standard assumptions on objective functions and compression operators for analyzing first-order optimization algorithms.

Assumption 1 *The function $f : \mathbb{R}^d \to \mathbb{R}$ is bounded from below, i.e., $f^{\inf} = \inf_{x\in\mathbb{R}^d} f(x) > -\infty$. Furthermore, each $f_i : \mathbb{R}^d \to \mathbb{R}^d$ is L-smooth if there exists $L > 0$ such that*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\,\|x-y\|\,, \quad \forall x,y \in \mathbb{R}^d.$$

Assumption 2 *The local stochastic gradient $\nabla f_i(x;\xi)$ at client i is an unbiased estimator of $\nabla f_i(x)$ with bounded variance if it satisfies*

$$\mathbb{E}\left[\nabla f_i(x;\xi_i)\right] = \nabla f_i(x), \quad \textit{and} \quad \mathbb{E}\left[\|\nabla f_i(x;\xi_i) - \nabla f(x)\|^2\right] \leq \sigma_g^2, \quad \forall x \in \mathbb{R}^d.$$

Furthermore, the local stochastic Hessian $\nabla^2 f_i(x;\xi)$ at client i is an unbiased estimator of $\nabla^2 f_i(x)$ with bounded variance if it satisfies

$$\mathbb{E}\left[\nabla^2 f_i(x;\xi_i)\right] = \nabla^2 f_i(x), \quad \textit{and} \quad \mathbb{E}\left[\left\|\nabla^2 f_i(x;\xi_i) - \nabla^2 f(x)\right\|^2\right] \leq \sigma_h^2, \quad \forall x \in \mathbb{R}^d.$$

Assumption 3 (Contractive compression) *A biased but possibly randomized compressor $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is α-contractive with its sample $\xi_i \sim \mathcal{D}_i$ if there exists $\alpha \in (0,1]$ such that*

$$\mathbb{E}\left[\|\mathcal{C}(v) - v\|^2\right] \leq (1-\alpha)\,\|v\|^2\,, \quad \forall v \in \mathbb{R}^d.$$

Table 1: A theoretical comparison of error feedback methods using contractive compressors for distributed optimization in a heterogeneous setting.

Method	Work	Complexity
EF14	Seide et al. [74]	$\mathcal{O}\left(\varepsilon^{-4}\right)$
Choco-SGD	Koloskova et al. [48]	$\mathcal{O}\left(\varepsilon^{-4}\right)$
EF21-SGD	Fatkhullin et al. [26]	$\mathcal{O}\left(\varepsilon^{-4}\right)$
EF21-SGDM	Fatkhullin et al. [27]	$\mathcal{O}\left(\varepsilon^{-4}\right)$
‖EF21-SGDM‖	Khirirat et al. [45]	$\mathcal{O}\left(\varepsilon^{-4}\right)$
‖EF21-SGDM-HES‖	**This work**	$\tilde{\mathcal{O}}\left(\varepsilon^{-3}\right)$

3. New method and upper bounds

We propose a distributed algorithm, ‖EF21-SGDM-HES‖ (Algorithm 1), which combines error feedback, normalization, and Hessian-corrected momentum. Unlike most error feedback methods that require knowledge of problem parameters to tune stepsizes, our algorithm is parameter-agnostic: it uses $\mathcal{O}(1)$ batch size per iteration and a time-varying learning rate depending only on the iteration count, not on L or the functional gap $f(x^0) - f^{\inf}$.

Previous work includes EF21 [71], which guarantees convergence with any contractive compressor without restrictive assumptions, and EF21-SGDM [27], which incorporates local stochastic gradients and first-order momentum for nonconvex problems.

We also study ‖EF21-SGDM‖, a normalized variant of EF21-SGDM from [45], originally analyzed for generalized-smooth nonconvex optimization. Normalization was shown to significantly stabilize error feedback in that setting. Here, we establish convergence under the standard smooth Assumption 1, while removing stepsize dependence on L, making the method parameter-agnostic and practical for training neural networks.

Let $\Delta_0 = f(x^0) - f^{\inf}$ and $\mathbb{E}\left[V_0\right] = \Delta_0 + \frac{2\gamma_0\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}$.

Theorem 1 (Convergence of ‖EF21-SGDM‖) *Let function f, functions $\{f_i\}_{i=1}^n$ and stochastic gradients satisfy Assumptions 1 and 2. Let the set of compressors satisfy Assumption 3. Denote through $\tilde{x}_T$ a random point equal to x_t with probability $\frac{\gamma_t}{\sum_{t=0}^{T-1}\gamma_t}$, $t = 0, \ldots, T-1$. Then the iterates $\{x_t\}_{t=0}^{T-1}$ of ‖EF21-SGDM‖ satisfy*

$$\mathbb{E}\left[\|\nabla f(\tilde{x}_T)\|\right] \leq \frac{V_0}{\gamma_0 T^{1/4}} + \frac{2D_1 L \log T}{T^{1/4}} + \frac{2\sigma_g}{\sqrt{n}}\frac{\log T}{T^{1/4}} + \frac{1}{T^{1/4}}\gamma_0\left(L + \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\overline{L}\right) + \frac{1}{T^{1/4}}\frac{64\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_g.$$

It follows from Theorem 1 that ‖EF21-SGDM‖ has a convergence rate of $\mathcal{O}\left(\frac{\log T}{T^{1/4}}\right)$. Therefore, the complexity of the algorithm is $T = \tilde{\mathcal{O}}\left(\varepsilon^{-4}\right)$.

In Theorem below we state the convergence result for the main algorithm of our paper — ‖EF21-SGDM-HES‖. In contrast to ‖EF21-SGDM‖, the first-order heavy ball momentum is replaced with Hessian-vector product correction in the momentum [83]. Using fast Hessian multiplication [62], Hessian-vector products can be evaluated as efficiently as gradients.

Algorithm 1 Normalized EF21 with Hessian-corrected momentum ||EF21-SGDM-HES||

1: **Input:** Starting point $x_0 \in \mathbb{R}^d$, number of epochs T, constant $\gamma_0 > 0$, initial batchsize $B_{\text{init}} \geq 1$,

2: Set $v_i^0 = g_i^0 = \frac{1}{B_{\text{init}}} \sum_{j=1}^{B_{\text{init}}} \nabla f_i(x^0,\xi_{i,j}^0)$, $i = 1, \ldots, n$

3: **for** $t = 0, \ldots, T-1$ **do**

4: Set $\gamma_t = \gamma_0 \left(\frac{1}{t+1}\right)^{\frac{3}{4}}$ and $\eta_t = \left(\frac{2}{t+2}\right)^{\frac{1}{2}}$ for ||EF21-SGDM||

5: Set $\gamma_t = \gamma_0 \left(\frac{1}{t+1}\right)^{2/3}$ and $\eta_t = \left(\frac{2}{t+2}\right)^{2/3}$ for ||EF21-SGDM-HES||

6: Master computes $x^{t+1} = x^t - \gamma_t \frac{g^t}{\|g^t\|}$

7: Master computes $\hat{x}^{t+1} = q_t x^{t+1} + (1-q_t)x^t$, where $q_t \sim U([0,1])$, only for ||EF21-SGDM-HES||

8: **for** all nodes $i = 1, \ldots, n$ **do**

9: $v_i^{t+1} = (1-\eta_t)\, v_i^t + \eta_t \nabla f_i\left(x^{t+1},\xi^{t+1}\right)$ heavy ball (HB) momentum for ||EF21-SGDM||

10: $v_i^{t+1} = (1-\eta_t)\left(v_i^t + \nabla^2 f_i\left(\hat{x}^{t+1},\hat{\xi}^{t+1}\right)\left(x^{t+1} - x^t\right)\right) + \eta_t \nabla f_i\left(x^{t+1},\xi^{t+1}\right)$ second-order momentum (SOM) for ||EF21-SGDM-HES||

11: Compress $c_i^{t+1} = \mathcal{C}_i^{t+1}\left(v_i^{t+1} - g_i^t\right)$

12: $g_i^{t+1} = g_i^t + c_i^{t+1}$

13: **end for**

14: Master computes $g^{t+1} = \frac{1}{n}\sum_{i=1}^n g_i^{t+1}$ via $g^{t+1} = g^t + \frac{1}{n}\sum_{i=1}^n c_i^{t+1}$

15: **end for**

Theorem 2 (Convergence of ||EF21-SGDM-HES||) *Let function f, functions $\{f_i\}_{i=1}^n$ and stochastic gradients satisfy Assumptions 1 and 2. Let the set of compressors satisfy Assumption 3. Denote through $\tilde{x}_T$ a random point equal to x_t with probability $\frac{\gamma_t}{\sum_{t=0}^{T-1}\gamma_t}$, $t = 0, \ldots, T-1$. Then the iterates $\{x_t\}_{t=0}^{T-1}$ of ||EF21-SGDM-HES|| satisfy*

$$\mathbb{E}\left[\|\nabla f(\tilde{x}_T)\|\right] \leq \frac{\mathbb{E}[V_0]}{\gamma_0 T^{1/3}} + 8D_1\gamma_0\left(\frac{\sigma_n}{\sqrt{n}} + L\right)\frac{\log T}{T^{1/3}} + 2D_2\frac{\sigma_g}{\sqrt{n}}\frac{\log T}{T^{1/3}}$$
$$+ 3\gamma_0\left(\frac{L}{2} + \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}(\sigma_n + \overline{L})\right)\frac{1}{T^{1/3}} + \frac{24\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_g \cdot \frac{1}{T^{1/3}}.$$

It follows from Theorem 2 that ||EF21-SGDM-HES|| has a convergence rate of $\mathcal{O}\left(\frac{\log T}{T^{1/3}}\right)$. Therefore, the complexity of the algorithm is $T = \tilde{\mathcal{O}}\left(\varepsilon^{-3}\right)$, which is better than the complexity $T = \tilde{\mathcal{O}}\left(\varepsilon^{-4}\right)$, of ||EF21-SGDM||. In the next section we establish the lower bounds and show that Algorithm 1 ||EF21-SGDM-HES|| is optimal.

4. Conclusion

This paper addresses the role of higher-order methods in distributed stochastic nonconvex optimization under communication constraints. We propose ||EF21-SGDM-HES||, the first algorithm to combine communication compression with Hessian-based momentum in the nonconvex setting. Our method achieves a nearly optimal convergence rate of $\widetilde{\mathcal{O}}(\varepsilon^{-3})$, improving upon existing first-order methods, and matches the established lower bound for second-order methods.

The algorithm is parameter-agnostic and converges with constant batch sizes. Synthetic experiments confirm that ||EF21-SGDM-HES|| offers improved convergence over its first-order counterpart, despite initial oscillations from Hessian noise. These results highlight the potential of higher-order information in efficient distributed optimization.

Acknowledgements

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST): i) KAUST Baseline Research Scheme, ii) Center of Excellence for Generative AI, under award number 5940, iii) SDAIA-KAUST Center of Excellence in Artificial Intelligence and Data Science.

References

[1] Alekh Agarwal and Leon Bottou. A lower bound for the optimization of finite sums. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 78–86, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/agarwal15.html.

[2] Foivos Alimisis, Peter Davies, and Dan Alistarh. Communication-efficient distributed optimization with quantized preconditioners. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. URL https://proceedings.icml.cc/static/paper_files/icml/2021/1231-[]Paper.pdf.

[3] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6c340f25839e6acdc73414517203f5f0-[]Paper.pdf.

[4] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Sarit Khirirat, Nikola Konstantinov, and Cédric Renggli. The convergence of sparsified gradient methods. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 5977–5987, Red Hook, NY, USA, 2018. Curran Associates Inc.

[5] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cedric Renggli. The convergence of sparsified gradient methods. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/314450613369e0ee72d0da7f6fee773c-[]Paper.pdf.

[6] Zeyuan Allen-Zhu. How to make the gradients small stochastically: Even faster convex and nonconvex sgd. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran

Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/996a7fa078cc36c46d02f9af3bef918b-[]Paper.pdf.

[7] Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1756–1764, Cambridge, MA, USA, 2015. MIT Press.

[8] Yossi Arjevani, Yair Carmon, John C. Duchi, Dylan J. Foster, Ayush Sekhari, and Karthik Sridharan. Second-order information in non-convex stochastic optimization: Power and limitations. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 242–299. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/arjevani20a.html.

[9] Yossi Arjevani, Yair Carmon, John C. Duchi, Dylan J. Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization, 2022. URL https://arxiv.org/abs/1912.02365.

[10] Eric Balkanski and Yaron Singer. Parallelization does not accelerate convex optimization: Adaptivity lower bounds for non-smooth convex minimization, 2019. URL https://arxiv.org/abs/1808.03880.

[11] Debraj Basu, Deepesh Data, Can Karakus, and Suhas Diggavi. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. In *Advances in Neural Information Processing Systems*, volume 32, pages 14668–14679, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/d202ed5bcfa858c15a9f383c3e386ab2-[]Abstract.html.

[12] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 560–569, Stockholm, Sweden, 10–15 Jul 2018. PMLR. URL https://proceedings.mlr.press/v80/bernstein18a.html.

[13] Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *J. Mach. Learn. Res.*, 24(1), January 2023. ISSN 1532-4435.

[14] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/0d3180d672e08b4c5312dcdafdf6ef36-[]Paper.pdf.

[15] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh,

Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-[]Paper.pdf.

[16] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Math. Program.*, 184(1–2):71–120, November 2020. ISSN 0025-5610. doi: 10.1007/s10107-[]019-[]01406-[]y. URL https://doi.org/10.1007/s10107-[]019-[]01406-[]y.

[17] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points ii: first-order methods. *Math. Program.*, 185(1–2):315–355, January 2021. ISSN 0025-5610. doi: 10.1007/s10107-[]019-[]01431-[]x. URL https://doi.org/10.1007/s10107-[]019-[]01431-[]x.

[18] Rixon Crane and Fred Roosta. Dingo: Distributed newton-type method for gradient-norm optimization. In *Advances in Neural Information Processing Systems*, volume 32, pages 9498–9508, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/621c18d1d881ef3f8eb35c6df5ebd55f-[]Abstract.html.

[19] Ashok Cutkosky and Harsh Mehta. Momentum improves normalized SGD. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2260–2268. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/cutkosky20b.html.

[20] Nicolò Dal Fabbro, Subhrakanti Dey, Michele Rossi, and Luca Schenato. A newton-type algorithm for federated learning based on incremental hessian eigenvector sharing. *arXiv preprint arXiv:2202.05800*, 2022. URL https://arxiv.org/abs/2202.05800.

[21] Jeffrey Dean, Gregory S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew W. Senior, Paul A. Tucker, Ke Yang, and A. Ng. Large scale distributed deep networks. In *Neural Information Processing Systems*, 2012.

[22] Yury Demidovich, Grigory Malinovsky, Igor Sokolov, and Peter Richtárik. A guide through the zoo of biased sgd. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

[23] Yury Demidovich, Grigory Malinovsky, Egor Shulgin, and Peter Richtárik. MAST: model-agnostic sparsified training. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sPuLtU32av.

[24] Yury Demidovich, Petr Ostroukhov, Grigory Malinovsky, Samuel Horváth, Martin Takáč, Peter Richtárik, and Eduard Gorbunov. Methods with local steps and random reshuffling for

generally smooth non-convex federated optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TrJ36UfD9P.

[25] Jelena Diakonikolas and Cristóbal Guzmán. Lower bounds for parallel and randomized convex optimization. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

[26] Ilyas Fatkhullin, Igor Sokolov, Eduard Gorbunov, Zhize Li, and Peter Richtárik. Ef21 with bells & whistles: Practical algorithmic extensions of modern error feedback, 2021. URL https://arxiv.org/abs/2110.03294.

[27] Ilyas Fatkhullin, Alexander Tyurin, and Peter Richtárik. Momentum provably improves error feedback! In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1h92PmnKov.

[28] Dylan J. Foster, Ayush Sekhari, Ohad Shamir, Nathan Srebro, Karthik Sridharan, and Blake Woodworth. The complexity of making the gradient small in stochastic convex optimization. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1319–1345. PMLR, 25–28 Jun 2019. URL https://proceedings.mlr.press/v99/foster19b.html.

[29] Yuan Gao, Rustem Islamov, and Sebastian U Stich. EControl: Fast distributed optimization with compression and error control. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=lsvlvWB9vz.

[30] Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.*, 23:2341–2368, 2013. URL https://api.semanticscholar.org/CorpusID:14112046.

[31] Avishek Ghosh, Raj Kumar Maity, Arya Mazumdar, and Kannan Ramchandran. Communication efficient distributed approximate newton method. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2539–2544. IEEE, 2020. doi: 10.1109/ISIT44484.2020.9174216. URL https://ieeexplore.ieee.org/document/9174216.

[32] Eduard Gorbunov, Dmitry Kovalev, Dmitry Makarenko, and Peter Richtárik. Linearly converging error compensated sgd. NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[33] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *ArXiv*, abs/1706.02677, 2017.

[34] Kaja Gruntkowska, Alexander Tyurin, and Peter Richtárik. EF21-p and friends: Improved theoretical communication complexity for distributed optimization with bidirectional compression. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 11761–11807. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/gruntkowska23a.html.

[35] Elad Hazan, Kfir Y. Levy, and Shai Shalev-Shwartz. Beyond convexity: stochastic quasi-convex optimization. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1594–1602, Cambridge, MA, USA, 2015. MIT Press.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[37] Yutong He, Xinmeng Huang, Yiming Chen, Wotao Yin, and Kun Yuan. Lower bounds and accelerated algorithms in distributed stochastic optimization with communication compression, 2025. URL https://arxiv.org/abs/2305.07612.

[38] Samuel Horvath, Chen-Yu Ho, Ludovit Horvath, Atal Narayan Sahu, Marco Canini, and Peter Richtarik. Natural compression for distributed deep learning, 2022. URL https://arxiv.org/abs/1905.10988.

[39] Xinmeng Huang, Yiming Chen, Wotao Yin, and Kun Yuan. Lower bounds and nearly optimal algorithms in distributed learning with communication compression. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 18955–18969. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/77f2d0c271e508278ea13e24cd8773d5-[]Paper-[]Conference.pdf.

[40] Rustem Islamov, Xun Qian, and Peter Richtárik. Distributed second order methods with fast rates and compressed communication. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. URL https://proceedings.mlr.press/v139/islamov21a.html.

[41] Rustem Islamov, Xun Qian, Slavomir Hanzely, Mher Safaryan, and Peter Richtárik. Distributed newton-type methods with communication compression and bernoulli aggregation. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=NekBTCKJ1H.

[42] Peter Kairouz, H. B. McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary B. Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim Y. El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Oluwasanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, R. Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Xiaodong Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14:1–210, 2019.

[43] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3252–3261. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/karimireddy19a.html.

[44] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, 2019.

[45] Sarit Khirirat, Abdurakhmon Sadiev, Artem Riabinin, Eduard Gorbunov, and Peter Richtárik. Error feedback under (l_0,l_1)-smoothness: Normalization and momentum, 2024. URL https://arxiv.org/abs/2410.16871.

[46] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[47] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European Conference on Computer Vision*, 2019.

[48] Anastasia Koloskova, Tao Lin, Sebastian U Stich, and Martin Jaggi. Decentralized deep learning with arbitrary communication compression. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=SkgGCkrKpr.

[49] Jakub Konecný, H. B. McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *ArXiv*, abs/1610.05492, 2016.

[50] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[51] Guanghui Lan, Zhize Li, and Yuyuan Zhou. A unified variance-reduced accelerated gradient method for convex optimization. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. A. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pages 10462–10472, Vancouver, BC, Canada, December 2019.

[52] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. 2023.

[53] Chieh-Yen Lin, Cheng-Hao Tsai, Ching-Pei Lee, and Chih-Jen Lin. Large-scale logistic regression and linear support vector machines using spark. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 519–528. IEEE, 2014. doi: 10.1109/BigData.2014.7004265.

[54] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

[55] H. B. McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 2016.

[56] Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, and Peter Richtarik. ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally! In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15750–15769. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/mishchenko22b.html.

[57] Arkadi Nemirovski and David Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons, 1983.

[58] Yu. E. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983. Translation from Dokl. Akad. Nauk SSSR, 269(3):543–547, 1983.

[59] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 edition, 2014. ISBN 1461346916.

[60] Andrei Panferov, Yury Demidovich, Ahmad Rammal, and Peter Richtárik. Correlated quantization for faster nonconvex distributed optimization, 2024. URL https://arxiv.org/abs/2401.05518.

[61] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[62] Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6 (1):147–160, 01 1994. ISSN 0899-7667. doi: 10.1162/neco.1994.6.1.147. URL https://doi.org/10.1162/neco.1994.6.1.147.

[63] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[64] Constantin Philippenko and Aymeric Dieuleveut. Bidirectional compression in heterogeneous settings for distributed or federated learning with partial participation: tight convergence guarantees, 2022. URL https://arxiv.org/abs/2006.14591.

[65] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[66] Daniel Povey, Xiong Zhang, and Sanjeev Khudanpur. Parallel training of dnns with natural gradient and parameter averaging. *arXiv preprint arXiv:1410.7455*, 2014. Neural and Evolutionary Computing.

[67] Xun Qian, Peter Richtárik, and Tong Zhang. Error compensated distributed sgd can be accelerated. In *Advances in Neural Information Processing Systems*, volume 34, pages 30401–30413, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ff1ced3097ccf17c1e67506cdad9ac95-[]Abstract.html.

[68] Xun Qian, Rustem Islamov, Mher Safaryan, and Peter Richtárik. Basis matters: Better communication-efficient second order methods for federated learning. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022. URL http://proceedings.mlr.press/v151/qian22a.html.

[69] Sashank J. Reddi, Jakub Konečný, Peter Richtárik, Barnabás Póczos, and Alexander J. Smola. Aide: Fast and communication efficient distributed optimization. *CoRR*, abs/1608.06879, 2016. URL https://arxiv.org/abs/1608.06879.

[70] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond, 2019. URL https://arxiv.org/abs/1904.09237.

[71] Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. Ef21: a new, simpler, theoretically better, and practically faster error feedback. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

[72] Fred Roosta, Yang Liu, Peng Xu, and Michael W. Mahoney. Newton-MR: Newton's method without smoothness or convexity. *arXiv preprint arXiv:1810.00303*, 2019. URL https://arxiv.org/abs/1810.00303.

[73] Mher Safaryan, Rustem Islamov, Xun Qian, and Peter Richtárik. Fednl: Making newton-type methods applicable to federated learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022. URL https://proceedings.mlr.press/v162/safaryan22a.html.

[74] 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 *Interspeech*, 2014. URL https://api.semanticscholar.org/CorpusID:2189412.

[75] 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 *Proc. Interspeech*, pages 1058–1062, 2014. URL https://www.isca-[]archive.org/interspeech_2014/seide14_interspeech.html.

[76] Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, volume 32, pages 1000–1008. PMLR, 2014. URL https://proceedings.mlr.press/v32/shamir14.html.

[77] Sebastian U. Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4452–4463, Red Hook, NY, USA, 2018. Curran Associates Inc.

[78] Sebastian U. Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, volume 31, pages 4447–4458, 2018.

[79] Ananda Theertha Suresh, Ziteng Sun, Jae Ro, and Felix Yu. Correlated quantization for distributed mean estimation and optimization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20856–20876. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/suresh22a.html.

[80] Rafał Szlendak, Alexander Tyurin, and Peter Richtárik. Permutation compressors for provably faster distributed nonconvex optimization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=GugZ5DzzAu.

[81] Hanlin Tang, Xiangru Lian, Chen Yu, Tong Zhang, and Ji Liu. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6155–6165. PMLR, 2019. URL https://proceedings.mlr.press/v97/tang19d.html.

[82] Hanlin Tang, Yao Li, Ji Liu, and Ming Yan. Errorcompensatedx: Error compensation for variance reduced algorithms. In *Advances in Neural Information Processing Systems*, volume 34, pages 18102–18113, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/968c9b4f09cbb7d7925f38aea3484111-[]Abstract.html.

[83] Hoang Tran and Ashok Cutkosky. Better SGD using second-order momentum. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=9PNsCQpg-[]Ak.

[84] Shusen Wang, Fred Roosta, Peng Xu, and Michael W. Mahoney. GIANT: Globally improved approximate newton method for distributed optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2332–2342, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/4fc4b7b7a631e47c6a7b20e34947e6ed-[]Paper.pdf.

[85] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, volume 31, pages 1306–1316, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/7af9c8c1ebf6c3b1c4e0e5f3c1f2b1d3-[]Abstract.html.

[86] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: ternary gradients to reduce communication in distributed deep learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1508–1518, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[87] Jiaxiang Wu, Weidong Huang, Junzhou Huang, and Tong Zhang. Error compensated quantized sgd and its applications to large-scale distributed optimization. In *International Conference on*

Machine Learning, 2018. URL https://api.semanticscholar.org/CorpusID:49349763.

[88] Zhewei Yao, Zhaosong Lu, Xiangyu Zhang, and Tong Zhang. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1200–1205. ACM, 2017.

[89] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks, 2017. URL https://arxiv.org/abs/1708.03888.

[90] Yang You, Jing Li, Sashank J. Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv: Learning*, 2019.

[91] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 9793–9803, 2018.

[92] Jiaqi Zhang, Keyou You, and Tamer Başar. Achieving globally superlinear convergence for distributed optimization with adaptive newton method. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 2329–2334. IEEE, 2020. doi: 10.1109/CDC42340.2020.9304321. URL https://doi.org/10.1109/CDC42340.2020.9304321.

[93] Yuchen Zhang and Lin Xiao. DiSCO: Distributed optimization for self-concordant empirical loss. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pages 362–370, Lille, France, 2015. PMLR.

[94] Dongruo Zhou and Quanquan Gu. Lower bounds for smooth nonconvex finite-sum optimization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7574–7583. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/zhou19b.html.

[95] Yong Zhuang, Wei-Sheng Chin, Yu-Chin Juan, and Chih-Jen Lin. Distributed newton methods for regularized logistic regression. In Truong Cao, Ee-Peng Lim, Zhi-Hua Zhou, Tu-Bao Ho, David Cheung, and Hiroshi Motoda, editors, *Advances in Knowledge Discovery and Data Mining*, volume 9077 of *Lecture Notes in Computer Science*, pages 690–703, Cham, 2015. Springer International Publishing. ISBN 978-3-319-18032-8. doi: 10.1007/978-[]3-[]319-[]18032-[]8_54.

Appendix A. Related work

Distributed optimization. The massive amount of data needed for state-of-the-art models has made distributed computing systems a necessity [21]. As data and model sizes continue to grow, single-machine approaches can no longer scale to meet storage and computational requirements. The distributed nature of data collection and processing has led to the emergence of Federated Learning (FL) [49, 55] — a framework in which heterogeneous clients collaboratively train a shared model on diverse, decentralized data, without sharing their raw data, thereby preserving privacy. In FL, devices communicate directly with a central server that coordinates the optimization process. Each device performs local computations on its private dataset and sends results—such as model updates—to the server. The server aggregates these updates, carries out global computations, and distributes the updated model parameters back to the devices. This cycle repeats until the model converges or reaches a satisfactory level of accuracy. Typically, in each round, every client transmits a dense gradient vector, which often contains millions of parameters, which imposes a substantial communication overhead on the network. Techniques capable of diminishing the volume of bits transmitted include: acceleration [51, 58, 59, 88], local training [24, 44, 55, 56, 66], and communication compression, which is investigated in our paper.

Communication compression. The majority of common compression techniques fall into one of two categories: sparsification or quantization. Quantization transforms input vectors from a high-precision domain such as 32-bit values into a reduced set of discrete representations such as 8-bit values. Algorithms that use quantization include SignSGD [12, 75], QSGD [3], TernGrad [86]. Natural compressors were introduced in [38]. Correlated quantizers were studied in [60, 79]. Sparsification techniques minimize communication overhead by transmitting only a selected sparse part of the vector at each step. A common sparsification strategy involves randomly discarding some entries to produce a sparse vector [85]. Another common approach is to transmit only a subset of the largest values in the gradient [78]. Convergence results with sparsification can be found in works [4, 23]. Many examples of biased and unbiased sparsifiers, such as TopK and RandK, as well as many quantizers, are explored in works [13, 22]. Szlendak et al. [80] considered correlated sparsification and suggested a PermK sparsifier.

Error feedback (error compensation). Error feedback mechanisms have been introduced to enhance the convergence of compression algorithms while maintaining communication efficiency. EF14, the earliest version of error feedback, was introduced by Seide et al. [74]. It was later rigorously analyzed in the context of first-order algorithms, both in single-node settings [43, 77] and distributed environments [5, 11, 32, 67, 81, 82, 87]. Building on the foundations of error feedback, EF21, introduced by [71], delivers fast convergence guarantees for distributed gradient methods under any contractive compression scheme without relying on restrictive assumptions such as bounded gradient norms or data heterogeneity. EF21 can be effectively extended to stochastic optimization settings via large mini-batch strategies [26] or momentum-based techniques [27]. Advancing the field even further, EControl, proposed by Gao et al. [29], establishes provably tighter complexity bounds for distributed stochastic optimization over previous error feedback frameworks.

Lower bounds. Lower bounds define the theoretical limits of how well an algorithm or a class of algorithms can perform in optimization. Much of the existing research has focused on deriving such bounds, especially in the context of convex problems, [1, 6, 7, 10, 25, 28, 59].

In the nonconvex setting, [16, 17] introduce the zero-chain model and derive tight complexity bounds for both deterministic and randomized first-order algorithms. [16] establish that for any randomized algorithm here exists a function f with p-th order Lipschitzian derivatives such that algorithm must perform at least $\varepsilon^{-\frac{p+1}{p}}$ oracle queries to locate an ε-stationary point. Arjevani et al. [9], Zhou and Gu [94] subsequently broaden the methodology to encompass both finite-sum and stochastic optimization settings. [8] establish a stochastic oracle complexity lower bound of $\Omega(\varepsilon^{-3})$ for finding an ε-approximate stationary point. Moreover, they demonstrate that this bound remains unimprovable even when employing stochastic p-th order methods for any $p \geq 2$, assuming all derivatives up to order p are Lipschitz. Within distributed stochastic optimization employing communication compression, Philippenko and Dieuleveut [64] establish an algorithm-specific lower bound for strongly convex functions. [39] studies nonconvex distributed scenario, [37] studies nonconvex, convex and strongly convex distributed scenarios.

Normalization. One popular modification of the SGD-type methods is the use of normalized updates [35, 89, 90]. This update method builds on the key idea that in nonconvex problems, unlike convex ones, the size of the gradient often says little about the function value, whereas its direction still points toward the steepest descent. Khirirat et al. [45] show that normalization stabilizes the behavior of error feedback algorithms for minimizing nonconvex functions. Normalization usually demands that the gradient noise be very low or that the algorithm use extremely large batch sizes to ensure convergence. This is because normalization can amplify even tiny errors. Cutkosky and Mehta [19] prove that adding momentum eliminates the need for large batch sizes when optimizing non-convex objectives.

Momentum. Inspired by the heavy-ball [65] and acceleration [58] algorithms in convex optimization, momentum seeks to enhance the convergence rate on non-convex objectives by altering the update rule. Essentially, the update maintains a running average of past gradients, aiming to improve stability and conditioning, thereby enabling better performance compared to standard SGD. Momentum has proven remarkably effective in practice [46]. Although several studies [70, 91] have examined momentum-based methods, none have established meaningful theoretical advantages over SGD. [9] showed that the rate of vanilla SGD is optimal.

Second-order methods. Due to the quadratic scaling of Hessian matrices with respect to the problem dimension—requiring d^2 floating-point values per matrix—the main bottleneck in deploying second-order methods in distributed settings lies in the communication overhead. To mitigate the prohibitive cost of transmitting full Hessians, numerous algorithms such as DiSCO [53, 72, 93, 95], GIANT [69, 76, 84], and DINGO [18, 31] have adopted strategies that leverage Hessian-vector products to encode second-order information more compactly. Parallel to these approaches, a distinct line of research has drawn inspiration from compressed first-order methods to directly apply lossy compression to Hessian matrices. Techniques such as DAN-LA [92], Quantized Newton [2], NewtonLearn [40], FedNL [73], Newton-3PC [41], Basis Learn [68], and IOS [20] significantly reduce communication by lowering the number of bits required to represent the Hessian.

Appendix B. Our contributions

In this paper, we explore the construction of lower bounds for a certain class of algorithms, as well as the development of stochastic estimators for p-th order derivatives. We address one of the important questions in distributed stochastic optimization: whether p-th-order methods offer any advantages over lower-order methods. This work makes the following core contributions:

- **New method.** We propose a new method for stochastic distributed nonconvex optimization, called ||EF21-SGDM-HES|| — Normalized EF21 with Hessian-corrected momentum. To the best of our knowledge, this is the first algorithm in nonconvex case that incorporates communication compression and the second-order momentum. It also exploits a modern error feedback mechanism to mitigate the negative effects of compression and enhance convergence while maintaining communication efficiency. Additionally, it employs normalized updates to stabilize practical performance. The pseudocode, discussion of the method, and convergence guarantees can be found in Section 3.

- **Optimal rate.** We investigate whether higher-order methods can improve the sample and communication complexities. We obtain an affirmative answer. Previous first-order distributed stochastic optimization metods such as ||EF21-SGDM|| by Khirirat et al. [45] and NEOLITHIC by He et al. [37] achieve the rate of $\mathcal{O}\left(\varepsilon^{-4}\right)$. We use a limited access to second-order information employing Hessian-corrected momentum to achieve a better rate of $\tilde{\mathcal{O}}\left(\varepsilon^{-3}\right)$ for ||EF21-SGDM-HES|| (see Section 3). We establish the lower bound for distributed stochastic methods of p-th order, $p \geq 2$, with communication compression on nonconvex problems (see Section **??**). The complexity bound is $\Omega\left(\varepsilon^{-3}\right)$ which implies that ||EF21-SGDM-HES|| is nearly optimal up to the logarithmic factor (see Table 1).

- **No parameter dependence.** The learning rate of ||EF21-SGDM-HES|| depends neither on the smoothness constant L nor on the suboptimality gap $f(x^0) - f^{\inf}$, but only on the iteration count (i.e., it uses a time-varying learning rate, see Line 5 in Algorithm 1). Both of these quantities are rarely known in practice. Error feedback algorithms may require large batch size to converge. ||EF21-SGDM-HES|| converges with the batchsize of $\mathcal{O}(1)$. The parameter-agnostic nature of our algorithm makes it particularly well-suited for real-world problems. Additionally, we incorporate time-varying stepsizes into ||EF21-SGDM|| method from [45] and prove its convergence in the L-smooth case.

- **Assumptions on samples.** In our analysis, we do not impose the standard assumption that individual sample functions are L-smooth, i.e., we do not require their gradients to be Lipschitz continuous. Instead, we relax this condition and only assume that the variance of the stochastic gradients, as well as the variance of the stochastic Hessians, are uniformly bounded. These assumptions are sufficient to ensure the convergence of the proposed methods without relying on strong smoothness conditions.

- **Numerical evaluation.** We conduct a synthetic experimental study to evaluate the performance of two parameter-agnostic optimization methods: ||EF21-SGDM|| and its Hessian-enhanced variant, ||EF21-SGDM-HES||. Using data generated via `scikit-learn` with controlled parameters ($M = 10$ clients, $n = 100$ samples per client, $d = 20$ dimensions, and regularization parameter $\lambda = 4$), we assess convergence behavior under identical initialization and update settings. Our results show that while ||EF21-SGDM-HES|| exhibits greater oscillations due to noisy Hessian approximations, it ultimately achieves superior convergence performance compared to the baseline method.

Appendix C. Technical Lemmas

Basic Facts. For a concave function $f(\cdot)$, $n \in \mathbb{N}$ and $x_1,x_2,\ldots,x_n,y \in \mathbb{R}^d$,

$$\begin{aligned} \langle x, y\rangle &\leq \|x\| \, \|y\| \,, & (2)\\ \|x+y\| &\leq \|x\| + \|y\| \,, & (3)\\ \|x+y\|^2 &\leq 2\|x\|^2 + 2\|y\|^2 \,, \quad \text{and} & (4)\\ f\left(\frac{\sum_{i=1}^n x_i}{n}\right) &\leq \frac{1}{n}\sum_{i=1}^n f(x_i). & (5) \end{aligned}$$

Appendix D. Upper Bounds

Recall that Algorithm 1 updates the iterates $\{x^k\}$ using the gradient update:

$$x^{t+1} = x^t - \gamma_t \frac{g^t}{\|g^t\|},$$

where $g^t = \frac{1}{n}\sum_{i=1}^n g_i^t$, and g_i^{t+1} is the memory vector defined by

$$g_i^{t+1} = g_i^t + \mathcal{C}_i^{t+1}\left(v_i^{t+1} - g_i^t\right). \tag{6}$$

Here, v_i^{t+1} is the momentum vector defined by

$$\begin{aligned} v_i^{t+1} &= (1-\eta_t)\, v_i^t + \eta_t \nabla f_i\left(x^{t+1},\xi^{t+1}\right) \quad \text{and} & (7)\\ v_i^{t+1} &= (1-\eta_t)\left(v_i^t + \nabla^2 f_i\left(\hat{x}^{t+1},\hat{\xi}^{t+1}\right)\left(x^{t+1}-x^t\right)\right) + \eta_t \nabla f_i\left(x^{t+1},\xi^{t+1}\right) & (8) \end{aligned}$$

for ||EF21-SGDM|| and ||EF21-SGDM-HES||, respectively.

$$\mathbb{E}_{\hat{\xi}^{t+1},q_{t+1}}[\nabla^2 f_i(\hat{x}^{t+1},\hat{\xi}^{t+1})(x^{t+1}-x^t)] = \mathbb{E}_{\hat{\xi}^{t+1},q_{t+1}}[\nabla^2 f_i(\hat{x}^{t+1},\hat{\xi}^{t+1})](x^{t+1}-x^t) = \mathbb{E}_{q_{t+1}}[\nabla^2 f_i(\hat{x}^{t+1})](x^{t+1}-x^t) = \int$$

Lyapunov Function: To analyze ||EF21-SGDM|| and ||EF21-SGDM-HES||, we rely on the following Lyapunov function.

$$V_t = \Delta_t + C_{1,t}\mathcal{V}_t + C_{2,t}\mathcal{U}_t,$$

where

$$\mathcal{V}_t = \frac{1}{n}\sum_{i=1}^n \|g_i^t - v_i^t\|, \quad \text{and} \quad \mathcal{U}_t = \frac{1}{n}\sum_{i=1}^n \|v_i^t - \nabla f_i(x^t)\|$$

with the coefficients defined by

$$C_{1,t} = \frac{2\gamma_t}{1-\sqrt{1-\alpha}}, \quad C_{2,t} = \frac{2\gamma_t\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}.$$

Lemma 3 *Let Assumption 1 hold. Then for the iterates $\{x^t\}_{t\geq 0}$ generated by the following gradient update*

$$x^{t+1} = x^t - \gamma_t \frac{g^t}{\|g^t\|} \tag{9}$$

satisfy

$$\Delta_{t+1} + \gamma_t \|\nabla f(x^t)\| \le \Delta_t + 2\gamma_t \|\nabla f(x^t) - g^t\| + \frac{\gamma_t^2 L}{2}, \tag{10}$$

where $\Delta_t := f(x^t) - f^{\inf}$ *for any* $t \ge 0$.

Proof Applying L-smoothness of $f(x)$ (Assumption 1) and the update (9), we have

$$\begin{aligned} f(x^{t+1}) &\le f(x^t) + \langle \nabla f(x^t), x^{t+1} - x^t \rangle + \frac{L}{2}\|x^{t+1} - x^t\|^2 \\ &= f(x^t) - \gamma_t \langle \nabla f(x^t), g^t/\|g^t\| \rangle + \frac{L}{2}\gamma_t^2 \\ &\le f(x^t) - \gamma_t \|g^t\| + \gamma_t \langle \nabla f(x^t) - g^t, g^t/\|g^t\| \rangle + \frac{L}{2}\gamma_t^2 \\ &\overset{(2)}{\le} f(x^t) - \gamma_t \|g^t\| - \gamma_t \left\|\nabla f(x^t) - g^t\right\| + \frac{L}{2}\gamma_t^2 \\ &\overset{(3)}{\le} f(x^t) + \gamma_t \|\nabla f(x^t)\| + 2\gamma_t \|\nabla f(x^t) - g^t\| + \frac{L}{2}\gamma_t^2. \end{aligned}$$

Denoting $\Delta_{t+1} := f(x^{t+1}) - f^{\inf}$ for $t \in \{0, \ldots, T-1\}$, we finish the proof. ■

Lemma 4 *Let Assumptions 1, 2, and 3 hold. Then, the iterates* $\{x^t\}$ *generated by Algorithm 1 satisfy*

$$\mathbb{E}_t[\mathcal{V}_{t+1}] \le \sqrt{1-\alpha}\mathcal{V}_t + \sqrt{1-\alpha}\eta_t \mathcal{U}_t + A_t + \sqrt{1-\alpha}\eta_t \sigma_g,$$

where

$A_t = $ [redacted] *for* ||*EF21-SGDM-HES*||

$\qquad \sqrt{1-\alpha}\eta_t\gamma_t\bar{L}$ *for* ||*EF21-SGDM*||

Here, $\bar{L} = \frac{1}{n}\sum_{i=1}^n L_i$.

Proof Denote $\mathcal{F}_t := \{\mathcal{C}_1^\tau, \ldots, \mathcal{C}_n^\tau\}_{\tau=1}^t$ as sigma-algebra. We have

$$\begin{aligned} \mathbb{E}[\|g^{t+1} - v^{t+1}\| | \mathcal{F}_t] &\overset{(3)}{\le} \frac{1}{n}\sum_{i=1}^n \mathbb{E}[\|g_i^{t+1} - v_i^{t+1}\| | \mathcal{F}_t] \\ &\overset{(6)}{=} \frac{1}{n}\sum_{i=1}^n \mathbb{E}[\|\mathcal{C}_i^{t+1}(v_i^{t+1} - g_i^t) - (v_i^{t+1} - g_i^t)\| | \mathcal{F}_t] \\ &\overset{(5)}{\le} \frac{1}{n}\sum_{i=1}^n \left(\mathbb{E}[\|\mathcal{C}_i^{t+1}(v_i^{t+1} - g_i^t) - (v_i^{t+1} - g_i^t)\|^2 | \mathcal{F}_t]\right)^{1/2} \\ &\overset{\text{A. } 3}{\le} \frac{1}{n}\sum_{i=1}^n \left(\mathbb{E}[(1-\alpha)\|v_i^{t+1} - g_i^t\|^2 | \mathcal{F}_t]\right)^{1/2} \\ &= \frac{\sqrt{1-\alpha}}{n}\sum_{i=1}^n \|v_i^{t+1} - g_i^t\| \\ &\overset{(3)}{\le} \frac{\sqrt{1-\alpha}}{n}\sum_{i=1}^n \|v_i^{t+1} - v_i^t\| + \frac{\sqrt{1-\alpha}}{n}\sum_{i=1}^n \|v_i^t - g_i^t\|. \end{aligned} \tag{11}$$

To complete the upper-bound for $\mathbb{E}[\|g^{t+1}-v^{t+1}\||\mathcal{F}_t]$, we must bound $\left\|v_i^{t+1}-v_i^t\right\|$ for ||EF21-SGDM-HES|| and ||EF21-SGDM||

Case I: ||EF21-SGDM||. From the definition of the Euclidean norm,

$$\begin{aligned}
\|v_i^{t+1}-v_i^t\| &\overset{(7)}{=} \|(1-\eta_t)v_i^t+\eta_t\nabla f_i(x^{t+1},\xi_i^{t+1})-v_i^t\| \\
&= \eta_t\|v_i^t-\nabla f_i(x^{t+1},\xi_i^{t+1})\| \\
&= \eta_t\|v_i^t-\nabla f_i(x^t)+\nabla f_i(x^t)-\nabla f_i(x^{t+1})+\nabla f_i(x^{t+1})-\nabla f_i(x^{t+1},\xi_i^{t+1})\| \\
&\overset{(3)}{\leq} \eta_t\|v_i^t-\nabla f_i(x^t)\|+\eta_t\|\nabla f_i(x^t)-\nabla f_i(x^{t+1})\| \\
&\quad +\eta_t\|\nabla f_i(x^{t+1})-\nabla f_i(x^{t+1},\xi_i^{t+1})\| \\
&\overset{\text{A. 1+(9)}}{\leq} \eta_t\|v_i^t-\nabla f_i(x^t)\|+\eta_t\gamma_tL_i+\eta_t\|\nabla f_i(x^{t+1})-\nabla f_i(x^{t+1},\xi^{t+1})\|.
\end{aligned}$$

By taking the expectation over stochastic gradients $\mathbb{E}_{\xi_i^{t+1}}[\cdot]$, and by Assumption 2,

$$\mathbb{E}_{\xi_i^{t+1}}[\|v_i^{t+1}-v_i^t\|] \leq \eta_t\|v_i^t-\nabla f_i(x^t)\|+\eta_t\gamma_tL_i+\eta_t\sigma_g.$$

Case II: ||EF21-SGDM-HES||. From the definition of v_i^{t+1},

$$\begin{aligned}
v_i^{t+1}-v_i^t &\overset{(8)}{=} (1-\eta_t)(v_i^t+\nabla^2 f_i(\hat{x}^{t+1},\hat{\xi}_i^{t+1})(x^{t+1}-x^t))+\eta_t\nabla f_i(x^{t+1},\xi_i^{t+1})-v_i^t \\
&= \eta_t(\nabla f_i(x^t)-v_i^t)+\eta_t(\nabla f_i(x^{t+1})-\nabla f_i(x^t)) \\
&\quad +(1-\eta_t)(\nabla^2 f_i(\hat{x}^{t+1},\hat{\xi}_i^{t+1})(x^{t+1}-x^t)-\nabla^2 f_i(\hat{x}^{t+1})(x^{t+1}-x^t)) \\
&\quad +(1-\eta_t)(\nabla^2 f_i(\hat{x}^{t+1})(x^{t+1}-x^t))+\eta_t(\nabla f_i(x^{t+1},\xi_i^{t+1})-\nabla f_i(x^{t+1})).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|v_i^{t+1}-v_i^t\| &\overset{(3)}{\leq} \eta_t\|v_i^t-\nabla f_i(x^t)\|+\eta_t\|\nabla f_i(x^{t+1})-\nabla f_i(x^t)\| \\
&\quad +(1-\eta_t)\|\nabla^2 f_i(\hat{x}^{t+1},\hat{\xi}_i^{t+1})-\nabla^2 f_i(\hat{x}^{t+1})\|_{\text{op}}\|x^{t+1}-x^t\| \\
&\quad +(1-\eta_t)\|\nabla^2 f_i(\hat{x}^{t+1})\|_{\text{op}}\|x^{t+1}-x^t\| \\
&\quad +\eta_t\|\nabla f_i(x^{t+1},\xi_i^{t+1})-\nabla f_i(x^{t+1})\| \\
&\overset{\text{A. 1+(9)}}{\leq} \eta_t\|v_i^t-\nabla f_i(x^t)\|+\eta_t\gamma_tL_i \\
&\quad +(1-\eta_t)\|\nabla^2 f_i(\hat{x}^{t+1},\hat{\xi}_i^{t+1})-\nabla^2 f_i(\hat{x}^{t+1})\|_{\text{op}}\|x^{t+1}-x^t\| \\
&\quad +(1-\eta_t)\gamma_t\|\nabla^2 f_i(\hat{x}^{t+1})\|_{\text{op}} \\
&\quad +\eta_t\|\nabla f_i(x^{t+1},\xi_i^{t+1})-\nabla f_i(x^{t+1})\|
\end{aligned}$$

By taking the expectation over stochastic gradients $\mathbb{E}_{\xi_i^{t+1},\hat{\xi}_i^{t+1}}[\cdot]$, and by Assumption 2,

$$\begin{aligned}
\mathbb{E}_{\xi_i^{t+1},\hat{\xi}_i^{t+1}}[\|v_i^{t+1}-v_i^t\|] &\leq \eta_t\|v_i^t-\nabla f_i(x^t)\|+\eta_tL_i\gamma_t+(1-\eta_t)L_i\gamma_t+ \\
&\quad +(1-\eta_t)\gamma_t\sigma_h+\eta_t\sigma_g.
\end{aligned}$$

Finally, plugging the bound for $\mathbb{E}_{\xi_i^{t+1},\hat{\xi}_i^{t+1}}[\|v_i^{t+1}-v_i^t\|]$ of two cases into (11), we obtain the results. ■

Lemma 5 *Let Assumptions 1 and 2 hold. Then, the iterates $\{x^t\}$ generated by Algorithm 1 satisfy*

$$
\begin{aligned}
\mathbb{E}_t[\mathcal{U}_{t+1}] &\le (1-\eta_t)\mathcal{U}_t + (1-\eta_t)\gamma_t\bar{L} + \eta_t\sigma_g, \quad \textit{for } \|\textit{ EF21-SGDM}\| \\
\mathbb{E}_t[\mathcal{U}_{t+1}] &\le (1-\eta_t)\mathcal{U}_t + (1-\eta_t)\gamma_t\left(\sigma_h + 2\bar{L}\right) + \eta_t\sigma_g, \quad \textit{for } \|\textit{ EF21-SGDM-HES}\|
\end{aligned}
$$

where $\bar{L} = \frac{1}{n}\sum_{i=1}^n L_i$.

Proof We begin by proving the bounds for $U_t, \bar{U}_t$ for $\|$EF21-SGDM$\|$.

Case I: $\|$EF21-SGDM$\|$ From the definition of $\mathcal{U}_{t+1}$ and v_i^{t+1},

$$
\begin{aligned}
\mathcal{U}_{t+1} &\overset{(7)}{=} \frac{1}{n}\sum_{i=1}^n \left\|(1-\eta_t)v_i^t + \eta_t\nabla f_i(x^{t+1},\xi_i^{t+1}) - \nabla f_i(x^{t+1})\right\| \\
&= \frac{1}{n}\sum_{i=1}^n \|(1-\eta_t)(v_i^t - \nabla f_i(x^t)) + (1-\eta_t)(\nabla f_i(x^t) - \nabla f_i(x^{t+1})) \\
&\quad + \eta_t\nabla f_i(x^{t+1};\xi_i^{t+1}) - \eta_t\nabla f_i(x^{t+1})\| \\
&\le \frac{1-\eta_t}{n}\sum_{i=1}^n \left\|v_i^t - \nabla f_i(x^t)\right\| + \frac{1-\eta_t}{n}\sum_{i=1}^n \left\|\nabla f_i(x^t) - \nabla f_i(x^{t+1})\right\| \\
&\quad + \frac{\eta_t}{n}\sum_{i=1}^n \left\|\nabla f_i(x^{t+1};\xi_i^{t+1}) - \nabla f_i(x^{t+1})\right\| \\
&\overset{\text{A. 1+(9)}}{\le} \frac{1-\eta_t}{n}\sum_{i=1}^n \left\|v_i^t - \nabla f_i(x^t)\right\| + (1-\eta_t)\bar{L}\gamma_t \\
&\quad + \frac{\eta_t}{n}\sum_{i=1}^n \left\|\nabla f_i(x^{t+1};\xi_i^{t+1}) - \nabla f_i(x^{t+1})\right\|.
\end{aligned}
$$

Therefore,

$$
\mathbb{E}_t[\mathcal{U}_{t+1}] \overset{\text{A. 2}}{\le} (1-\eta_t)\mathcal{U}_t + (1-\eta_t)\bar{L}\gamma_t + \eta_t\sigma_g.
$$

Case II: $\|$ EF21-SGDM-HES$\|$ From the definition of v_i^{t+1},

$$
\begin{aligned}
\|v_i^{t+1} - \nabla f_i(x^{t+1})\| &= \|(1-\eta_t)(v_i^t - \nabla f_i(x^t)) + (1-\eta_t)\hat{S}_{i,t+1} + \eta_t e_{i,t+1}\| \\
&\le (1-\eta_t)\|v_i^t - \nabla f_i(x^t)\| + (1-\eta_t)\|\hat{S}_{i,t+1}\| + \eta_t\|e_{i,t+1}\|,
\end{aligned}
$$

where

$$
\begin{aligned}
\hat{S}_{i,t+1} &= \nabla^2 f_i(\hat{x}^{t+1},\hat{\xi}_i^{t+1})(x^{t+1} - x^t) - \nabla f_i(x^{t+1}) + \nabla f_i(x^t), \quad \text{and} \\
e_{i,t+1} &= \nabla f_i(x^{t+1},\xi_i^{t+1}) - \nabla f_i(x^{t+1}).
\end{aligned}
$$

Therefore, from the definition of $\mathcal{U}_{t+1}$,

$$\begin{aligned}
\mathbb{E}_t[\mathcal{U}_{t+1}] &\leq (1-\eta_t)\mathbb{E}_t[\mathcal{U}_t] + \frac{1-\eta_t}{n}\sum_{i=1}^n \mathbb{E}_t\left[\left\|\hat{S}_{i,t+1}\right\|\right] + \eta_t \frac{1}{n}\sum_{i=1}^n \mathbb{E}_t[\|e_{i,t+1}\|] \\
&= (1-\eta_t)\mathcal{U}_t + \frac{1-\eta_t}{n}\sum_{i=1}^n \mathbb{E}_t\left[\left\|\hat{S}_{i,t+1}\right\|\right] + \eta_t \frac{1}{n}\sum_{i=1}^n \mathbb{E}_t[\|e_{i,t+1}\|] \\
&\overset{\text{A. 2}}{\leq} (1-\eta_t)\mathcal{U}_t + \frac{1-\eta_t}{n}\sum_{i=1}^n \mathbb{E}_t\left[\left\|\hat{S}_{i,t+1}\right\|\right] + \eta_t\sigma_g \\
&\overset{(5)}{\leq} (1-\eta_t)\mathcal{U}_t + \frac{1-\eta_t}{n}\sum_{i=1}^n \sqrt{\mathbb{E}_t\left[\left\|\hat{S}_{i,t+1}\right\|^2\right]} + \eta_t\sigma_g.
\end{aligned}$$

To complete the bound for $\mathbb{E}_t[\mathcal{U}_{t+1}]$, we must bound $\mathbb{E}_t\left[\left\|\hat{S}_{i,t+1}\right\|^2\right]$.

$$\begin{aligned}
\mathbb{E}_t\left[\left\|\hat{S}_{i,t+1}\right\|^2\right] &= \mathbb{E}_{\hat{\xi}_i^{t+1}}\left[\left\|\nabla^2 f_i(\hat{x}^{j+1},\hat{\xi}_i^{t+1})(x^{t+1}-x^t) - (\nabla f_i(x^{t+1}) - \nabla f_i(x^t))\right\|^2\right] \\
&\overset{\text{A. 2}}{=} \mathbb{E}_{\hat{\xi}_i^{t+1}}\left[\left\|(\nabla^2 f_i(\hat{x}^{t+1},\hat{\xi}_i^{t+1}) - \nabla^2 f_i(\hat{x}^{t+1}))(x^{t+1}-x^t)\right\|^2\right] \\
&\quad + \left\|\nabla^2 f_i(\hat{x}^{t+1})(x^{t+1}-x^t) - (\nabla f_i(x^{t+1}) - \nabla f(x^t))\right\|^2 \\
&\overset{(4)}{\leq} \mathbb{E}_{\hat{\xi}_i^{t+1}}\left[\left\|(\nabla^2 f_i(\hat{x}^{t+1},\hat{\xi}_i^{t+1}) - \nabla^2 f_i(\hat{x}^{t+1}))(x^{t+1}-x^t)\right\|^2\right] \\
&\quad +2\left\|\nabla^2 f(\hat{x}^{t+1})\right\|_{\text{op}}^2 \left\|x^{t+1}-x^t\right\|^2 + 2\left\|\nabla f(x^{t+1}) - \nabla f(x^t)\right\|^2 \\
&\overset{\text{A. 2+(9)}}{\leq} \gamma_t^2\left(\sigma_h^2 + 4L_i^2\right).
\end{aligned}$$

Finally, by plugging the upper-bound of $\mathbb{E}_t\left[\left\|\hat{S}_{i,t+1}\right\|^2\right]$ into the upper-bound of $\mathbb{E}_t[\mathcal{U}_{t+1}]$, and by the fact that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a,b \geq 0$,

$$\begin{aligned}
\mathbb{E}_t[\mathcal{U}_{t+1}] &\leq (1-\eta_t)\mathcal{U}_t + \frac{1-\eta_t}{n}\sum_{i=1}^n \sqrt{\gamma_t^2\left(\sigma_h^2 + 4L_i^2\right)} + \eta_t\sigma_g \\
&\leq (1-\eta_t)\mathcal{U}_t + (1-\eta_t)\gamma_t(\sigma_h + 2\bar{L}) + \eta_t\sigma_g
\end{aligned}$$

■

Lemma 6 (Descent inequality for ||EF21-SGDM||) *Let Assumptions 1, 2, and 3 hold. Then, the iterates $\{x^t\}$ generated by ||EF21-SGDM|| with the decreasing stepsizes γ_t satisfy*

$$\mathbb{E}[V_{t+1}] \leq \mathbb{E}[V_t] - \gamma_t\mathbb{E}\left[\|\nabla f(x^t)\|\right] + 2\gamma_t\mathbb{E}\left[\left\|\nabla f(x^t) - v^t\right\|\right] + \frac{L}{2}\gamma_t^2 + \gamma_t^2 \cdot B_1 + \eta_t\gamma_t \cdot B_2,$$

where $V_t = f(x^t) - f^{\inf} + C_{1,t}\mathcal{V}_t + C_{2,t}\mathcal{U}_t$, $\mathcal{V}_t = \frac{1}{n}\sum_{i=1}^n \|g_i^t - v_i^t\|$, $\mathcal{U}_t = \frac{1}{n}\sum_{i=1}^n \|v_i^t - \nabla f_i(x^t)\|$, $C_{1,t} = \frac{2\gamma_t}{1-\sqrt{1-\alpha}}$, $C_{2,t} = \frac{2\gamma_t\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}$, $B_1 = \frac{2\sqrt{1-\alpha}\bar{L}}{1-\sqrt{1-\alpha}}$ *and* $B_2 = \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_g$.

Proof We derive the result in the following steps.

Step 1) Bound $\Delta_t = f(x^t) - f^{\inf}$ by L-smoothness of $f(x)$. From Theorem 3,

$$\Delta_{t+1} \le \Delta_t - \gamma_t \|\nabla f(x^t)\| + 2\gamma_t \|\nabla f(x^t) - g^t\| + \frac{L}{2}\gamma_t^2.$$

Since

$$\begin{aligned} \left\|\nabla f(x^t) - g^t\right\| &\overset{(3)}{\le} \left\|\nabla f(x^t) - v^t\right\| + \left\|v^t - g^t\right\| \\ &\overset{(3)}{\le} \left\|\nabla f(x^t) - v^t\right\| + \mathcal{V}_t, \end{aligned}$$

where $\mathcal{U}_t = \frac{1}{n}\sum_{i=1}^n \left\|v_i^t - \nabla f_i(x^t)\right\|$ and $\mathcal{V}_t = \frac{1}{n}\sum_{i=1}^n \left\|g_i^t - v_i^t\right\|$, we obtain

$$\Delta_{t+1} \le \Delta_t - \gamma_t \|\nabla f(x^t)\| + 2\gamma_t \left\|\nabla f(x^t) - v^t\right\| + 2\gamma_t \mathcal{V}_t + \frac{L}{2}\gamma_t^2. \tag{12}$$

Step 2) Bound $V_t = \Delta_t + C_{1,t}\mathcal{V}_t + C_{2,t}\mathcal{U}_t$ for some $C_{1,t}, C_{2,t} > 0$. Denote $V_t = \Delta_t + C_{1,t}\mathcal{V}_t + C_{2,t}\mathcal{U}_t$ with $C_{1,t} = \frac{2\gamma_t}{1-\sqrt{1-\alpha}}$ and $C_{2,t} = \frac{2\gamma_t\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}$. Therefore,

$$\begin{aligned} V_{t+1} &= \Delta_t + C_{1,t}\mathcal{V}_{t+1} + C_{2,t}\mathcal{U}_{t+1} \\ &\overset{(12)}{\le} \Delta_t - \gamma_t \|\nabla f(x^t)\| + \frac{L}{2}\gamma_t^2 \\ &\quad + 2\gamma_t \left\|\nabla f(x^t) - v^t\right\| + 2\gamma_t \mathcal{V}_t + C_{1,t+1}\mathcal{V}_{t+1} + C_{2,t+1}\mathcal{U}_{t+1}. \end{aligned}$$

By taking the expectation $\mathbb{E}_t[\cdot]$,

$$\begin{aligned} \mathbb{E}_t[V_{t+1}] &\le \Delta_t - \gamma_t \|\nabla f(x^t)\| + 2\gamma_t \left\|\nabla f(x^t) - v^t\right\| \\ &\quad + 2\gamma_t \mathcal{V}_t + C_{1,t+1}\mathbb{E}_t[\mathcal{V}_{t+1}] + C_{2,t+1}\mathbb{E}_t[\mathcal{U}_{t+1}] + \frac{L}{2}\gamma_t^2 \\ &\overset{\text{Theorem 4}}{\le} \Delta_t - \gamma_t \|\nabla f(x^t)\| + 2\gamma_t \left\|\nabla f(x^t) - v^t\right\| \\ &\quad + (2\gamma_t + C_{1,t+1}\sqrt{1-\alpha})\mathcal{V}_t + (C_{1,t+1}\sqrt{1-\alpha}\eta_t)\mathcal{U}_t + C_{2,t+1}\mathbb{E}_t[\mathcal{U}_{t+1}] \\ &\quad + \frac{L}{2}\gamma_t^2 + C_{1,t+1}(\sqrt{1-\alpha}\eta_t\gamma_t\bar{L} + \sqrt{1-\alpha}\eta_t\sigma_g) \\ &\overset{\text{Theorem 5}}{\le} \Delta_t - \gamma_t \|\nabla f(x^t)\| + 2\gamma_t \left\|\nabla f(x^t) - v^t\right\| \\ &\quad + (2\gamma_t + C_{1,t+1}\sqrt{1-\alpha})\mathcal{V}_t + (C_{1,t+1}\sqrt{1-\alpha}\eta_t + C_{2,t+1}(1-\eta_t))\mathcal{U}_t \\ &\quad + \frac{L}{2}\gamma_t^2 + C_{1,t+1}(\sqrt{1-\alpha}\eta_t\gamma_t\bar{L} + \sqrt{1-\alpha}\eta_t\sigma_g) + C_{2,t+1}\left((1-\eta_t)\gamma_t\bar{L} + \eta_t\sigma_g\right). \end{aligned}$$

If $\gamma_{t+1} \le \gamma_t$, then we can prove that $C_{1,t+1} \le C_{1,t}$, that $C_{2,t+1} \le C_{2,t}$, that

$$\begin{aligned} 2\gamma_t + C_{1,t+1}\sqrt{1-\alpha} &\le 2\gamma_t + C_{1,t}\sqrt{1-\alpha} \\ &= C_{1,t}, \end{aligned}$$

and that

$$\begin{aligned} C_{1,t+1}\sqrt{1-\alpha}\eta_t + C_{2,t+1}(1-\eta_t) &\le C_{1,t}\sqrt{1-\alpha}\eta_t + C_{2,t}(1-\eta_t) \\ &= C_{2,t}. \end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{E}_t\left[V_{t+1}\right] &\leq V_t - \gamma_t\|\nabla f(x^t)\| + 2\gamma_t\left\|\nabla f(x^t) - v^t\right\| \\ &\quad + \frac{L}{2}\gamma_t^2 + C_{1,t}(\sqrt{1-\alpha}\eta_t\gamma_t\bar{L} + \sqrt{1-\alpha}\eta_t\sigma_g) + C_{2,t}\left((1-\eta_t)\gamma_t\bar{L} + \eta_t\sigma_g\right).\end{aligned}$$

By taking the full expectation,

$$\begin{aligned}\mathbb{E}\left[V_{t+1}\right] &\leq \mathbb{E}\left[V_t\right] - \gamma_t\mathbb{E}\left[\|\nabla f(x^t)\|\right] + 2\gamma_t\mathbb{E}\left[\left\|\nabla f(x^t) - v^t\right\|\right] \\ &\quad + \frac{L}{2}\gamma_t^2 + \gamma_t^2\cdot B_1 + 2\eta_t\gamma_t\cdot B_2,\end{aligned}$$

where $B_1 = \frac{2\sqrt{1-\alpha}\bar{L}}{1-\sqrt{1-\alpha}}$ and $B_2 = \frac{2\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_g$. ■

Lemma 7 (Descent inequality for ||EF21-SGDM-HES||) *Let Assumptions 1, 2, and 3 hold. Then, the iterates $\{x^t\}$ generated by ||EF21-SGDM-HES|| with the decreasing stepsizes γ_t satisfy*

$$\begin{aligned}\mathbb{E}\left[V_{t+1}\right] &\leq \mathbb{E}\left[V_t\right] - \gamma_t\mathbb{E}\left[\|\nabla f(x^t)\|\right] + 2\gamma_t\mathbb{E}\left[\left\|\nabla f(x^t) - v^t\right\|\right] \\ &\quad + \frac{L}{2}\gamma_t^2 + \gamma_t^2\cdot\hat{B}_1 + \eta_t\gamma_t\cdot\hat{B}_2 + (1-\eta_t)\gamma_t^2\cdot\hat{B}_3,\end{aligned}$$

where $V_t = \Delta_t + C_{1,t}\mathcal{V}_t + C_{2,t}\mathcal{U}_t$, $\mathcal{V}_t = \frac{1}{n}\sum_{i=1}^n\|g_i^t - v_i^t\|$, $\mathcal{U}_t = \frac{1}{n}\sum_{i=1}^n\|v_i^t - \nabla f_i(x^t)\|$, $C_{1,t} = \frac{2\gamma_t}{1-\sqrt{1-\alpha}}$, $C_{2,t} = \frac{2\gamma_t\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}$, $\hat{B}_1 = \frac{3\sqrt{1-\alpha}\bar{L}}{1-\sqrt{1-\alpha}}$, $\hat{B}_2 = \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_g$, and $\hat{B}_3 = \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_h$.

Proof Denote $V_t = \Delta_t + C_{1,t}\mathcal{V}_t + C_{2,t}\mathcal{U}_t$ with $C_{1,t} = \frac{2\gamma_t}{1-\sqrt{1-\alpha}}$ and $C_{2,t} = \frac{2\gamma_t\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}$. By following the proof arguments in Theorem 6, we can prove that

$$\begin{aligned}\mathbb{E}_t\left[V_{t+1}\right] &\leq \Delta_t - \gamma_t\|\nabla f(x^t)\| + 2\gamma_t\left\|\nabla f(x^t) - v^t\right\| \\ &\quad + 2\gamma_t\mathcal{V}_t + C_{1,t+1}\mathbb{E}_t\left[\mathcal{V}_{t+1}\right] + C_{2,t+1}\mathbb{E}_t\left[\mathcal{U}_{t+1}\right] + \frac{L}{2}\gamma_t^2.\end{aligned}$$

Therefore, from the upper-bounds for $\mathbb{E}_t\left[\mathcal{V}_{t+1}\right]$ and $\mathbb{E}_t\left[\mathcal{U}_{t+1}\right]$ for ||EF21-SGDM-HES||,

$$\begin{aligned}\mathbb{E}_t\left[V_{t+1}\right] &\overset{\text{Theorem 4}}{\leq} \Delta_t - \gamma_t\|\nabla f(x^t)\| + 2\gamma_t\left\|\nabla f(x^t) - v^t\right\| \\ &\quad + (2\gamma_t + C_{1,t+1}\sqrt{1-\alpha})\mathcal{V}_t + C_{1,t+1}\sqrt{1-\alpha}\eta_t\mathcal{U}_t + C_{2,t+1}\mathbb{E}_t\left[\mathcal{U}_{t+1}\right] \\ &\quad + \frac{L}{2}\gamma_t^2 + C_{1,t+1}\sqrt{1-\alpha}\gamma_t\bar{L} + C_{1,t+1}\sqrt{1-\alpha}(1-\eta_t)\gamma_t\sigma_h + C_{1,t+1}\sqrt{1-\alpha}\eta_t\sigma_g \\ &\overset{\text{Theorem 5}}{\leq} \Delta_t - \gamma_t\|\nabla f(x^t)\| + 2\gamma_t\left\|\nabla f(x^t) - v^t\right\| \\ &\quad + (2\gamma_t + C_{1,t+1}\sqrt{1-\alpha})\mathcal{V}_t + \left(C_{1,t+1}\sqrt{1-\alpha}\eta_t + C_{2,t+1}(1-\eta_t)\right)\mathcal{U}_t \\ &\quad + \frac{L}{2}\gamma_t^2 + C_{1,t+1}\sqrt{1-\alpha}\gamma_t\bar{L} + C_{1,t+1}\sqrt{1-\alpha}(1-\eta_t)\gamma_t\sigma_h + C_{1,t+1}\sqrt{1-\alpha}\eta_t\sigma_g \\ &\quad + C_{2,t+1}\left((1-\eta_t)\gamma_t\left(\sigma_h + 2\bar{L}\right) + \eta_t\sigma_g\right).\end{aligned}$$

If $\gamma_{t+1} \leq \gamma_t$, then we can prove that $C_{1,t+1} \leq C_{1,t}$, that $C_{2,t+1} \leq C_{2,t}$, that

$$\begin{aligned}2\gamma_t + C_{1,t+1}\sqrt{1-\alpha} &\leq 2\gamma_t + C_{1,t}\sqrt{1-\alpha} \\ &= C_{1,t},\end{aligned}$$

and that

$$\begin{aligned} C_{1,t+1}\sqrt{1-\alpha}\eta_t + C_{2,t+1}(1-\eta_t) &\leq C_{1,t}\sqrt{1-\alpha}\eta_t + C_{2,t}(1-\eta_t) \\ &= C_{2,t}. \end{aligned}$$

Therefore,

$$\begin{aligned} \mathbb{E}_t\left[V_{t+1}\right] &\leq V_t - \gamma_t\|\nabla f(x^t)\| + 2\gamma_t\left\|\nabla f(x^t) - v^t\right\| \\ &\quad + \frac{L}{2}\gamma_t^2 + C_{1,t+1}\sqrt{1-\alpha}\gamma_t\bar{L} + C_{1,t+1}\sqrt{1-\alpha}(1-\eta_t)\gamma_t\sigma_h + C_{1,t+1}\sqrt{1-\alpha}\eta_t\sigma_g \\ &\quad + C_{2,t+1}\left((1-\eta_t)\gamma_t\left(\sigma_h + 2\bar{L}\right) + \eta_t\sigma_g\right). \end{aligned}$$

By taking the full expectation, and by the fact that $\eta_t > 0$,

$$\begin{aligned} \mathbb{E}\left[V_{t+1}\right] &\leq \mathbb{E}\left[V_t\right] - \gamma_t\mathbb{E}\left[\|\nabla f(x^t)\|\right] + 2\gamma_t\mathbb{E}\left[\left\|\nabla f(x^t) - v^t\right\|\right] \\ &\quad + \frac{L}{2}\gamma_t^2 + \gamma_t^2\cdot\hat{B}_1 + \eta_t\gamma_t\cdot\hat{B}_2 + (1-\eta_t)\gamma_t^2\cdot\hat{B}_3, \end{aligned}$$

where $\hat{B}_1 = \frac{3\sqrt{1-\alpha}\bar{L}}{1-\sqrt{1-\alpha}}$, $\hat{B}_2 = \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_g$, and $\hat{B}_3 = \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_h$. ■

Lemma 8 *Let Assumptions 1 and 2 hold. Then, the iterates $\{x^t\}$ generated by Algorithm 1 with $\eta_0 = 1$ satisfy*

$$\left\|v^{t+1} - \nabla f(x^{t+1})\right\| \leq \frac{\sigma_g}{\sqrt{n}}\left(\sum_{j=0}^{t}\left(\prod_{\tau=j+1}^{t}(1-\eta_\tau)^2\right)\eta_j^2\right)^{1/2} + B_t,$$

where

$$B_t = \begin{cases} L\sum_{l=0}^{t}\left(\prod_{j=l}^{t}(1-\eta_j)\right)\gamma_l & \textit{for } \|\text{EF21-SGDM}\|; \\ 4\left(\sum_{j=1}^{t}\left(\prod_{\tau=j}^{t}(1-\eta_\tau)^2\right)\gamma_j^2\right)^{1/2}\left(\frac{\sigma_h}{\sqrt{n}} + L\right) & \textit{for } \|\text{EF21-SGDM-HES}\|. \end{cases}$$

Proof Denote $\hat{e}_t = v^t - \nabla f(x^t)$.

First, we bound $\left\|v^{t+1} - \nabla f(x^{t+1})\right\|$ for $\|$EF21-SGDM$\|$:

$$\hat{e}_{t+1} = (1-\eta_t)\hat{e}_t + (1-\eta_t)\hat{S}_{t+1} + \eta_t e_{t+1},$$

where $\hat{S}_{t+1} = \nabla f(x^t) - \nabla f(x^{t+1})$, $e_{t+1} = \frac{1}{n}\sum_{i=1}^{n} e_{i,t+1}$, and $e_{i,t+1} = \nabla f_i(x^{t+1}, \xi_i^{t+1}) - \nabla f_i(x^{t+1})$.

Next, by recursively applying the equation for $\hat{e}_{t+1}$,

$$\begin{aligned} \hat{e}_{t+1} &= \left(\prod_{j=0}^{t}(1-\eta_j)\right)\hat{e}_0 + \sum_{l=0}^{t}\left(\prod_{j=l+1}^{t}(1-\eta_j)\right)\cdot(1-\eta_l)\hat{S}_{l+1} \\ &\quad + \sum_{l=0}^{t}\left(\prod_{j=l+1}^{t}(1-\eta_j)\right)\cdot\eta_l e_{l+1}. \end{aligned}$$

If $\eta_0 = 1$, then

$$\begin{aligned}\hat{e}_{t+1} &= \sum_{l=0}^{t}\left(\prod_{j=l+1}^{t}(1-\eta_j)\right)\cdot(1-\eta_l)\hat{S}_{l+1} + \sum_{l=0}^{t}\left(\prod_{j=l+1}^{t}(1-\eta_j)\right)\cdot\eta_l e_{l+1} \\ &= \sum_{l=0}^{t}\left(\prod_{j=l}^{t}(1-\eta_j)\right)\cdot\hat{S}_{l+1} + \sum_{l=0}^{t}\left(\prod_{j=l+1}^{t}(1-\eta_j)\right)\cdot\eta_l e_{l+1}\end{aligned}$$

Therefore,

$$\|\hat{e}_{t+1}\| \overset{(3)}{\le} \sum_{l=0}^{t}\left(\prod_{j=l}^{t}(1-\eta_j)\right)\left\|\hat{S}_{l+1}\right\| + \left\|\sum_{l=0}^{t}\left(\prod_{j=l+1}^{t}(1-\eta_j)\right)\cdot\eta_l e_{l+1}\right\|.$$

Since

$$\left\|\hat{S}_{l+1}\right\| \le L\left\|x^{l+1}-x^{l}\right\| \le L\gamma_l,$$

we obtain

$$\|\hat{e}_{t+1}\| \overset{(3)}{\le} L\sum_{l=0}^{t}\left(\prod_{j=l}^{t}(1-\eta_j)\right)\gamma_l + \left\|\sum_{l=0}^{t}\left(\prod_{j=l+1}^{t}(1-\eta_j)\right)\cdot\eta_l e_{l+1}\right\|.$$

By taking the expectation,

$$\mathbb{E}\left[\|\hat{e}_{t+1}\|\right] \overset{(5)}{\le} L\sum_{l=0}^{t}\left(\prod_{j=l}^{t}(1-\eta_j)\right)\gamma_l + \sqrt{\mathbb{E}\left[\left\|\sum_{l=0}^{t}\left(\prod_{j=l+1}^{t}(1-\eta_j)\right)\cdot\eta_l e_{l+1}\right\|^2\right]}.$$

From Assumption 2, we can prove that $\mathbb{E}\left[e_l\right]=0$, $\mathbb{E}\left[\|e_l\|^2\right]=\sigma_g^2/n$, and $\mathbb{E}\left[\langle e_l, e_i\rangle\right]=0$ for $l\neq j$. Thus,

$$\mathbb{E}\left[\|\hat{e}_{t+1}\|\right] \le L\sum_{l=0}^{t}\left(\prod_{j=l}^{t}(1-\eta_j)\right)\gamma_l + \frac{\sigma_g}{\sqrt{n}}\left(\sum_{l=0}^{t}\prod_{j=l+1}^{t}(1-\eta_j)^2\eta_l^2\right)^{1/2}.$$

Second, we bound $\left\|v^{t+1}-\nabla f(x^{t+1})\right\|$ for ||EF21-SGDM-HES||:

$$\begin{aligned}\hat{e}_{t+1} &= (1-\eta_t)\hat{e}_t + (1-\eta_t)\hat{S}_{t+1} + \eta_t e_{t+1} \\ &= \prod_{\tau=0}^{t}(1-\eta_\tau)\hat{e}_0 + \sum_{j=0}^{t}\left(\prod_{\tau=j+1}^{t}(1-\eta_\tau)\right)\hat{S}_{j+1} + \sum_{j=0}^{t}\left(\prod_{\tau=j}^{t}(1-\eta_\tau)\right)\eta_j e_{j+1},\end{aligned}$$

where $\hat{S}_{t+1}=\frac{1}{n}\sum_{i=1}^{n}\left(\nabla^2 f_i(\hat{x}^{t+1},\hat{\xi}_i^{t+1})(x^{t+1}-x^t)-(\nabla f_i(x^{t+1})-\nabla f_i(x^t))\right)$, $e_{t+1}=\frac{1}{n}\sum_{i=1}^{n}e_{i,t+1}$, and $e_{i,t+1}=\nabla f_i(x^{t+1},\xi_i^{t+1})-\nabla f_i(x^{t+1})$.

If $\eta_0 = 1$, then by taking the Euclidean norm and the expectation,

$$\begin{aligned}
\mathbb{E}\left[\|\hat{e}_{t+1}\|\right] &\overset{(3)}{\leq} \mathbb{E}\left[\left\|\sum_{j=0}^{t}\left(\prod_{\tau=j}^{t}(1-\eta_\tau)\right)\hat{S}_{j+1}\right\|\right] + \mathbb{E}\left[\left\|\sum_{j=0}^{t}\left(\prod_{\tau=j}^{t}(1-\eta_\tau)\right)\eta_j e_{j+1}\right\|\right] \\
&\overset{(5)}{\leq} \left(\mathbb{E}\left\|\sum_{j=0}^{t}\prod_{\tau=j}^{t}(1-\eta_\tau)\hat{S}_{j+1}\right\|^2\right)^{1/2} + \left(\mathbb{E}\left\|\sum_{j=0}^{t}\prod_{\tau=j+1}^{t}(1-\eta_\tau)\eta_j e_{j+1}\right\|^2\right)^{1/2}.
\end{aligned}$$

From Assumption 2, we can prove that $\mathbb{E}\left[e_l\right] = 0$, $\mathbb{E}\left[\|e_l\|^2\right] = \sigma_g^2/n$, and $\mathbb{E}\left[\langle e_l, e_i\rangle\right] = 0$ for $l \neq j$. Thus,

$$\mathbb{E}\left[\|\hat{e}_{t+1}\|\right] \leq \left(\mathbb{E}\left\|\sum_{j=0}^{t}\prod_{\tau=j}^{t}(1-\eta_\tau)\hat{S}_{j+1}\right\|^2\right)^{1/2} + \frac{\sigma_g}{\sqrt{n}}\left(\sum_{j=0}^{t}\prod_{\tau=j+1}^{t}(1-\eta_\tau)^2\eta_j^2\right)^{1/2}.$$

Next, from Assumption 2, we can show that $\mathbb{E}\left[\langle \hat{S}_l, \hat{S}_j\rangle\right] = 0$ for $l \neq j$, and that

$$\mathbb{E}\left[\|\hat{e}_{t+1}\|\right] \leq \left(\sum_{j=0}^{t}\prod_{\tau=j}^{t}(1-\eta_\tau)^2\mathbb{E}\left\|\hat{S}_{j+1}\right\|^2\right)^{1/2} + \frac{\sigma_g}{\sqrt{n}}\left(\sum_{j=0}^{t}\prod_{\tau=j+1}^{t}(1-\eta_\tau)^2\eta_j^2\right)^{1/2}.$$

Since

$$\begin{aligned}
\mathbb{E}[\|\hat{S}_{j+1}\|^2] &= \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}\left(\nabla^2 f_i(\hat{x}^{j+1}, \xi_i^{j+1})(x^{j+1}-x^j) - (\nabla f_i(x^{j+1}) - \nabla f_i(x^j))\right)\right\|^2\right] \\
&\leq \frac{1}{n^2}\sum_{i=1}^{n}\left(\mathbb{E}\left[\left\|(\nabla^2 f_i(\hat{x}^{j+1}, \xi_i^{j+1}) - \nabla^2 f_i(\hat{x}^{j+1}))(x^{j+1}-x^j)\right\|^2\right]\right) \\
&\quad + \mathbb{E}\left[\left\|\nabla^2 f(\hat{x}^{j+1})(x^{j+1}-x^j) - (\nabla f(x^{j+1}) - \nabla f(x^j))\right\|^2\right] \\
&\leq \frac{1}{n^2}\sum_{i=1}^{n}\left(\mathbb{E}\left\|(\nabla^2 f_i(\hat{x}^{j+1}, \hat{\xi}_i^{j+1}) - \nabla^2 f_i(x^{j+1}))(x^{j+1}-x^j)\right\|^2\right) \\
&\quad + 2\mathbb{E}\left\|\nabla^2 f(\hat{x}^{j+1})\right\|_{\text{op}}^2\left\|x^{j+1}-x^j\right\|^2 + 2\mathbb{E}\left\|\nabla f(x^{j+1}) - \nabla f(x^j)\right\|^2 \\
&\leq \gamma_j^2\left(\frac{1}{n^2}\sum_{i=1}^{n}\sigma_h^2 + 4L^2\right) \\
&\leq 4\left(\frac{\sigma_h^2}{n} + L^2\right)\gamma_j^2,
\end{aligned}$$

we obtain:

$$\begin{aligned}\mathbb{E}\left[\|\hat{e}_{t+1}\|\right] \quad \leq \quad & 4\left(\sum_{j=1}^{t}\left(\prod_{\tau=j}^{t}(1-\eta_\tau)^2\right)\gamma_j^2\right)^{1/2}\left(\frac{\sigma_h}{\sqrt{n}}+L\right) \\ & +\frac{\sigma_g}{\sqrt{n}}\left(\sum_{j=0}^{t}\left(\prod_{\tau=j+1}^{t}(1-\eta_\tau)^2\right)\eta_j^2\right)^{1/2}.\end{aligned}$$

■

D.1. Proof of Theorem 1

From Theorem 6,

$$\begin{aligned}& \gamma_t\mathbb{E}\left[\|\nabla f(x^t)\|\right] \\ & \leq \mathbb{E}\left[V_t\right]-\mathbb{E}\left[V_{t+1}\right]+2\gamma_t\mathbb{E}\left[\left\|\nabla f(x^t)-v^t\right\|\right]+\frac{L}{2}\gamma_t^2+\gamma_t^2\cdot B_1+\eta_t\gamma_t\cdot B_2 \\ & \overset{\text{Theorem 8}}{\leq} \mathbb{E}\left[V_t\right]-\mathbb{E}\left[V_{t+1}\right]+2\gamma_t\cdot L\sum_{l=0}^{t}\left(\prod_{j=l}^{t}(1-\eta_j)\right)\gamma_l+2\gamma_t\cdot\frac{\sigma_g}{\sqrt{n}}\left(\sum_{l=0}^{t}\prod_{j=l+1}^{t}(1-\eta_j)^2\eta_l^2\right)^{1/2} \\ & \quad +\frac{L}{2}\gamma_t^2+\gamma_t^2\cdot B_1+\eta_t\gamma_t\cdot B_2,\end{aligned}$$

where $B_1=\frac{2\sqrt{1-\alpha}\bar{L}}{1-\sqrt{1-\alpha}}$ and $B_2=\frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_g$. Therefore,

$$\begin{aligned}& \frac{\sum_{t=0}^{T-1}\gamma_t\mathbb{E}\left[\|\nabla f(x^t)\|\right]}{\sum_{t=0}^{T-1}\gamma_t} \\ & \leq \frac{\mathbb{E}\left[V_0\right]+2\sum_{t=0}^{T-1}\gamma_t\cdot L\sum_{l=0}^{t}\left(\prod_{j=l}^{t}(1-\eta_j)\right)\gamma_l+2\frac{\sigma_g}{\sqrt{n}}\sum_{t=0}^{T-1}\gamma_t\left(\sum_{l=0}^{t}\prod_{j=l+1}^{t}(1-\eta_j)^2\eta_l^2\right)^{1/2}}{\sum_{t=0}^{T-1}\gamma_t} \\ & \quad +\frac{\frac{L}{2}\sum_{t=0}^{T-1}\gamma_t^2+B_1\sum_{t=0}^{T-1}\gamma_t^2+B_2\sum_{t=0}^{T-1}\eta_t\gamma_t}{\sum_{t=0}^{T-1}\gamma_t}.\end{aligned}$$

If $\gamma_t=\gamma_0\left(\frac{1}{t+1}\right)^{3/4}$ and $\eta_t=\left(\frac{2}{t+2}\right)^{1/2}$, then we can prove the following:

1. $\sum_{j=0}^{t-1}\left(\prod_{\tau=j}^{t-1}(1-\eta_\tau)\right)\gamma_j\leq\sum_{j=0}^{t-1}\left(\prod_{\tau=j+1}^{t-1}(1-\eta_\tau)\right)\gamma_j\leq C_1\gamma_t/\eta_t$

2. $\left(\sum_{j=0}^{t-1}\left(\prod_{\tau=j}^{t-1}(1-\eta_\tau)^2\right)\eta_j^2\right)^{1/2} \leq \left(\sum_{j=0}^{t-1}\left(\prod_{\tau=j}^{t-1}(1-\eta_\tau)\right)\eta_j^2\right)^{1/2} \leq (C_2\eta_t^2/\eta_t)^{1/2} = C_2\sqrt{\eta_t}$

3. $\sum_{t=0}^{T-1}\gamma_t^2/\eta_t=\sum_{t=0}^{T-1}\gamma_0\left(\frac{1}{t+1}\right)^{3/2}\cdot\left(\frac{2}{t+2}\right)^{-1/2}\leq\sum_{t=0}^{T-1}\gamma_0\frac{1}{t+1}\leq\int_1^T\gamma_0\frac{1}{z}dz=\gamma_0\log T$

4. $\sum_{t=0}^{T-1}\gamma_t\sqrt{\eta_t} = \sum_{t=0}^{T-1}\gamma_0\left(\frac{1}{t+1}\right)^{3/4}\left(\frac{2}{t+2}\right)^{1/4} \le \sum_{t=0}^{T-1}\gamma_0\frac{2}{t+2} \le \int_1^{T+1}\gamma_0\frac{2}{z}dz = 2\gamma_0\log T$

5. $\sum_{t=0}^{T-1}\gamma_t^2 = \sum_{t=0}^{T-1}\gamma_0^2\left(\frac{1}{t+1}\right)^{3/2} \le \gamma_0^2\int_1^T\frac{1}{z^{3/2}}dt = -2\gamma_0^2\left.\frac{1}{z^{1/2}}\right|_1^T = (1-\frac{1}{\sqrt{T}})\cdot 2\gamma_0^2 \le 2\gamma_0^2$

6. $\sum_{t=0}^{T-1}\gamma_t\eta_t = \gamma_0\sum_{t=0}^{T-1}\left(\frac{1}{t+1}\right)^{3/4}\left(\frac{2}{t+2}\right)^{1/2} \le \gamma_0\sum_{t=0}^{T-1}\left(\frac{2}{t+2}\right)^{5/4}$
$\le \gamma_0\int_2^{T+1}\frac{2^{5/4}}{z^{5/4}}dz = 4\gamma_0\cdot 4\left(-\left.\frac{1}{z^{1/4}}\right|_2^{T+1}\right) \le 16\gamma_0$

7. $\sum_{t=0}^{T-1}\gamma_t = \sum_{t=0}^{T-1}\gamma_0\left(\frac{1}{t+1}\right)^{3/4} \ge \gamma_0 T\cdot\frac{1}{T^{3/4}} = \gamma_0 T^{1/4}$

By applying these facts, we finally have

$$
\begin{aligned}
\frac{\sum_{t=0}^{T-1}\gamma_t\mathbb{E}\|\nabla f(x^t)\|}{\sum_{t=0}^{T-1}\gamma_t} &\le \frac{V_0}{\gamma_0 T^{1/4}} + \frac{2LC_1\log T}{\gamma_0 T^{1/4}} + \frac{2C_2\sigma_g}{\sqrt{n}}\cdot\frac{\gamma_0\log T}{\gamma_0 T^{1/4}} \\
&\quad + \frac{1}{2}\left(L + \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}L\right)\cdot\frac{2\gamma_0^2}{\gamma_0 T^{1/4}} + \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\cdot\frac{8\sigma_g\gamma_0}{\gamma_0 T^{1/4}} \\
&= \frac{V_0}{\gamma_0 T^{1/4}} + \frac{2LC_1\log T}{T^{1/4}} + \frac{2\sigma_g}{\sqrt{n}}\cdot\frac{\log T}{T^{1/4}} \\
&\quad + \frac{1}{T^{1/4}}\gamma_0\left(L + \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}L\right) + \frac{1}{T^{1/4}}\cdot\frac{64\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_g.
\end{aligned}
$$

D.2. Proof of Theorem 2

From Theorem 7 we obtain

$$
\begin{aligned}
\gamma_t\mathbb{E}\left[\|\nabla f(x^t)\|\right] &\le \mathbb{E}[V_t] - \mathbb{E}[V_{t+1}] + 2\gamma_t\mathbb{E}\left[\left\|\nabla f(x^t) - v^t\right\|\right] \\
&\quad + \frac{L}{2}\gamma_t^2 + \gamma_t^2\cdot\hat{B}_1 + \eta_t\gamma_t\cdot\hat{B}_2 + (1-\eta_t)\gamma_t^2\cdot\hat{B}_3 \\
&\overset{\text{Theorem 8}}{\le} \mathbb{E}[V_t] - \mathbb{E}[V_{t+1}] + 2\gamma_t\frac{\sigma_g}{\sqrt{n}}\left(\sum_{j=0}^{t}\left(\prod_{\tau=j+1}^{t}(1-\eta_\tau)^2\right)\eta_j^2\right)^{1/2} \\
&\quad + 2\gamma_t\cdot 4\left(\sum_{j=1}^{t}\left(\prod_{\tau=j}^{t}(1-\eta_\tau)^2\right)\gamma_j^2\right)^{1/2}\left(\frac{\sigma_h}{\sqrt{n}} + L\right) \\
&\quad + \frac{L}{2}\gamma_t^2 + \gamma_t^2\cdot\hat{B}_1 + \eta_t\gamma_t\cdot\hat{B}_2 + (1-\eta_t)\gamma_t^2\cdot\hat{B}_3,
\end{aligned}
$$

where $\hat{B}_1 = \frac{3\sqrt{1-\alpha}\bar{L}}{1-\sqrt{1-\alpha}}$, $\hat{B}_2 = \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_g$, and $\hat{B}_3 = \frac{4\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}\sigma_h$. Therefore,

$$
\begin{aligned}
\frac{\sum_{t=0}^{T-1}\gamma_t \mathbb{E}\left[\left\|\nabla f(x^t)\right\|\right]}{\sum_{t=0}^{T-1}\gamma_t} \quad \leq \quad & \frac{V_0 + 2\sum_{t=0}^{T-1}\gamma_t \frac{\sigma_g}{\sqrt{n}}\left(\sum_{j=0}^{t}\left(\prod_{\tau=j+1}^{t}(1-\eta_\tau)^2\right)\eta_j^2\right)^{1/2}}{\sum_{t=0}^{T-1}\gamma_t} \\
&+\frac{2\sum_{t=0}^{T-1}\gamma_t \cdot 4\left(\sum_{j=1}^{t}\left(\prod_{\tau=j}^{t}(1-\eta_\tau)^2\right)\gamma_j^2\right)^{1/2}\left(\frac{\sigma_h}{\sqrt{n}}+L\right)}{\sum_{t=0}^{T-1}\gamma_t} \\
&+\frac{\frac{L}{2}\sum_{t=0}^{T-1}\gamma_t^2 + \hat{B}_1\sum_{t=0}^{T-1}\gamma_t^2 + \hat{B}_2\sum_{t=0}^{T-1}\eta_t\gamma_t + \hat{B}_3\sum_{t=0}^{T-1}(1-\eta_t)\gamma_t^2}{\sum_{t=0}^{T-1}\gamma_t}.
\end{aligned}
$$

If $\gamma_t = \left(\frac{1}{1+t}\right)^{2/3}\gamma_0$ with $\gamma_0 > 0$, and $\eta_t = \left(\frac{2}{t+2}\right)^{2/3}$, then we can prove the following:

1. $\left(\sum_{j=1}^{t}\left(\prod_{\tau=j}^{t}(1-\eta_\tau)^2\right)\eta_j^2\right)^{1/2} \leq \left(\sum_{j=1}^{t}\left(\prod_{\tau=j}^{t}(1-\eta_\tau)\right)\eta_j^2\right)^{1/2} \leq C_1\sqrt{\eta_t}$
2. $\left(\sum_{j=0}^{t}\left(\prod_{\tau=j+1}^{t}(1-\eta_\tau)^2\right)\eta_j^2\right)^{1/2} \leq \left(\sum_{j=0}^{t}\left(\prod_{\tau=j+1}^{t}(1-\eta_\tau)\right)\eta_j^2\right)^{1/2} \leq C_2\sqrt{\eta_t}$
3. $\sum_{t=0}^{T-1}\frac{\gamma_t^2}{\sqrt{\eta_t}} = \gamma_0^2\sum_{t=0}^{T-1}\left(\frac{1}{1+t}\right)^{4/3}\left(\frac{t+2}{2}\right)^{1/3} \leq \gamma_0^2\sum_{t=0}^{T-1}\frac{1}{1+t} \leq \gamma_0^2\int_1^T \frac{1}{z}\,dz = \gamma_0^2\log T$
4. $\sum_{t=0}^{T-1}\gamma_t\sqrt{\eta_t} = \gamma_0\sum_{t=0}^{T-1}\left(\frac{1}{1+t}\right)^{2/3}\left(\frac{2}{t+2}\right)^{1/3} \leq \gamma_0\sum_{t=0}^{T-1}\left(\frac{8}{(t+2)(1+t)^2}\right)^{1/3} \leq \gamma_0\int_2^{T+1}\frac{1}{z}\,dz = \gamma_0\log T$
5. $\sum_{t=0}^{T-1}\gamma_t^2 = \gamma_0^2\sum_{t=0}^{T-1}\left(\frac{1}{1+t}\right)^{4/3} \leq \gamma_0^2\int_1^T\frac{1}{z^{4/3}}\,dz \leq 3\gamma_0^2$
6. $\sum_{t=0}^{T-1}\gamma_t\eta_t = \gamma_0\sum_{t=0}^{T-1}\left(\frac{1}{1+t}\right)^{2/3}\left(\frac{2}{t+2}\right)^{2/3} \leq \gamma_0\int_2^{T+1}\frac{1}{(z+2)^{4/3}}\,dz \leq 6\gamma_0$
7. $\sum_{t=0}^{T-1}\gamma_t \geq \gamma_0 T^{1/3} = \gamma_0 T^{1/3}$

By applying these facts, we finally obtain

$$
\begin{aligned}
\frac{\sum_{t=0}^{T-1}\gamma_t\mathbb{E}\|\nabla f(x^t)\|}{\sum_{t=0}^{T-1}\gamma_t} \quad \leq \quad & \frac{V_0}{\gamma_0 T^{1/3}} + 8C_1\gamma_0\left(\frac{\sigma_h}{\sqrt{n}}+L\right)\frac{\log T}{T^{1/3}} \\
&+2C_2\frac{\sigma_g}{\sqrt{n}}\frac{\log T}{T^{1/3}} + 3\gamma_0\left(\frac{L}{2}+\frac{4\sqrt{1-\alpha^2}}{1-\sqrt{1-\alpha}}\cdot(\sigma_h+\bar{L})\right)\frac{1}{T^{1/3}} \\
&+\frac{24\sqrt{1-\alpha^2}}{1-\sqrt{1-\alpha}}\sigma_g\cdot\frac{1}{T^{1/3}}.
\end{aligned}
$$

Appendix E. Numerical Experiments

We consider the logistic regression problem with non-convex regularizer:

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{Mn}\sum_{m=1}^{M}\sum_{i=1}^{n}\log\left(1+e^{-b_{m,i}\langle a_{m,i},x\rangle}\right)+\lambda r(x)\right\},\ \text{where}\ \ r(x):=\sum_{j=1}^{d}\frac{x_j^2}{1+x_j^2}. \tag{13}$$

Let $\lambda > 0$ be a regularization parameter, $a_{m,i} \in \mathbb{R}^d$ denote the input features, and $b_{m,i} \in \{-1,+1\}$ the corresponding binary labels. Here, M represents the number of clients and n the number of data points per client. In our experiments, we set $\lambda = 4$, and synthetically generate the data using the `scikit-learn` library [63], with $M = 10$, $n = 10$, and $d = 20$.

We evaluate and compare the performance of ||EF21-SGDM-HES|| and ||EF21-SGDM||. Both methods are parameter-agnostic, requiring no problem-specific tuning. For all experiments, we initialize the algorithm with x^0 as a vector of ones, and the control variates are set to the gradients of the local objective functions: $g_i^0 = v_i^0 = \nabla f_i(x^0)$. The initial learning rate is set to $\gamma_0 = 1$.

Figure 1: Comparison between ||EF21-SGDM-HES|| and ||EF21-SGDM|| on problem (13) with $\lambda = 4$.

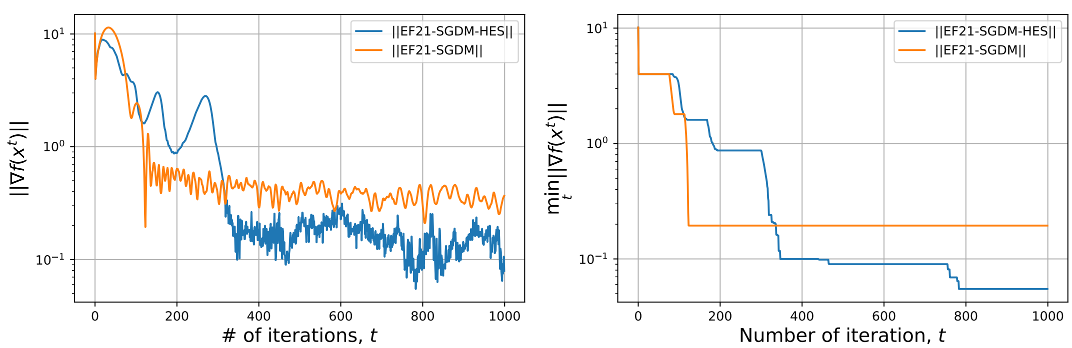

Figure 1 presents two plots that illustrate the convergence behavior of the proposed methods. The plot on the left depicts the convergence of the gradient norm. Initially, the ||EF21-SGDM|| method exhibits slightly faster convergence compared to ||EF21-SGDM-HES||. This can be attributed to the higher oscillations in ||EF21-SGDM-HES||, which are caused by noise in the Hessian approximations. However, once the optimization process stabilizes, ||EF21-SGDM|| demonstrates superior performance relative to its counterpart that does not incorporate Hessian information.

Since our theoretical analysis pertains to the minimum gradient norm achieved over the course of the iterations—rather than the gradient norms along the entire optimization trajectory—we illustrate this metric in the plot on the right. As shown, the convergence is initially slower, but significantly improves after the first few iterations.

Consistent with the results shown in Figure 1, the method || ||EF21-SGDM-HES|| || slightly outperforms || ||EF21-SGDM|| ||, although it exhibits greater oscillatory behavior due to the inclusion of Hessian information.

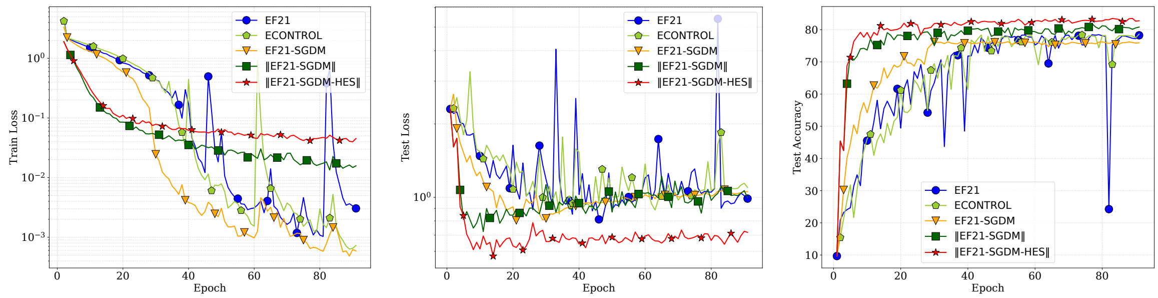

Figure 2: Performance comparison of ‖EF21-SGDM-HES‖ against baseline methods on the CIFAR-10 dataset using ResNet-18. The plots display (a) training loss, (b) test loss, and (c) test accuracy, all as a function of epochs.

Appendix F. Deep Learning experiments

Datasets, Hardware, and Implementation. To evaluate the performance of the proposed methods in training Deep Neural Networks (DNNs), we utilized the ResNet-18 architecture [36]. ResNet-18 is a prominent model for image classification, and its architecture is also frequently adapted for tasks such as feature extraction in image segmentation, object detection, image embedding, and image captioning. Our experiments involved training all layers of the ResNet-18 model, corresponding to an optimization problem with $d = 11{,}173{,}962$ parameters.

All implementations were developed in PyTorch [61], and experiments were conducted on the CIFAR-10 dataset [50]. Numerical evaluations were performed on a server-grade machine running Ubuntu 18.04 (Linux Kernel v5.4.0). This system was equipped with dual 16-core 3.3 GHz Intel Xeon processors (totaling 32 cores) and four NVIDIA A100 GPUs, each with 40GB of memory.

To simulate a federated learning environment, we adopted a data distribution strategy inspired by Gao et al. [29]. Specifically, 50% of the CIFAR-10 dataset was allocated to 10 clients based on class labels, such that data points with the i-th label (for $i \in \{0, \ldots, 9\}$) were assigned to client $i+1$. The remaining 50% of the dataset was distributed randomly and uniformly among the clients. Subsequently, each client's local data was partitioned into a training set (90%) and a test set (10%). This partitioning scheme introduces data heterogeneity, a common characteristic of federated settings. For communication compression, we employed the Top-K sparsifier, retaining 10% of the coordinates (i.e., $K/d = 0.1$).

In our PyTorch implementation of ‖EF21-SGDM-HES‖, hessian-vector products are computed efficiently using automatic differentiation, leveraging the identity $\nabla^2 f(x,z)v = \nabla_x \langle \nabla_x f(x,z), v\rangle$. PyTorch's capability to backpropagate through the differentiation process itself facilitates a straightforward implementation. To optimize computational cost, we approximate the Hessian-vector product term $\nabla^2 f_i(\hat{x}^{t+1}, \hat{\xi}^{t+1})(x^{t+1} - x^t)$. Instead of using a separate $\hat{x}^{t+1}$ and sampling new $\hat{\xi}^{t+1}$, we set $\hat{x}^{t+1} = x^t$ and reuse the same stochastic batch ξ^{t+1} that is employed for computing the stochastic gradient $\nabla f_i(x^{t+1}, \xi^{t+1})$. This practical simplification avoids two additional backpropagation calculations per iteration.

Baselines and Hyperparameter Tuning. We benchmark the proposed ‖EF21-SGDM-HES‖ against several state-of-the-art error feedback methods: EF21-SGD [26], EF21-SGDM [27], ‖EF21-SGDM‖ [45], and EControl [29].

For the baseline methods EF21-SGDM and EControl, the momentum parameter η was set to 0.1, following the recommendations in Fatkhullin et al. [27] and Gao et al. [29], respectively. For our proposed ||EF21-SGDM-HES|| and the ||EF21-SGDM|| baseline, we explored both constant η values from the set $\{0.01, 0.1, 0.2\}$ and theoretically motivated decreasing schedules. Specifically, for ||EF21-SGDM||, we tested $\eta_e = (2/e+2)^{0.5}$, and for ||EF21-SGDM-HES||, $\eta_e = (2/e+2)^{0.67}$, where e denotes the epoch counter. These decreasing schedules are inspired by Algorithm 1 and the convergence analyses in Theorems 1 and 2. In practice, to prevent η from diminishing too rapidly, we update it on a per-epoch basis rather than per-iteration.

Regarding stepsizes, EF21-SGD, EF21-SGDM, and EControl utilized a constant stepsize scheme, with values tuned from the set $\{1.0, 0.1, 0.05, 0.01, 0.005\}$. For ||EF21-SGDM|| and our proposed ||EF21-SGDM-HES||, corresponding to each selected η schedule (constant or decreasing), we evaluated both constant stepsizes $\gamma_e \equiv \gamma \in \{1.0, 0.1, 0.05, 0.01\}$ and epoch-dependent decreasing schedules: $\gamma_e = \gamma_0(1/e+1)^{0.75}$ for ||EF21-SGDM||, and $\gamma_e = \gamma_0(1/e+1)^{0.67}$ for ||EF21-SGDM-HES||. Here, e is the epoch counter, and the initial learning rate γ_0 for the decreasing schedules was tuned from a similar range as the constant stepsizes.

All methods were trained for a fixed budget of 90 epochs. Since the per-iteration communication cost is identical for all compared algorithms, the total number of epochs serves as a direct proxy for the total bits communicated. Upon completion of all experimental runs, the optimal hyperparameters (stepsize γ_e and momentum parameter η_e) for each method were selected based on the best validation accuracy achieved and observed stable convergence behavior. A summary of the selected tuned hyperparameters is provided in Table 2, and the best-achieved accuracy metrics for each method are detailed in Table 3.

Table 2: Summary of the tuned hyper-parameters.

Method	**Learning rate γ**	**Momentum η**
EF21-SGD	1.0	—
EF21-SGDM	0.1	0.1
EControl	1.0	0.1
||EF21-SGDM||	0.1	$(2/e+2)^{0.5}$ †
||EF21-SGDM-HES||	0.1	$(2/e+2)^{0.67}$ †

† e is the epoch's index .

Table 3: Best performance metrics achieved by each method when training ResNet-18 on the CIFAR-10 dataset. The best results are highlighted in **bold**.

Method	**Best Validation Accuracy (%)**	**Corresponding Test Accuracy (%)**	**Epoch of Best Validation Accuracy**
EF21-SGD	80.30	78.30	77
EF21-SGDM	77.74	76.26	33
EControl	80.36	78.63	90
||EF21-SGDM||	82.22	81.88	76
||EF21-SGDM-HES||	**84.18**	**83.42**	87

Performance Comparison As indicated in Table 3, ||EF21-SGDM-HES|| achieves the best accuracy on both the validation and test sets. Its advantage over other baselines is notably illustrated in Figure 2. This figure demonstrates that ||EF21-SGDM-HES|| clearly attains the highest test accuracy among all methods, and this superiority is consistently and stably maintained throughout the learning procedure. In particular, it also achieves the lowest test loss (see Figure 2b).

It is worth noting that, while ||EF21-SGDM-HES|| demonstrates the best performance, other methods such as ||EF21-SGDM|| also show notable improvements over earlier approaches. In contrast, EF21-SGD and EControl exhibit more unstable convergence trajectories, characterized by significant fluctuations in their performance metrics.