SPAM: Stochastic Proximal Point Method with Momentum Variance Reduction for Non-convex Cross-Device Federated Learning

author names withheld

Under Review for OPT 2024

Abstract

Cross-device training is a crucial subfield of federated learning, where the number of clients can reach into the billions. Standard approaches and local methods are prone to issues such as client drift and insensitivity to data similarities. We propose a novel algorithm (SPAM) for cross-device federated learning with non-convex and non-smooth losses. We provide sharp analysis under second-order (Hessian) similarity, a condition satisfied by a variety of machine learning problems in practice. Additionally, we extend our results to the partial participation setting, where a cohort of selected clients communicate with the server at each communication round.

1. Introduction

Federated learning (FL) [11, 18, 26] is a machine learning approach where multiple entities, known as *clients*, work together to solve a machine learning problem under the guidance of a *central server*. Each client's raw data stays on their local devices and is not shared or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning goal [11].

This paper focuses on *cross-device* training [13], where the clients are mobile or IoT devices. To model such a large number of clients, we study the following stochastic optimization problem:

$$\min_{x\in\mathbb{R}^d} f(x), \quad \text{where} \quad f(x) := \mathrm{E}_{\xi\sim\mathcal{D}}\left[f_\xi(x)\right], \tag{1}$$

where f_ξ may be non-convex. Here, we do not have access to the full function f, nor its gradient. This reflects the cross-device setting, where the number of clients is extremely large (e.g., billions of mobile phones), so each client participates in the training process only a few times or maybe even once. Therefore, we cannot expect full participation to obtain the exact gradient.

Instead of the exact function or gradient values, we can sample from the distribution $\mathcal{D}$ and compute $f_\xi(x)$ and $\nabla f_\xi(x)$ at each point x. We assume that the gradient and the expectation are interchangeable, meaning $\mathrm{E}_{\xi\sim\mathcal{D}}\left[\nabla f_\xi(x)\right] = \nabla f(x)$. In the context of cross-device training, f_ξ represents the loss of client ξ on its local data [13].

The formulation (1) is more appropriate than the finite-sum (*cross-silo*) formulation [39]:

$$\min_{x\in\mathbb{R}^d} f(x), \quad \text{where} \quad f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x),$$

as the number of clients n is relatively small, which is more relevant for collaborative training by organizations (e.g., medical [31]).

Communication bottleneck. In federated learning, broadcasting or communicating information between computing nodes, such as the current gradient vector or model state, is necessary. This communication often becomes the main challenge, particularly in the cross-device setting where the nodes are less powerful devices with slow network connections [5, 11, 17]. Two main approaches to reducing communication overhead are compression and local training. Communication compression uses inexact but relevant approximations of the transferred messages at each round. These approximations often rely on (stochastic) compression operators, which can be applied to both the gradient and the model. For a more detailed discussion on compression mechanisms, see [34, 41].

Local training. The second technique for reducing communication overhead is to perform local training. Local SGD steps have been a crucial component of practical federated training algorithms since the inception of the field, demonstrating strong empirical performance by improving communication efficiency [24–26]. However, rigorous theoretical explanations for this phenomenon were lacking until the recent introduction of the ProxSkip method [27]. ScaffNew (ProxSkip specialized to the distributed setting) has been shown to provide accelerated communication complexity in the convex setting. While ScaffNew works for any level of heterogeneity, it does not benefit from the similarity between clients. In addition, methods like ScaffNew, designed to fix the client drift issue [1, 12], require each client to maintain state (control variate), which is incompatible with cross-device FL [33].

Partial participation. In generic (cross-silo) federated learning, periodically, all clients may be active in a single communication round. However, an important property of cross-device learning is the impracticality of accessing all clients simultaneously. Most clients might be available only once during the entire training process. Therefore, it is crucial to design federated learning methods where only a small cohort of devices participates in each round. Modeling the problem according to (1) naturally avoids the possibility of engaging all clients at once. We refer the reader to [13, 14, 33] for more details on partial participation.

Data heterogeneity. Despite recent progress in federated learning, handling the heterogeneity of data across clients remains a significant challenge [11]. Empirical observations show that clients' labels for similar inputs can vary significantly [2, 36]. This variation arises from clients having different preferences. When local steps are used in this context, clients tend to overfit their own data, a phenomenon known as client drift.

An alternative to local gradient steps is to use a local proximal point operator oracle, which involves solving a regularized local optimization problem on the selected client(s). This approach underlies FedProx [19], which relies on a restrictive heterogeneity assumption. The algorithm was analyzed from the perspective of the Stochastic Proximal Point Method (SPPM) in [42]. Independently, the theory of SPPM has been shown to be compatible with the second-order similarity condition (Assumption 2) from an analytical perspective. Based on these connections, various studies have explored SPPM-based federated learning algorithms, and we refer interested readers to [14, 21] for more details.

1.1. Contributions

In this paper, we introduce a novel method called Stochastic Proximal point And Momentum (SPAM). This method combines Momentum Variance Reduction (MVR) on the server side to lever-

Table 1: Comparison of the proposed algorithm with other relevant methods.

Algorithm	**Hessian similarity**	**Partial Participation**	**No Smoothness assumption**	**Cross Device**	**Server update**	**Client oracle**
FedProx [42]	✗	✓	✓	✓	–	PPM
SABER [28]	✓	✗	✓	✗	PAGE	PPM
MIME [13]	✓	✗	✗	✓	MVR	SGD
CE-LSGD [32]	✓	✓	✗	✓	MVR	SARAH
SPAM	✓	✓	✓	✓	MVR	PPM

age its efficiency in stochastic optimization while employing Stochastic Proximal Point Method (SPPM) updates on the clients' side. We analyze four versions of the proposed algorithm:

- SPAM - using exact PPM with constant parameters,
- SPAM - employing exact PPM with varying parameters,
- SPAM-inexact - employing inexact PPM with varying parameters,
- SPAM-PP - using inexact PPM with varying parameters and partial participation.

We then carry out a complete theoretical analysis of the proposed methods, showcasing their advantages compared to relevant competitors and addressing the limitations present in those works. The analysis includes the stationarity guarantees for all variants of SPAM. Specifically, we demonstrate the convergence of the average expected gradient norm to a neighborhood of 0 for all variants.

We also conduct a communication complexity analysis based on our convergence results. Specifically, we match the lower bounds, established in [32], for the number of iterations required to reach precision error ε for SPAM-PP. In addition, following the varying stepsize scheme introduced in the original MVR paper, we design a stepsize schedule that removes the neighborhood from the stationarity bounds. Leveraging this scheme, our algorithm achieves the optimal convergence rate of $O(1/K^{1/3})$, where K denotes the number of iterations.

Our algorithms, in particular, SPAM-PP shine in the cross-device setting when compared to its competitors. First, in contrast to non-SPPM-based algorithms, such as MIME and CE-LSGD, we allow greater *flexibility for the local solvers*. Thus, unlike MIME and CE-LSGD, we do not require neither convexity nor smoothness of the local objectives. In fact, our algorithm is compatible with any local solver, as soon as it satisfies certain conditions outlined in Definition 1. Furthermore, when compared to SABER, our partial participation setting does not require (weak) convexity of the objective. We present a visual comparison of the relevant methods in Table 1.

Another important aspect of our algorithms is that they do not need local states/control variates to be stored on each client, as opposed to many standard federated learning techniques [1, 12, 27]. This is crucial for cross-device learning as each client may participate in training a single time.

Finally, we validate our theoretical findings through meticulously designed experiments. Specifically, we tackle a federated ridge regression problem, where we can control the second-order heterogeneity parameter δ, as well as the computation of the local proximal operator.

2. Notation and assumptions

We use ∇f for the gradient, $\|\cdot\|$ for the Euclidean norm, $\mathrm{E}[\cdot]$ for the expectation. $\mathsf{Unif}(S)$ denotes uniform distribution over the discrete set S. The proximal point operator of a real-valued function $g : \mathbb{R}^d \to \mathbb{R}$ is defined as the solution of the following optimization

$$\operatorname{prox}_g(x) := \arg\min_y \left\{ g(y) + \frac{1}{2}\|x-y\|^2 \right\}. \tag{2}$$

We refer the reader to [4] for the properties of the proximal point operator. There exists a lower bound for function f and it is denote as $f_{\text{inf}} > -\infty$.

We use index i for a non-random client, while ξ is used for a randomly selected client. One of the main assumption of our analysis, is that we have access to stochastic samples $\xi \sim \mathcal{D}$ and in particular we can evaluate the gradient ∇f_ξ at any point $x \in \mathbb{R}^d$.

Assumption 1 (Bounded variance). *We assume there exists $\sigma \geq 0$ such that for any $x \in \mathbb{R}^d$*

$$\mathrm{E}\left[\|\nabla f_\xi(x) - \nabla f(x)\|^2\right] \leq \sigma^2 \tag{3}$$

We say that the function f is L-smooth, if its gradient is Lipschitz continuous $\forall x, y \in \mathbb{R}^d$:

$$\|\nabla f(y) - \nabla f(x)\| \leq L\|x-y\|. \tag{4}$$

In many machine learning scenarios, the non-convex objective functions do not satisfy (4). Moreover, several prior works [7, 43] showed that such smoothness condition does not capture the properties of popular models like LSTM, Recurrent Neural Networks, and Transformers.

Our second assumption is the second order heterogeneity. Further in the analysis, this assumption will take the role of smoothness.

Assumption 2 (Hessian similarity). *Assume there exists $\delta \geq 0$ such that for any ξ and $x, y \in \mathbb{R}^d$*

$$\|\nabla f_\xi(x) - \nabla f(x) - \nabla f_\xi(y) + \nabla f(y)\| \leq \delta\|x-y\|. \tag{5}$$

When all functions f_ξ are twice-differentiable condition (5) can also be formulated as

$$\left\|\nabla^2 f_\xi(x) - \nabla^2 f(x)\right\| \leq \delta, \tag{6}$$

motivating the name *second-order heterogeneity* used interchangeably with *Hessian similarity* [14].

This assumption [22, 35] holds for a large class of machine learning problems where the input data are similar but the labels vary. Typical examples include regression tasks logistic loss functions [40], statistical learning for quadratics [35], generalized linear models [9], and semi-supervised learning [6]. Specifically, the parameter δ remains small, even if different clients have similar input distributions but widely varying outputs for the same input. See more details on the assumption in [14, Section 9]. Furthermore, a similar assumption was used to improve convergence results in centralized [38] and communication-constrained distributed settings [37].

We focus on non-convex optimization problem (1), where it is typically desired to find an ε-approximate stationary point $x \in \mathbb{R}^d$ such that $\mathrm{E}\left[\|\nabla f(x)\|^2\right] \leq \varepsilon$.

3. SPAM

In this section, we describe our main algorithm in its simpler form, that is SPAM with one sampled client and exact proximal point computations. We then provide theoretical convergence guarantees and a complexity analysis of the proposed methods.

The algorithm proceeds as follows. We first choose a stepsize sequence γ_k and a momentum sequence p_k. The server samples a client. Selected client then computes the new gradient estimator g_k and assigns the new iterate as the proximal point operator with a shifted gradient term:

$$x_{k+1} = \mathrm{prox}_{\gamma_k f_{\xi_k}}\left(x_k + \gamma_k(\nabla f_{\xi_k}(x_k) - g_k)\right) = \arg\min_y \phi_k(y),$$

where ϕ_k is defined as

$$\phi_k(y) := f_k(y) + \langle g_k - \nabla f_k(x_k), y - x_k\rangle + \frac{1}{2\gamma_k}\|y - x_k\|^2. \tag{7}$$

The new iterate is then sent to the server and the process repeats itself. For the pseudocode of the algorithm, please refer to Algorithm 1. The following proposition is the cornerstone of our analysis.

Algorithm 1 SPAM, SPAM-inexact

```
Input: Starting point $x_0 = x_{-1} \in \mathbb{R}^d$, initialize $g_0 = g_{-1}$,
choose $\gamma_k > 0$ and $p_k > 0$;
for $k = 0, 1, 2, \ldots$ do
    samples $\xi_k \sim \mathcal{D}$;
    sets $g_k = \nabla f_{\xi_k}(x_k) + (1 - p_k)\left(g_{k-1} - \nabla f_{\xi_k}(x_{k-1})\right)$;
    sets $x_{k+1} \in \begin{cases} \mathrm{prox}_{\gamma_k f_{\xi_k}}\left(x_k + \gamma_k(\nabla f_{\xi_k}(x_k) - g_k)\right); & \triangleright \text{SPAM} \\ \text{a-prox}_\epsilon\left(x_k, g_k, \gamma_k, \xi_k\right); & \triangleright \text{SPAM-inexact} \end{cases}$
    sends $x_{k+1}$ to the server.
end for
```

It provides a recurrent bound for a certain sequence V_k, which serves as a Lyapunov function:

$$V_k = f(x_k) - f_{\inf} + \frac{15\gamma_k}{16(2p_k - p_k^2)}\|g_k - \nabla f(x_k)\|^2. \tag{8}$$

Proposition 1. *Let x_k be the iterates of SPAM for an objective function f, which satisfies Assumptions 1 and 2. If $\gamma_k^2 \le \min\left\{\frac{1}{16\delta^2}, \frac{p_k}{96\delta^2(1-p_k)}\right\}$, then for every $k \ge 1$*

$$\mathrm{E}\left[V_{k+1}\right] \quad \le \quad \mathrm{E}\left[V_k\right] - \frac{\gamma_k}{32}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] + 2\gamma_k p_k \sigma^2,$$

where V_k is defined in (8).

The proof can be found in Appendix G.1. This proposition then leads to a convergence result for SPAM with fixed stepsizes.

Theorem 2 (SPAM with constant parameters)**.** *Suppose Assumptions 1, 2 are satisfied. Then,*

$$\frac{1}{K}\sum_{k=1}^{K}\mathrm{E}\left[\|\nabla f(x_k)\|^2\right] \quad\leq\quad \frac{32(f(x_0)-f_{\inf})}{\gamma K}+\frac{32\|g_0-\nabla f(x_0)\|^2}{(2p-p^2)K}+64p\sigma^2, \tag{9}$$

where $\gamma^2 \leq \min\left\{\frac{1}{16\delta^2}, \frac{p}{96\delta^2(1-p)}\right\}$.

The proof of the theorem can be found in Appendix G.2.

Corollary 3. *The result can also be written as*

$$\mathrm{E}\left[\|\nabla f(\tilde{x}_{K+1})\|^2\right] \quad\leq\quad \frac{32(f(x_0)-f_{\inf})}{\gamma K}+\frac{32\|g_0-\nabla f(x_0)\|^2}{(2p-p^2)K}+64p\sigma^2,$$

where $\tilde{x}_{K+1}$ is taken uniformly randomly from the iterates of the algorithm $\{x_1, x_2, \ldots, x_{K+1}\}$.

Our primary focus is on communication complexity, which is typically the main bottleneck in cross-device federated settings [11]. Below, we present the communication complexity of SPAM with fixed parameters.

Corollary 4. *Choose constant stepsize $\gamma_k = \gamma = \min\left(\frac{1}{\delta}, \left(\frac{F}{2\delta^2\sigma^2 K}\right)^{1/3}\right)$ and momentum parameter $p_k = p = \max(\gamma^2\delta^2, 1/K)$. Then, the communication complexity of SPAM, to obtain ε error is of order $\mathcal{O}\left(\frac{\delta F+\sigma^2}{\varepsilon}+\frac{\delta\sigma F}{\varepsilon^{3/2}}\right)$, where $F := f(x_0) - f_{\inf}$.*

The proof is deferred to Appendix H.1. Our result indicates that higher similarity (smaller δ) leads to fewer communication rounds required to solve the problem. Obtained complexity remarkably improves upon the lower bound $\mathcal{O}\left(\frac{LF+\sigma^2}{\varepsilon}+\frac{L\sigma F}{\varepsilon^{3/2}}\right)$ for $\delta \ll L$ [3].

Suppose now that we can initialize $g_0 = \nabla f(x_0)$. Then, the second term in the convergence upper bound (9) vanishes. Repeating the exact steps as in the proof of Corollary 4, we obtain the convergence rate: $\mathcal{O}\left(\frac{\delta F}{K}+\left(\frac{\delta\sigma F}{K}\right)^{2/3}\right)$, which leads to a communication complexity of $\mathcal{O}\left(\frac{\delta F}{\varepsilon}+\frac{\delta\sigma F}{\varepsilon^{3/2}}\right)$. Thus, our result shows that in the homogeneous case (i.e., $\delta = 0$), communication is not needed at all, as each client can solve the problem locally.

Remark 1. *Lower bounds for two-point first-order oracle federated learning algorithms with local steps were investigated in [32]. However, these bounds are specifically designed for local SGD-type methods, such as MIME. In addition, results by [32] require smoothness. As our methods are agnostic to the choice of local solvers, the applicability of these bounds to our setting remains limited.*

It is important to highlight that the stepsize γ in SPAM differs from the stepsize used in local methods such as MIME and CE-LSGD. In these methods, the stepsize is intended for running the algorithms locally on a selected client. However, SPAM only requires an oracle for proximal points, allowing the oracle to use any optimization method suitable for the problem at hand. Additionally, the stepsize for local SGD-based methods depends on the smoothness parameter, which is not required in our theorem. Thus, our approach allows much more flexibility for choosing local solvers that are adaptive to the curvature of the loss [23, 29]. For a detailed comparison of the methods, see Table 1.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2020.

[2] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[3] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199(1):165–214, 2023.

[4] Amir Beck. *First-order methods in optimization*. SIAM, 2017.

[5] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

[6] El Mahdi Chayti and Sai Praneeth Karimireddy. Optimization with access to auxiliary information. *Transactions on Machine Learning Research*, 2022.

[7] Michael Crawshaw, Mingrui Liu, Francesco Orabona, Wei Zhang, and Zhenxun Zhuang. Robustness to unbounded smoothness of generalized signsgd. *Advances in neural information processing systems*, 35:9955–9968, 2022.

[8] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in Neural Information Processing Systems*, 32, 2019.

[9] Hadrien Hendrikx, Lin Xiao, Sebastien Bubeck, Francis Bach, and Laurent Massoulie. Statistically preconditioned accelerated gradient method for distributed optimization. In *International conference on machine learning*, pages 4203–4227. PMLR, 2020.

[10] Samuel Horváth, Maziar Sanjabi, Lin Xiao, Peter Richtárik, and Michael Rabbat. Fedshuffle: Recipes for better use of local work in federated learning. *Transactions on Machine Learning Research*, 2022.

[11] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021. doi: 10.1561/2200000083. URL https://doi.org/10.1561/2200000083.

[12] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[13] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Breaking the centralized barrier for cross-device federated learning. *Advances in Neural Information Processing Systems*, 34:28663–28676, 2021.

[14] Ahmed Khaled and Chi Jin. Faster federated optimization under second-order similarity. In *The Eleventh International Conference on Learning Representations*, 2022.

[15] 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*, pages 4519–4529. PMLR, 2020.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Jakub Konečný, H. Brendan McMahan, Felix Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: strategies for improving communication efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016.

[18] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *NIPS Private Multi-Party Machine Learning Workshop*, 2016.

[19] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

[20] Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International Conference on Machine Learning*, pages 6286–6295. PMLR, 2021.

[21] Dachao Lin, Yuze Han, Haishan Ye, and Zhihua Zhang. Stochastic distributed optimization under average second-order similarity: Algorithms and analysis. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] Julien Mairal. Optimization with first-order surrogate functions. In *International Conference on Machine Learning*, pages 783–791. PMLR, 2013.

[23] Yura Malitsky and Konstantin Mishchenko. Adaptive gradient descent without descent. In *International Conference on Machine Learning*, pages 6702–6712. PMLR, 2020.

[24] Olvi L Mangasarian and Mikhail V Solodov. Backpropagation convergence via deterministic nonmonotone perturbed minimization. *Advances in Neural Information Processing Systems*, 6, 1993.

[25] Ryan McDonald, Keith Hall, and Gideon Mann. Distributed training strategies for the structured perceptron. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 456–464, 2010.

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

[27] Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, and Peter Richtárik. ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally! In *International Conference on Machine Learning*, pages 15750–15769. PMLR, 2022.

[28] Konstantin Mishchenko, Rui Li, Hongxiang Fan, and Stylianos Venieris. Federated learning under second-order data heterogeneity. *Openreview,* *https://openreview.net/forum?id=jkhVrIllKg*, 2023.

[29] Aaron Mishkin, Ahmed Khaled, Yuanhao Wang, Aaron Defazio, and Robert M Gower. Directional smoothness and gradient methods: Convergence and adaptivity. *arXiv preprint arXiv:2403.04081*, 2024.

[30] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pages 2613–2621. PMLR, 2017.

[31] Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, et al. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *Advances in Neural Information Processing Systems*, 35:5315–5334, 2022.

[32] Kumar Kshitij Patel, Lingxiao Wang, Blake E Woodworth, Brian Bullins, and Nati Srebro. Towards optimal communication complexity in distributed non-convex optimization. *Advances in Neural Information Processing Systems*, 35:13316–13328, 2022.

[33] Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2020.

[34] Mher Safaryan, Egor Shulgin, and Peter Richtárik. Uncertainty principle for communication compression in distributed and federated learning and the search for an optimal compressor. *Information and Inference: A Journal of the IMA*, 11(2):557–580, 2022.

[35] Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pages 1000–1008. PMLR, 2014.

[36] Andrew Silva, Katherine Metcalf, Nicholas Apostoloff, and Barry-John Theobald. Fedembed: Personalized private federated learning. *arXiv preprint arXiv:2202.09472*, 2022.

[37] 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.

[38] Alexander Tyurin, Lukang Sun, Konstantin Burlachenko, and Peter Richtárik. Sharper rates and flexible framework for nonconvex SGD with client and data sampling. *Transactions on Machine Learning Research*, 2023.

[39] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

[40] Blake Woodworth, Konstantin Mishchenko, and Francis Bach. Two losses are better than one: Faster optimization using a cheaper proxy. In *International Conference on Machine Learning*, pages 37273–37292. PMLR, 2023.

[41] Hang Xu, Chen-Yu Ho, Ahmed M. Abdelmoniem, Aritra Dutta, El Houcine Bergou, Konstantinos Karatsenidis, Marco Canini, and Panos Kalnis. Compressed communication for distributed deep learning: Survey and quantitative evaluation. *Technical report, http://hdl.handle.net/10754/662495*, 2020. URL http://hdl.handle.net/10754/662495.

[42] Xiaotong Yuan and Ping Li. On convergence of FedProx: Local dissimilarity invariant bounds, non-smoothness and beyond. *Advances in Neural Information Processing Systems*, 35:10752–10765, 2022.

[43] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations (ICLR)*, 2019.

Appendix

Contents

Appendix A. Prior work

Momentum. Momentum Variance Reduction (MVR) was introduced in the context of server-only stochastic non-convex optimization [8]. The primary motivation behind this method, also known as STORM, was to avoid computing full gradients (which is impractical in the stochastic setting) or requiring "giant batch sizes" of order $\mathcal{O}(1/\varepsilon^2)$. Such large batch sizes are necessary for other methods like PAGE [20] to find an ε-stationary point.

The authors assume bounded variance for stochastic gradients ∇f_ξ and analyze the method under additional restrictive conditions. However, these conditions can be replaced with the second-order heterogeneity (Assumption 2). The convergence result of MVR for non-convex objectives includes the stochastic gradient noise term σ^2 in the upper bound. To eliminate the dependence on this parameter, they propose an adaptive stepsize schedule, under the additional assumption that f_ξ is Lipschitz continuous.

MIME. MIME is a flexible framework that makes existing optimization algorithms applicable in the distributed setting [13]. They describe a general scheme that combines local SGD updates with a generic optimization algorithm. The authors then study particular instances of framework, such as MIME + ADAM [16] and MIME + MVR [8].

However, their analysis with local steps is limited from the non-convex cross-device learning perspective. First, they assume smoothness also in the case of one sampled client. Moreover, MIME suffers from a common issue of local methods. In Theorem 4 of [13], the stepsize is taken to be of order $O(1/Lm)$, where L is the smoothness parameter of the client loss and m is the number of local steps. This means that the stepsize is so small that multiple steps become equivalent to a single, smoother stochastic gradient descent step, negating the potential benefits of local SGD. Finally, their analysis requires an additional weak convexity assumption for the objective in the partial participation setting.

CE-LSGD. The Communication Efficient Local Stochastic Gradient Descent (CE-LSGD) was introduced by [32]. They propose and analyze two algorithms, with the second one tailored for the cross-device setting (1). This algorithm comprises two components: the MVR update on the server and SARAH local steps on the selected client. The latter, known as the Stochastic Recursive Gradient Algorithm, is a variance-reduced version of SGD that periodically requires the gradient of the objective function [30].

The analysis of [32] explicitly describes how to choose the number of local updates and the local stepsize. They also provide lower bounds for two-point first-order oracle-based federated learning algorithms. The drawback of their setting is that in order to have meaningful local updates, they need smoothness of each client function f_ξ. In addition, similar to MIME, it has a dependence of the stepsize on the number of local steps, which undermines the benefits of doing many steps.

SABER. The SABER algorithm combines SPPM updates on the clients with PAGE updates on the server [28]. Their paper utilizes Hessian similarity (Assumption 2) and leverages it for the finite-sum optimization objective. However, their analysis for the partial participation setting relies on an assumption that is difficult to verify in the general non-convex regime. In fact, if the function is not weakly convex, as in the case of MIME, this assumption may not hold true. Specifically, it requires that $f\left(\frac{1}{b}\sum_{i=1}^b w_i\right) \leq \frac{1}{b}\sum_{i=1}^b f(w_i)$, where w_i are arbitrary vectors in $\mathbb{R}^d$ obtained using proximal point operators.

Appendix B. SPAM with time varying parameters

In (9) we notice that the last term, which is due to the stochastic nature of our problem, does not vanish when K is large. In order to remove the stationarity neighborhood, let us now consider varying stepsizes for SPAM, with decaying momentum parameters p_k.

Theorem 5 (SPAM)**.** *Consider SPAM for an objective function f that satisfies Assumptions 1 and 2. Let γ_k be a sequence of varying stepsizes satisfying $\gamma_k^2 \leq \frac{1}{16\delta^2}$ and choose $p_k = \frac{96\delta^2\gamma_k^2}{96\delta^2\gamma_k^2+1}$. Then,*

$$\frac{1}{\Gamma_K}\sum_{k=1}^{K}\gamma_k \mathrm{E}\left[\|\nabla f(x_k)\|^2\right] \quad \leq \quad \frac{32V_0}{\Gamma_K} + \frac{2}{\Gamma_K}\sum_{k=1}^{K}\frac{96\delta^2\gamma_k^3}{96\delta^2\gamma_k^2+1}\sigma^2, \tag{10}$$

where $\Gamma_K = \sum_{k=1}^{K}\gamma_k$.

The proof of Theorem 5 can be found in Appendix G.3.

Remark 2. *Similar to Theorem 2, we can represent the left-hand side of* (10) *with a single expectation:* $\mathrm{E}\left[\|\nabla f(\tilde{x}_K)\|^2\right]$*, where $\tilde{x}_K = x_i$, for $i = 1, \dots, K$ with probability γ_i/Γ_K.*

To ensure that the right-hand side converges to zero as $K \to \infty$, we need $\gamma_K \to 0$ and $\Gamma_K \to +\infty$. This suggests using a stepsize schedule of order $\gamma_k = O(k^{\beta-1})$, implying $\Gamma_K = O(K^{\beta})$ for some $\beta \in (0,1)$. Consequently, the right-hand side of (10) is of order $O(K^{-\beta} + K^{2\beta-2})$. By optimizing over β, we deduce that $\gamma_k = O(k^{-1/3})$ results in a stationarity bound of order $O(K^{-2/3})$.

Corollary 6 (Optimal stepsize schedule)**.** *If $\gamma_k = \frac{1}{4\delta k^{1/3}}$ and $p_k = \frac{96\delta^2\gamma_k^2}{96\delta^2\gamma_k^2+1}$, then to obtain ε-stationarity for SPAM we need $K = O(\varepsilon^{3/2})$ iterations under assumptions 1 and 2.*

This coincides with the existing lower bounds by [3], meaning that our result is tight up to constants.

Appendix C. Inexact proximal operator

In the previous theorems, we assume that each sampled client ξ_k can do an exact computation of the proximal operator to obtain the new iterate x_{k+1}. The latter means, that this client can exactly solve a (potentially) non-convex minimization problem, which might be hard in practice. However, in the proofs of these theorems, we do not use that the new iterate x_{k+1} is the exact solution of the proximal operator (see Appendix J.1). Instead, we use two properties of the proximal point operator:

- decrease in function value: $\phi_k(x_{k+1}) \leq \phi_k(x_k)$;
- stationarity: $\nabla\phi_k(x_{k+1}) = 0$.

Thus, we can replace the step of finding an exact proximal point in Algorithm 1 with finding a point that satisfies the above two conditions. Furthermore, we will relax the latter condition by taking an approximate stationary point. These arguments are summarized in the below assumption.

Definition 1 (a-prox)**.** *For a given client k, a gradient estimator g_k, a current state x_k, a stepsize γ_k and a precision level ϵ, the approximate proximal point* $\text{a-prox}_\epsilon(x_k, g_k, \gamma_k, k)$ *is the set of vectors y_{ap}, which satisfy*

- *decrease in function value:* $\mathrm{E}[\phi_k(y_{\mathsf{ap}})] \leq \phi_i(x^k)$*;*
- *approximate stationarity:* $\mathrm{E}\left[\|\nabla \phi_k(y_{\mathsf{ap}})\|^2\right] \leq \epsilon^2$*,*

where ϕ_k is defined in (7). We then replace the exact proximal step in line 6 of Algorithm 1 with the approximate operator.

Theorem 7 (SPAM-inexact)**.** *Consider* SPAM-inexact *for an objective function f that satisfies Assumptions 1 and 2. Let γ_k be a sequence of varying stepsizes satisfying $\gamma^2 \leq \frac{1}{16\delta^2}$ and choose $p_k = \frac{96\delta^2\gamma_k^2}{96\delta^2\gamma_k^2+1}$. Then,*

$$\frac{1}{\Gamma_K}\sum_{k=1}^{K}\gamma_k \mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] \quad \leq \quad \frac{40 V_0}{\Gamma_K} + \frac{2}{\Gamma_K}\sum_{k=1}^{K} p_k\gamma_k^2\sigma^2 + \frac{\epsilon^2}{8}. \tag{11}$$

where $\Gamma_K = \sum_{k=1}^{K}\gamma_k$.

The proof is postponed to Appendix G.4. We observe that the level of inexactness ϵ^2 appears explicitly in the theorem. In case, when $\epsilon = 0$, we recover the result in Theorem 5 up to a constants. SPAM-inexact allows to avoid solving the local minimization problem required for finding the inexact proximal point operator. This is a significant improvement over SPAM, as the latter requires minimizing (potentially) non-convex objectives at each iteration.

Appendix D. Partial participation

In this section, we present the most general form of our algorithm, which works with the approximate proximal operator and samples multiple clients (cohort) at each round. Specifically, it uses the random cohort S_k to construct a better gradient estimator g_k. This gradient estimator is then broadcasted to a single random client $\xi_k \sim \mathcal{D}$, who computes the approximate proximal point locally. The pseudocode can be found in Algorithm 2.

Theorem 8 (SPAM-PP)**.** *Suppose Assumptions 1 and 2 are satisfied. If $\xi_k \sim \mathsf{Unif}(S_k)$ at every iteration, then the iterates of* SPAM-PP *with $\gamma_k \leq \frac{1}{4\delta}$ and $p_k = \frac{96\delta^2\gamma_k^2}{96\delta^2\gamma_k^2+B^2}$ satisfy*

$$\frac{1}{\Gamma_K}\sum_{k=0}^{K-1}\gamma_k \mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] \quad \leq \quad \frac{40}{\Gamma_K}\left(V_0 - \mathrm{E}[V_K]\right) + \frac{240}{\Gamma_K}\sum_{k=0}^{K-1} p_k\gamma_k\frac{\sigma^2}{B} + 7.5\epsilon^2.$$

The proof of the theorem is postponed to Appendix G.5. When the client cohort size B increases, the neighborhood shrinks. This is intuitive as when $B \to \infty$, we can have access to the exact objective f, and the neighborhood will vanish.

Corollary 9. *For properly chosen constant $\gamma_k = \gamma$ and momentum parameter $p_k = p$ communication complexity of* SPAM-PP*, to obtain ε error is of order*

$$\mathcal{O}\left(\frac{\delta F}{\varepsilon} + \frac{\sigma^2}{B\varepsilon} + \frac{\delta\sigma F}{\sqrt{B}\varepsilon^{3/2}}\right).$$

Algorithm 2 SPAM-PP

```
Input: learning rate $\gamma > 0$, cohort size $B$, starting point $x_0 \in \mathbb{R}^d$;
proximal precision level $\epsilon$; initialize $g_0 = g_{-1}$;
for $k = 0, 1, 2, \dots$ do
    samples a subset of clients $S_k$, with size $|S_k| = B$;
    broadcasts $x_k$ to the clients from $S_k$.
    for $i \in S_k$ in parallel do
        Set $g_k^i = \nabla f_i(x_k) + (1 - p_k)\left(g_{k-1} - \nabla f_i(x_{k-1})\right)$;
        Send $g_k^i$ to the server;
    end for
    $g_k = \frac{1}{B} \sum_{i \in S_k} g_k^i$ ;
    $\xi^{k+1} \sim \mathcal{D}$;
    $x_{k+1} \in \text{a-prox}_\epsilon\left(x_k, g_k, \gamma_k, \xi^{k+1}\right)$;
end for
```

This result significantly improves upon the lower bound for L-smooth case $\mathcal{O}\left(\frac{LF}{\varepsilon} + \frac{\sigma^2}{B\varepsilon} + \frac{L\sigma F}{\sqrt{B}\varepsilon^{3/2}}\right)$ when $\delta \ll L$ [3]. We also observe that the complexity is an increasing function of δ. Thus, our bound improves when data on different clients is similar. Moreover, increasing cohort size B brings acceleration but to a certain level. The proof of the corollary can be found in Appendix H.2.

In Appendix I, we present another version of SPAM-PP, called SPAM-PPA, which uses the sampled cohort of clients to compute local proximal points. These points are then communicated to the server, and the new iterate is their average. Hence, the name SPAM-PP with Averaging.

Appendix E. Experiments

To empirically validate our theoretical framework and its implications, we focus on a carefully controlled experimental setting similar to [14, 21]. Specifically, we consider a distributed ridge regression problem formulated in (25), which allows us to calculate and control the Hessian similarity δ. An essential advantage of this optimization problem is that the proximal operator has an explicit (closed-form) representation and can be computed precisely (up to machine accuracy). This allows us to isolate the effect of varying parameters on the method's performance. Appendix K provides more details on the setup.

In Figure 1, we display convergence of Algorithm 1 with constant parameters p and γ. The legend is shared, and labels refer to proximal operator computations: "exact" means using closed-form solution, "1" and "10" correspond to the number of local gradient descent steps. We evaluate the logarithm of a relative gradient norm $\log(\|\nabla f(x_k)\|/\|\nabla f(x_0)\|)$ in the vertical axis. At every iteration, one client is sampled uniformly at random.

Observations. All the plots indicate convergence of the method to the neighborhood of the stationary point, followed by subsequent oscillations around the error floor. The first (left) plot shows that for small momentum $p = 0.1$ and γ exceeding the theoretical bound $1/\delta$, the algorithm can be very unstable with exact proximal point computations. Interestingly, approximate computation (1 or 10 local steps) results in more robust convergence. The second (middle) plot demonstrates that a greater $p = 0.9$ results in steady convergence even for misspecified (too large) γ. In addition, one can observe that in this case, more accurate proximal point evaluation results in significantly faster convergence but to a larger neighborhood than for one local step. This agrees well with observations for local gradient descent methods [15]. The last (right) figure shows that a properly chosen, smaller $\gamma = 0.5/\delta$ slows down convergence (twice as many communication rounds are shown). However, the method reaches a significantly lower error floor (as the vertical axis is shared across plots), which does not depend much on the accuracy of proximal point operator calculation. Moreover, 10 local steps are enough for basically the same fast convergence as with exact proximal point computation.

We would like to specifically note that momentum-based variance reduction has already shown empirical success [10, 13] in practical federated learning scenarios. That is why our experiments focus on simpler but insightful setting to carefully study the properties of the proposed algorithm.

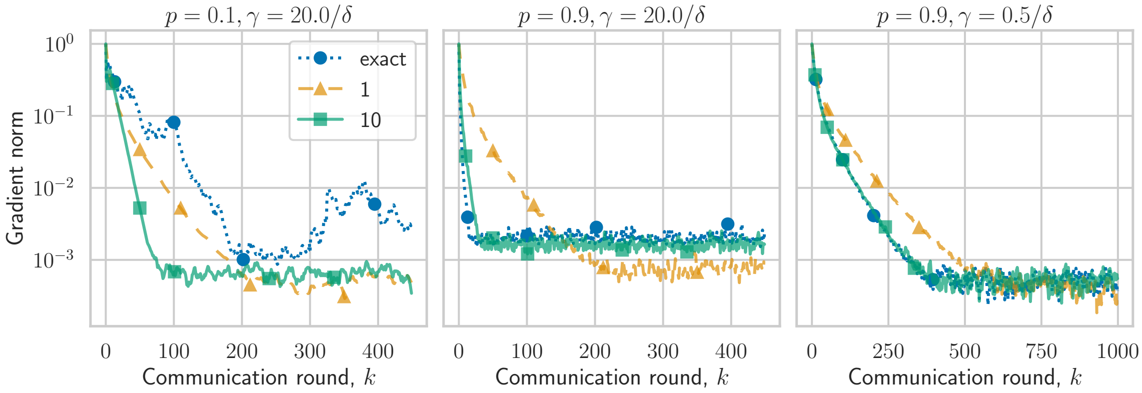

Figure 1: Convergence of SPAM-inexact on a ridge regression problem with different p and γ.

Appendix F. Conclusion

We introduced SPAM, an algorithm tailored for cross-device federated learning, which combines momentum variance reduction with the stochastic proximal point method. Operating under conditions of second-order heterogeneity and bounded variance, SPAM does not necessitate smoothness of the objective function. In its most general form, SPAM achieves optimal communication complexity. Furthermore, it does not prescribe a specific local method for analysis, providing practitioners with flexibility and responsibility in selecting suitable local solver.

Limitations and future work. The paper is of a theoretical nature and focuses on improving the understanding of stochastic non-convex optimization under hessian similarity in the context of cross-device federated learning. We believe that separate experiments should be conducted to evaluate the experimental performance in a setting close to real life.

In standard optimization, the stepsize usually depends on the smoothness parameter. Adaptive methods allow to iteratively adjust the stepsize without additional information. In our case, the smoothness parameter is replaced by the second-order heterogeneity parameter δ, on which the stepsize and momentum sequences of SPAM depend. Removing this dependence using adaptive techniques under general assumptions remains an open problem even for the server-only MVR, which serves as the basis for our algorithm.

Finally, federated learning comprises other aspects that we have not discussed above. These include privacy, security, personalization, etc., while our focus is on optimization and communication complexity. We leave the study of their interplay as future work.

Appendix G. Proofs

G.1. Proof of Proposition 1

Recall that

$$V_k = f(x_k) - f_{\text{inf}} + \frac{3\gamma_k}{2p_k - p_k^2}\|g_k - \nabla f(x_k)\|^2.$$

We bound each term separately. We formulate three technical lemmas, which are proved in Appendix J. We start with bounding the first term, that is the function values.

Lemma 10. *Under the conditions of Proposition 1, the following recurrent inequality takes place*

$$f(x_{k+1}) - f_{\text{inf}} \le f(x_k) - f_{\text{inf}} - \frac{1}{4\gamma_k}\|x_{k+1} - x_k\|^2 + 2\gamma_k\|\nabla f(x_k) - g_k\|^2 \tag{12}$$

Then, we bound the second term of V_k.

Lemma 11. *Under the conditions of Proposition 1, the following recurrent inequality takes place*

$$\mathrm{E}\left[\|g_{k+1} - \nabla f(x_{k+1})\|^2 \middle| \mathcal{F}_k\right] \le (1-p_k)^2\|g_k - \nabla f(x_k)\|^2 + 2(1-p_k)^2\delta^2\|x_{k+1} - x_k\|^2 + 2p_k^2\sigma^2. \tag{13}$$

We observe that in both upper bounds, there is the term $\|x_{k+1} - x_k\|^2$. The following lemma, provides a lower bound for this expression.

Lemma 12. *Under the conditions of Proposition 1, the following recurrent inequality is true*

$$\mathrm{E}\left[\|x_{k+1} - x_k\|^2\right] \ge \frac{\gamma_k^2}{4}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] - \gamma_k^2\mathrm{E}\left[\|g_k - \nabla f(x_k)\|^2\right]. \tag{14}$$

We now combine the results of the lemmas to bound V_{K+1}:

$$\begin{aligned}
\mathrm{E}\left[V_{k+1}\right] \overset{(12)+(13)}{\le}\; & \alpha(1-p_k)^2\|g_k - \nabla f(x_k)\|^2 + 2\alpha\delta^2(1-p_k)^2\|x_{k+1} - x_k\|^2 + 2\alpha p_k^2\sigma^2 \\
& + \mathrm{E}\left[f(x_k) - f_{\text{inf}}\right] - \frac{1}{4\gamma_k}\mathrm{E}\left[\|x_{k+1} - x_k\|^2\right] + 2\gamma_k\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] \\
=\; & \mathrm{E}\left[V_k\right] + \left(2\alpha\delta^2(1-p_k)^2 - \frac{1}{4\gamma_k}\right)\mathrm{E}\left[\|x_{k+1} - x_k\|^2\right] + 2\alpha p_k^2\sigma^2 \\
& + (2\gamma_k - \alpha(2p_k - p_k^2))\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right].
\end{aligned}$$

The last inequality is true for every positive value of α. Let us now choose $\alpha = \frac{3\gamma_k}{2p_k - p_k^2}$. Then,

$$2\alpha\delta^2(1-p_k)^2 - \frac{1}{4\gamma_k} = \frac{6\gamma_k\delta^2(1-p_k)^2}{2p_k - p_k^2} - \frac{1}{4\gamma_k} \le -\frac{1}{8\gamma_k},$$

where the latter is due to $4\delta\gamma_k \leq \sqrt{p_k/6(1-p_k)}$. Therefore, we deduce

$$\begin{aligned}
\mathrm{E}\left[V_{k+1}\right] &\leq \mathrm{E}\left[V_k\right] - \frac{1}{8\gamma_k}\mathrm{E}\left[\|x_{k+1}-x_k\|^2\right] - \gamma_k\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] + 2\alpha p_k^2\sigma^2 \\
&\overset{(14)}{\leq} \mathrm{E}\left[V_k\right] - \frac{\gamma_k}{32}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] + \frac{\gamma_k}{8}\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] \\
&\quad -\gamma_k\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] + \frac{6\gamma_k p_k}{2-p_k}\sigma^2 \\
&\leq \mathrm{E}\left[V_k\right] - \frac{\gamma_k}{32}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] + 6\gamma_k p_k\sigma^2.
\end{aligned}$$

This concludes the proof of the proposition.

G.2. Proof of Theorem 2

Let us apply Proposition 1 for the fixed stepsize $\gamma_k = \gamma$ and a fixed momentum coefficient $p_k = p$.

$$\mathrm{E}\left[V_{k+1}\right] \leq \mathrm{E}\left[V_k\right] - \frac{\gamma}{32}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] + 6\gamma p\sigma^2.$$

Summing up these inequalities for $k = 0, \ldots, K-1$ leads to

$$\begin{aligned}
\frac{1}{K}\sum_{k=1}^{K}\mathrm{E}\left[\|\nabla f(x_k)\|^2\right] &\leq \frac{32}{\gamma K}\left(V_0 - \mathrm{E}\left[V_K\right]\right) + 192p\sigma^2 \\
&\leq \frac{32(f(x_0) - f_{\inf})}{\gamma K} + \frac{30\|g_0 - \nabla f(x_0)\|^2}{(2p-p^2)K} + 192p\sigma^2.
\end{aligned}$$

where $\gamma^2 \leq \min\left\{\frac{1}{16\delta^2}, \frac{4p}{3\delta^2(1-p)}\right\}$. This concludes the proof of the theorem.

G.3. Proof of Theorem 5

From Proposition 1 we have

$$-\frac{\gamma_k}{32}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] \leq \mathrm{E}\left[V_k\right] - \mathrm{E}\left[V_{k+1}\right] + 6\gamma_k p_k\sigma^2.$$

Let us now sum up these inequalities for $k = 0, 1, \ldots, K-1$. We have telescoping sum on the right-hand side. Then, dividing both sides on $\Gamma_K = \sum_{i=1}^{K}\gamma_i$, we deduce the following bound:

$$\frac{1}{\Gamma_K}\sum_{k=1}^{K}\gamma_k\mathrm{E}\left[\|\nabla f(x_k)\|^2\right] \leq \frac{32V_0}{\Gamma_K} + \frac{2}{\Gamma_K}\sum_{k=1}^{K}\frac{15\delta^2\gamma_k^3}{15\delta^2\gamma_k^2 + 4}\sigma^2.$$

This concludes the proof.

G.4. Proof of Theorem 7

We start by repeating the steps of the proof for Proposition 1. Notice that, in the statement of the proposition, we assume that the iterate is exactly equal to the proximal point operator. However, as stated in Appendix C, in the proofs of lemmas 10 and 11 we only use the property that $\phi_k(x_{k+1}) \leq \phi_k(x_k)$ (see (21)). Thus, both (12) and (13) are true for SPAM-inexact. Therefore,

$$\begin{aligned}\mathrm{E}\left[V_{k+1}\right] \quad \leq \quad \mathrm{E}\left[V_k\right] \quad & -\frac{1}{8\gamma_k}\mathrm{E}\left[\|x_{k+1}-x_k\|^2\right] - (2\gamma_k - \alpha(2p_k - p_k^2))\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] \\ & +2\alpha p_k^2\sigma^2.\end{aligned}$$

Below, reformulate the adaptation of Lemma 12 for the inexact case to lower bound the second term on the right-hand side.

Lemma 13. *Under the conditions of Proposition 1, we have the following bound*

$$\mathrm{E}\left[\|x_{k+1}-x_k\|^2\right] \geq \frac{\gamma_k^2}{5}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] - \gamma_k^2\mathrm{E}\left[\|g_k - \nabla f(x_k)\|^2\right] - \gamma_k^2\epsilon^2. \tag{15}$$

The proof can be found in Appendix J.4. Thus,

$$\begin{aligned}\mathrm{E}\left[V_{k+1}\right] \quad & \overset{(15)}{\leq} \quad \mathrm{E}\left[V_k\right] - \frac{\gamma_k}{40}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] + \frac{15\gamma_k p_k}{8(2-p_k)}\sigma^2 + \frac{\gamma_k\epsilon^2}{8} \\ & \qquad +\frac{\gamma_k}{8}\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] - (2\gamma_k - \alpha(2p_k - p_k^2))\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] \\ & \leq \quad \mathrm{E}\left[V_k\right] - \frac{\gamma_k}{40}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] + 2\gamma_k p_k\sigma^2 + \frac{\gamma_k\epsilon^2}{8}.\end{aligned}$$

Repeating this step for $k = 0, \ldots, K-1$, we deduce

$$\frac{1}{\Gamma_K}\sum_{k=0}^{K-1}\gamma_k\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] \quad \leq \quad \frac{40V_0}{\Gamma_K} + \frac{2}{\Gamma_K}\sum_{k=0}^{K-1}\frac{15\sigma^2\gamma_k^3}{15\sigma^2\gamma_k^2+4}\sigma^2 + \frac{\epsilon^2}{8}.$$

G.5. Proof of Theorem 8

The proof follows the logic of Proposition 1. Recall that

$$V_k = f(x_k) - f_{\inf} + \frac{3\gamma_k}{2p_k - p_k^2}\|g_k - \nabla f(x_k)\|^2.$$

Recall that Lemma 10 is true for any gradient estimator g_k. Thus, (12) is valid for SPAM-PP as well. Next, we estimate the second term of the Lyapunov function. Recall that

$$\begin{aligned}g_{k+1} &= \frac{1}{S_k}\sum_{i\in S_k}\left\{\nabla f_i(x_{k+1}) + (1-p_k)\left(g_k - \nabla f_i(x_k)\right)\right\} \\ &= \nabla\tilde{f}_k(x_{k+1}) + (1-p_k)\left(g_k - \nabla\tilde{f}_k(x_k)\right),\end{aligned}$$

where $\tilde{f}_k(x) := \frac{1}{S_k}\sum_{i\in S_k}\nabla f_i(x)$. Notice also that $\mathrm{E}\left[\tilde{f}_k(x)\right] = f(x)$, for every fixed $x \in \mathbb{R}^d$. Furthermore, combining the convexity of the Euclidean norm and Hessian similarity (5) we deduce that the estimator $\tilde{f}_k$ satisfies the Hessian similarity condition

$$\begin{aligned}\left\|\nabla\tilde{f}_k(x) - \nabla f(x) - \nabla\tilde{f}_k(y) + \nabla f(y)\right\| &\leq \frac{1}{B}\sum_{i\in S_k}\|\nabla f_i(x) - \nabla f(x) - \nabla f_i(y) + \nabla f(y)\| \\ &\leq \frac{\delta}{B}\|x-y\|.\end{aligned}$$

Finally, Jensen's inequality implies that $\tilde{f}_k$ satisfies the bounded variance condition as well:

$$\mathrm{E}\left[\left\|\nabla \tilde{f}_k(x) - \nabla f(x)\right\|\right] \le \sigma^2/B.$$

Repeating the analysis exactly as in the proof of Lemma 11, we obtain

$$\begin{aligned}\mathrm{E}\left[\|g_{k+1} - \nabla f(x_{k+1})\|^2\right] \quad &\le \quad (1-p_k)^2\mathrm{E}\left[\|g_k - \nabla f(x_k)\|^2\right] \\ &\quad +2(1-p_k)^2\frac{\delta^2}{B^2}\mathrm{E}\left[\|x_{k+1}-x_k\|^2\right] + \frac{2p_k^2\sigma^2}{B}. \end{aligned} \tag{16}$$

Let us now bound the Lyapunov function using (12) and (16):

$$\begin{aligned}\mathrm{E}\left[V_{k+1}\right] \quad &\le \quad \mathrm{E}\left[f(x_k) - f_{\inf}\right] + 2\gamma_k\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] - \frac{1}{4\gamma_k}\mathrm{E}\left[\|x_{k+1}-x_k\|^2\right] \\ &\quad +\alpha(1-p_k)^2\mathrm{E}\left[\|g_k - \nabla f(x_k)\|^2\right] + 2\alpha(1-p_k)^2\frac{\delta^2}{B^2}\mathrm{E}\left[\|x_{k+1}-x_k\|^2\right] + \frac{2\alpha p_k^2\sigma^2}{B} \\ &= \quad \mathrm{E}\left[V_k\right] + \left(2\alpha\frac{\delta^2}{B^2}(1-p_k)^2 - \frac{1}{4\gamma_k}\right)\mathrm{E}\left[\|x_{k+1}-x_k\|^2\right] + \frac{2\alpha p_k^2\sigma^2}{B} \\ &\quad +(2\gamma_k - \alpha(2p_k - p_k^2))\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right].\end{aligned}$$

The latter is true for every positive α. Let us now plug in the value of $\alpha = \frac{3\gamma_k}{2p_k - p_k^2}$. Then, using $\gamma \le \sqrt{\frac{B^2 p_k}{96\delta^2(1-p_k)}}$, we obtain

$$2\alpha\frac{\delta^2}{B^2}(1-p_k)^2 - \frac{1}{4\gamma_k} \le \frac{6\gamma_k\delta^2}{B^2(2p_k - p_k^2)}(1-p_k)^2 - \frac{1}{4\gamma_k} \le -\frac{1}{8\gamma_k}. \tag{17}$$

Hence, we have the following bound

$$\begin{aligned}\mathrm{E}\left[V_{k+1}\right] \quad &\le \quad \mathrm{E}\left[V_k\right] - \frac{1}{8\gamma_k}\mathrm{E}\left[\|x_{k+1}-x_k\|^2\right] - \gamma_k\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] + \frac{6p_k\gamma_k\sigma^2}{B(2-p_k)} \\ &\overset{(15)}{\le} \quad \mathrm{E}\left[V_k\right] - \frac{1}{8\gamma_k}\left(\frac{\gamma_k^2}{5}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] - \gamma_k^2\mathrm{E}\left[\|g_k - \nabla f(x_k)\|^2\right] - \gamma_k^2\epsilon^2\right) \\ &\quad -\gamma_k\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] + \frac{6p_k\gamma_k\sigma^2}{B} \\ &\le \quad \mathrm{E}\left[V_k\right] - \frac{\gamma_k}{40}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] - \frac{7\gamma_k}{8}\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] + \frac{6p_k\gamma_k\sigma^2}{B} + \frac{\gamma_k\epsilon^2}{8} \\ &\le \quad \mathrm{E}\left[V_k\right] - \frac{\gamma_k}{40}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] + \frac{6p_k\gamma_k\sigma^2}{B} + \frac{\gamma_k\epsilon^2}{8}.\end{aligned}$$

Thus, we have

$$\frac{1}{\Gamma_K}\sum_{k=0}^{K-1}\gamma_k\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] \quad \le \quad \frac{40}{\Gamma_K}\left(V_0 - \mathrm{E}\left[V_K\right]\right) + \frac{240}{\Gamma_K}\sum_{k=0}^{K-1}p_k\gamma_k\frac{\sigma^2}{B} + 7.5\epsilon^2.$$

This concludes the proof of the theorem.

Appendix H. Complexity analysis of the methods

We use $\lesssim$ to ignore numerical constants in the subsequent analysis.

H.1. Proof of Corollary 4

We have stepsize condition $\gamma \lesssim \min\{1/\delta, \sqrt{p/(\delta^2(1-p))}\}$, which implies that $\gamma \lesssim \sqrt{p}/\delta$ or $p \gtrsim (\gamma\delta)^2$. Denote $F := f(x_0) - f_{\inf}$, then convergence rate of SPAM can be expressed as

$$R_K := \frac{1}{K}\sum_{k=1}^{K} \mathrm{E}\left[\|\nabla f(x_k)\|^2\right] \lesssim \frac{f(x_0)-f_{\inf}}{\gamma K} + \frac{\|g_0 - \nabla f(x_0)\|^2}{(2p-p^2)K} + p\sigma^2$$
$$\lesssim \frac{F}{\gamma K} + \frac{\|g_0 - \nabla f(x_0)\|^2}{pK} + p\sigma^2,$$

where in the last inequality we used condition for the stepsize and the fact that $p(2-p) \geq p$. Next, by using an argument similar to that in [13], we suppose (without loss of generality) that the method is run for K iterations. For the first $K/2$ iterations, we simply sample ∇f_ξ at x_0 to set $g_0 = \frac{1}{K/2}\sum_{i=1}^{K/2} \nabla f_{\xi_i}(x_0)$. Then, according to (3), we have $\mathrm{E}\left[\|g_0 - \nabla f(x_0)\|^2\right] \leq \frac{\sigma^2}{K/2}$. Now, choose $p = \max(\gamma^2\delta^2, 1/K)$

$$R_K \lesssim \frac{F}{\gamma K} + \frac{\sigma^2}{pK^2} + p\sigma^2 \lesssim \frac{F}{\gamma K} + \frac{\sigma^2}{K} + \gamma^2\delta^2\sigma^2 + \frac{\sigma^2}{K}.$$

Next set $\gamma = \min\left(\frac{1}{\delta}, \left(\frac{F}{2\delta^2\sigma^2 K}\right)^{1/3}\right)$ and the rate results in

$$R_K \lesssim \frac{\delta F}{K} + \frac{F}{K}\left(\frac{2\delta^2\sigma^2 K}{F}\right)^{1/3} + \left(\frac{F}{2\delta^2\sigma^2 K}\right)^{2/3}\delta^2\sigma^2 + \frac{\sigma^2}{K}$$
$$\lesssim \frac{\delta F + \sigma^2}{K} + \left(\frac{\delta\sigma F}{K}\right)^{2/3},$$

which leads to communication complexity of $\mathcal{O}\left(\frac{\delta F+\sigma^2}{\varepsilon} + \frac{\delta\sigma F}{\varepsilon^{3/2}}\right)$. This concludes the proof.

H.2. Proof of Corollary 9

In this part we perform analyze communication complexity similarly to Section H.2. The focus is on constant stepsize case $\gamma_k \equiv \gamma \lesssim \min\{1/\delta, \sqrt{p}B/\delta\}$ and exact proximal computation $\epsilon = 0$. Denote $F := f(x_0) - f_{\inf}$, then convergence rate of SPAM-PP can be expressed as

$$R_K := \frac{1}{K}\sum_{k=1}^{K} \mathrm{E}\left[\|\nabla f(x_k)\|^2\right] \lesssim \frac{F}{\gamma K} + \frac{\|g_0 - \nabla f(x_0)\|^2}{pK} + \frac{p\sigma^2}{B}.$$

By using the same reasoning as in H.1 set

$$g_0 = \frac{1}{BK/2}\sum_{j=1}^{K/2}\sum_{i=1}^{S_j} \nabla f_{\xi_i}(x_0)$$

to make sure $\mathrm{E}\left[\|g_0 - \nabla f(x_0)\|^2\right] \leq \sigma^2/(BK/2)$. Then

$$R_K \lesssim \frac{F}{\gamma K} + \frac{\sigma^2}{pBK^2} + \frac{p\sigma^2}{B}.$$

Now choose $p = \max(\gamma^2\delta^2, 1/K)$ which leads to

$$R_K \lesssim \frac{F}{\gamma K} + \frac{\sigma^2}{BK} + \frac{\gamma^2\delta^2\sigma^2}{B} + \frac{\sigma^2}{BK}.$$

Next set $\gamma = \min\left(\frac{1}{\delta}, \left(\frac{BF}{2\delta^2\sigma^2K}\right)^{1/3}\right)$ and the rate results in

$$\begin{aligned} R_K &\lesssim \frac{\delta F}{K} + \frac{F}{K}\left(\frac{2\delta^2\sigma^2 K}{BF}\right)^{1/3} + \frac{\sigma^2}{BK} + \left(\frac{BF}{2\delta^2\sigma^2K}\right)^{2/3}\frac{\delta^2\sigma^2}{B} + \frac{\sigma^2}{BK} \\ &\lesssim \frac{\delta F}{K} + \frac{\sigma^2}{BK} + \left(\frac{\delta\sigma F}{\sqrt{B}K}\right)^{2/3}, \end{aligned}$$

which leads to communication complexity

$$\mathcal{O}\left(\frac{\delta F}{\varepsilon} + \frac{\sigma^2}{B\varepsilon} + \frac{\delta\sigma F}{\sqrt{B}\varepsilon^{3/2}}\right).$$

Appendix I. Partial participation with averaging

Algorithm 3 SPAM-PPA

```
Input: learning rate $\gamma > 0$, starting point $x_0 \in \mathbb{R}^d$;
proximal precision level $\epsilon$; initialize $g_0 = g_{-1}$;
for $k = 0, 1, 2, \dots$ do
    Sample a subset of clients $S_k$, with size $|S_k| = B$;
    Selected clients do local SPAM updates;
    for $i \in S_k$ do
        Set $g_k^i = \nabla f_i(x_k) + (1 - p_k)(g_{k-1} - \nabla f_i(x_{k-1}))$;
        Broadcast $g_k^i$ to the server;
    end for
    $g_k = \frac{1}{B}\sum_{i \in S_k} g_k^i$ ;
    for $i \in S_k$ do
        Set $x_{k+1}^i = \text{a-prox}_\epsilon(x_k, g_k, \gamma_k, i)$;
        Broadcast $x_{k+1}^i$ to the server;
    end for
    The server aggregates the local iterates: $x_{k+1} = \frac{1}{B}\sum_{i \in S_k} x_{k+1}^i$;
end for
```

Theorem 14 (SPAM-PPA). *Suppose Assumptions 1, 2 are satisfied and the objective function f is L-smooth. If $\xi_k \sim \mathsf{Unif}(S_k)$ at every iteration, then the iterates of SPAM-PPA with $\gamma_k \le \frac{1}{4(\delta+L)}$ and $p_k = \frac{96\delta^2\gamma_k^2}{96\delta^2\gamma_k^2+B^2}$ satisfy*

$$\frac{1}{\Gamma_K}\sum_{k=0}^{K-1}\gamma_k \mathrm{E}\left[\left\|\nabla f(x_{k+1}^{\xi})\right\|^2\right] \quad\le\quad \frac{40}{\Gamma_K}\left(V_0 - \mathrm{E}\left[V_K\right]\right) + \frac{240}{\Gamma_K}\sum_{k=0}^{K-1} p_k\gamma_k\frac{\sigma^2}{B} + 7.5\epsilon^2.$$

The result of the theorem is similar to the one in Theorem 8. In fact, following the proof scheme of Corollary 9, one can derive the complexity analysis for SPAM-PPA. However, unlike previous results, we require the objective function f to be smooth.

I.1. Proof of Theorem 14

The proof follows the logic of Proposition 1. Recall that

$$V_k = f(x_k) - f_{\inf} + \frac{3\gamma_k}{2p_k - p_k^2}\|g_k - \nabla f(x_k)\|^2.$$

We start with proving a descent lemma. Recall that $\xi_k \sim \mathsf{Unif}(S_k)$, for the fixed S_k.

Lemma 15. *For an L-smooth objective f satisfying assumptions 1,2 and parameters $\gamma_k^2 \le \min\left\{\frac{1}{16(L+\delta)^2}, \frac{4p_k}{15\delta^2(1-p_k)}\right\}$, the iterates of the SPAM-PPA algorithm satisfy*

$$\mathrm{E}\left[f(x_{k+1}) - f_{\inf}\right] \le \mathrm{E}\left[f(x_k) - f_{\inf}\right] + 2\gamma_k\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] - \frac{1}{4\gamma_k}\mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - x_k\right\|^2\right]. \tag{18}$$

The proof of the lemma is deferred to Appendix J.5. Next, we estimate the second term of the Lyapunov function. Recall that

$$\begin{aligned} g_{k+1} &= \frac{1}{S_k}\sum_{i\in S_k}\left\{\nabla f_i(x_{k+1}) + (1-p_k)\left(g_k - \nabla f_i(x_k)\right)\right\} \\ &= \nabla\tilde{f}_k(x_{k+1}) + (1-p_k)\left(g_k - \nabla\tilde{f}_k(x_k)\right), \end{aligned}$$

where $\tilde{f}_k(x) := \frac{1}{S_k}\sum_{i\in S_k}\nabla f_i(x)$. Notice that $\mathrm{E}\left[\tilde{f}_k(x)\right] = f(x)$, for every fixed $x \in \mathbb{R}^d$. Furthermore, combining the convexity of the Euclidean norm and Hessian similarity (5) we deduce that the estimator $\tilde{f}_k$ satisfies the Hessian similarity condition

$$\begin{aligned} \left\|\nabla\tilde{f}_k(x) - \nabla f(x) - \nabla\tilde{f}_k(y) + \nabla f(y)\right\| &\le \frac{1}{B}\sum_{i\in S_k}\|\nabla f_i(x) - \nabla f(x) - \nabla f_i(y) + \nabla f(y)\| \\ &\le \frac{\delta}{B}\|x-y\|. \end{aligned}$$

Furthermore, Jensen's inequality implies that $\tilde{f}_k$ satisfies the bounded variance condition as well:

$$\mathrm{E}\left[\left\|\nabla\tilde{f}_k(x) - \nabla f(x)\right\|\right] \le \sigma^2/B.$$

Repeating the analysis exactly as in the proof of Lemma 11, we obtain

$$\begin{aligned}\mathrm{E}\left[\|g_{k+1}-\nabla f(x_{k+1})\|^2\right] \;\le\;& (1-p_k)^2\mathrm{E}\left[\|g_k-\nabla f(x_k)\|^2\right]\\ &+2(1-p_k)^2\frac{\delta^2}{B^2}\mathrm{E}\left[\|x_{k+1}-x_k\|^2\right]+\frac{2p_k^2\sigma^2}{B}.\end{aligned}$$

Assume now that $\xi_k \sim \mathsf{Unif}(S_k)$, for a fixed S_k. The latter means $x_{k+1} = \mathrm{E}\left[x_{k+1}^{\xi_k}\middle|\mathcal{G}_k\right]$, and subsequently, Jensen's inequality yields

$$\begin{aligned}\mathrm{E}\left[\|g_{k+1}-\nabla f(x_{k+1})\|^2\right] \le& (1-p_k)^2\mathrm{E}\left[\|g_k-\nabla f(x_k)\|^2\right]\\ &+2(1-p_k)^2\frac{\delta^2}{B^2}\mathrm{E}\left[\left\|\mathrm{E}\left[x_{k+1}^{\xi_k}\middle|\mathcal{G}_k\right]-x_k\right\|^2\right]+\frac{2p_k^2\sigma^2}{B}\\ \le& (1-p_k)^2\mathrm{E}\left[\|g_k-\nabla f(x_k)\|^2\right]\\ &+2(1-p_k)^2\frac{\delta^2}{B^2}\mathrm{E}\left[\mathrm{E}\left[\left\|x_{k+1}^{\xi_k}-x_k\right\|^2\middle|\mathcal{G}_k\right]\right]+\frac{2p_k^2\sigma^2}{B}\\ =& (1-p_k)^2\mathrm{E}\left[\|g_k-\nabla f(x_k)\|^2\right]\\ &+2(1-p_k)^2\frac{\delta^2}{B^2}\mathrm{E}\left[\left\|x_{k+1}^{\xi_k}-x_k\right\|^2\right]+\frac{2p_k^2\sigma^2}{B}.\end{aligned}$$

Now, we need to bound $\mathrm{E}\left[\left\|x_{k+1}^{\xi_k}-x_k\right\|^2\right]$ from below.

Lemma 16. *Under assumptions 1 and 2, we have the following lower bound for the iterates of* SPAM-PPA *algorithm*

$$\mathrm{E}\left[\left\|x_{k+1}^{\xi_k}-x_k\right\|^2\right] \ge \frac{\gamma_k^2}{5}\mathrm{E}\left[\left\|\nabla f(x_{k+1}^{\xi_k})\right\|^2\right]-\gamma_k^2\mathrm{E}\left[\|g_k-\nabla f(x_k)\|^2\right]-\gamma_k\epsilon^2. \tag{19}$$

The proof of the lemma can be found in Appendix J.6. Let us now bound the Lyapunov function using (18) and (19):

$$\begin{aligned}\mathrm{E}\left[V_{k+1}\right] \;\le\;& \mathrm{E}\left[f(x_k)-f_{\inf}\right]+2\gamma_k\mathrm{E}\left[\|\nabla f(x_k)-g_k\|^2\right]-\frac{1}{4\gamma_k}\mathrm{E}\left[\left\|x_{k+1}^{\xi_k}-x_k\right\|^2\right]\\ &+\alpha(1-p_k)^2\mathrm{E}\left[\|g_k-\nabla f(x_k)\|^2\right]+2\alpha(1-p_k)^2\frac{\delta^2}{B^2}\mathrm{E}\left[\left\|x_{k+1}^{\xi_k}-x_k\right\|^2\right]+\frac{2\alpha p_k^2\sigma^2}{B}\\ =\;& \mathrm{E}\left[V_k\right]+\left(2\alpha\frac{\delta^2}{B^2}(1-p_k)^2-\frac{1}{4\gamma_k}\right)\mathrm{E}\left[\left\|x_{k+1}^{\xi_k}-x_k\right\|^2\right]+\frac{2\alpha p_k^2\sigma^2}{B}\\ &+(2\gamma_k-\alpha(2p_k-p_k^2))\mathrm{E}\left[\|\nabla f(x_k)-g_k\|^2\right].\end{aligned}$$

The latter is true for every positive α. Let us now plug in the value of $\alpha=\frac{3\gamma_k}{2p_k-p_k^2}$. Then, using $\gamma \le \sqrt{\frac{B^2 p_k}{96\delta^2(1-p_k)}}$, we obtain

$$2\alpha\frac{\delta^2}{B^2}(1-p_k)^2-\frac{1}{4\gamma_k} \le \frac{6\gamma_k\delta^2}{B^2(2p_k-p_k^2)}(1-p_k)^2-\frac{1}{4\gamma_k} \le -\frac{1}{8\gamma_k}. \tag{20}$$

Hence, we have the following bound

$$\begin{aligned}
\mathrm{E}\left[V_{k+1}\right] &\leq \mathrm{E}\left[V_k\right] - \frac{1}{8\gamma_k}\mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - x_k\right\|^2\right] - \gamma_k \mathrm{E}\left[\left\|\nabla f(x_k) - g_k\right\|^2\right] + \frac{6p_k\gamma_k\sigma^2}{B(2-p_k)} \\
&\overset{(19)}{\leq} \mathrm{E}\left[V_k\right] - \frac{1}{8\gamma_k}\left(\frac{\gamma_k^2}{5}\mathrm{E}\left[\left\|\nabla f(x_{k+1}^{\xi_k})\right\|^2\right] - \gamma_k^2\mathrm{E}\left[\left\|g_k - \nabla f(x_k)\right\|^2\right] - \gamma_k^2\epsilon^2\right) \\
&\quad -\gamma_k \mathrm{E}\left[\left\|\nabla f(x_k) - g_k\right\|^2\right] + \frac{6p_k\gamma_k\sigma^2}{B} \\
&\leq \mathrm{E}\left[V_k\right] - \frac{\gamma_k}{40}\mathrm{E}\left[\left\|\nabla f(x_{k+1}^{\xi_k})\right\|^2\right] - \frac{7\gamma_k}{8}\mathrm{E}\left[\left\|\nabla f(x_k) - g_k\right\|^2\right] + \frac{6p_k\gamma_k\sigma^2}{B} + \frac{\gamma_k\epsilon^2}{8} \\
&\leq \mathrm{E}\left[V_k\right] - \frac{\gamma_k}{40}\mathrm{E}\left[\left\|\nabla f(x_{k+1}^{\xi_k})\right\|^2\right] + \frac{6p_k\gamma_k\sigma^2}{B} + \frac{\gamma_k\epsilon^2}{8}.
\end{aligned}$$

Thus, we have

$$\frac{1}{\Gamma_K}\sum_{k=0}^{K-1}\gamma_k \mathrm{E}\left[\left\|\nabla f(x_{k+1}^{\xi_k})\right\|^2\right] \leq \frac{40}{\Gamma_K}\left(V_0 - \mathrm{E}\left[V_K\right]\right) + \frac{240}{\Gamma_K}\sum_{k=0}^{K-1} p_k\gamma_k\frac{\sigma^2}{B} + 7.5\epsilon^2.$$

This concludes the proof of the theorem.

Appendix J. Proofs of the technical lemmas

J.1. Proof of Lemma 10

By the main theorem of Calculus, we have

$$\begin{aligned}
f(x_{k+1}) - f(x_k) &= \int_0^1 \left\langle \nabla f(\underbrace{x_k + \tau(x_{k+1} - x_k)}_{:=x(\tau)}), x_{k+1} - x_k \right\rangle d\tau, \\
f_{\xi_k}(x_{k+1}) - f_{\xi_k}(x_k) &= \int_0^1 \left\langle \nabla f_{\xi_k}(\underbrace{x_k + \tau(x_{k+1} - x_k)}_{:=x(\tau)}), x_{k+1} - x_k \right\rangle d\tau
\end{aligned}$$

Therefore the difference in function value can be bounded as follows:

$$\begin{aligned}
f(x_{k+1}) - f(x_k) &= f_{\xi_k}(x_{k+1}) - f_{\xi_k}(x_k) \\
&\quad + \int_0^1 \left\langle \nabla f(x(\tau)) - \nabla f_{\xi_k}(x(\tau)), x_{k+1} - x_k \right\rangle d\tau \\
&= f_{\xi_k}(x_{k+1}) - f_{\xi_k}(x_k) + \left\langle g_k - \nabla f_{\xi_k}(x_k), x_{k+1} - x_k \right\rangle \\
&\quad + \int_0^1 \left\langle \nabla f(x(\tau)) - \nabla f_{\xi_k}(x(\tau)) - g_k + \nabla f_{\xi_k}(x_k), x_{k+1} - x_k \right\rangle d\tau \\
&\leq -\frac{1}{2\gamma_k}\left\|x_{k+1} - x_k\right\|^2 + \left\langle \nabla f(x_k) - g_k, x_{k+1} - x_k \right\rangle \\
&\quad + \int_0^1 \left\langle \nabla f(x(\tau)) - \nabla f_{\xi_k}(x(\tau)) - \nabla f(x_k) + \nabla f_{\xi_k}(x_k), x_{k+1} - x_k \right\rangle d\tau.
\end{aligned}$$

The last inequality is due to

$$f_{\xi_k}(x_{k+1}) + \langle g_k - \nabla f_{\xi_k}(x_k), x_{k+1} - x_k\rangle + \frac{1}{2\gamma_k}\|x_{k+1} - x_k\|^2 \le f_{\xi_k}(x_k), \tag{21}$$

which is a direct consequence of $x_{k+1} = \arg\min_x \left\{ f_{\xi_k}(x) + \langle g_k - \nabla f_{\xi_k}(x_k), x - x_k\rangle + \frac{1}{2\gamma_k}\|x - x_k\|^2 \right\}$. Let us now apply Cauchy-Schwartz inequality to bound both scalar products:

$$\begin{aligned}
f(x_{k+1}) - f(x_k) \quad &\le \quad -\frac{1}{2\gamma_k}\|x_{k+1} - x_k\|^2 + 2\gamma_k\|\nabla f(x_k) - g_k\|^2 + \frac{1}{8\gamma_k}\|x_{k+1} - x_k\|^2 \\
&\quad + \int_0^1 \|\nabla f(x(\tau)) - \nabla f_{\xi_k}(x(\tau)) - \nabla f(x_k) + \nabla f_{\xi_k}(x_k)\|\|x_{k+1} - x_k\| d\tau \\
&\overset{(5)}{\le} \quad -\frac{1}{2\gamma_k}\|x_{k+1} - x_k\|^2 + 2\gamma_k\|\nabla f(x_k) - g_k\|^2 + \frac{1}{8\gamma_k}\|x_{k+1} - x_k\|^2 \\
&\quad +\delta \int_0^1 \|x(\tau) - x_k\|\|x_{k+1} - x_k\| d\tau \\
&= \quad -\frac{1}{2\gamma_k}\|x_{k+1} - x_k\|^2 + 2\gamma_k\|\nabla f(x_k) - g_k\|^2 + \frac{1}{8\gamma_k}\|x_{k+1} - x_k\|^2 \\
&\quad +\frac{\delta}{2}\|x_{k+1} - x_k\|^2 \\
&\overset{\gamma_k \le \frac{1}{4\delta}}{\le} \quad -\frac{1}{4\gamma_k}\|x_{k+1} - x_k\|^2 + 2\gamma_k\|\nabla f(x_k) - g_k\|^2.
\end{aligned}$$

Thus, we have

$$f(x_{k+1}) - f_{\inf} \le f(x_k) - f_{\inf} - \frac{1}{4\gamma_k}\|x_{k+1} - x_k\|^2 + 2\gamma_k\|\nabla f(x_k) - g_k\|^2. \tag{22}$$

This concludes the proof of the lemma.

J.2. Proof of Lemma 11

Recall that $g_{k+1} = \nabla f_{\xi_{k+1}}(x_{k+1}) + (1 - p_k)\left(g_k - \nabla f_{\xi_{k+1}}(x_k)\right)$. We define $\mathcal{F}_k := \{x_{k+1}, x_k, g_k\}$. Then,

$$\begin{aligned}
& \mathrm{E}\left[\|g_{k+1} - \nabla f(x_{k+1})\|^2 \middle| \mathcal{F}_k\right] \\
= \quad & \mathrm{E}\left[\left\|\nabla f_{\xi_{k+1}}(x_{k+1}) + (1 - p_k)\left(g_k - \nabla f_{\xi_{k+1}}(x_k)\right) - \nabla f(x_{k+1})\right\|^2 \middle| \mathcal{F}_k\right] \\
= \quad & \mathrm{E}\left[\left\|\nabla f_{\xi_{k+1}}(x_{k+1}) - \nabla f(x_{k+1}) + (1 - p_k)\left(\nabla f(x_k) - \nabla f_{\xi_{k+1}}(x_k)\right)\right\|^2 \middle| \mathcal{F}_k\right] \\
& +(1 - p_k)^2\|g_k - \nabla f(x_k)\|^2.
\end{aligned}$$

The last equality $(*)$ is due to the bias-variance formula and the fact that ξ_{k+1} is independent from $\mathcal{F}_k$ and that the stochastic gradients are unbiased. Using the Cauchy-Schwartz inequality we deduce the following bound for the first term on the right-hand side, where $\alpha > 0$ is an arbitrary constant:

$$\begin{aligned}
&\mathrm{E}\left[\left\|\nabla f_{\xi_{k+1}}(x_{k+1}) - \nabla f(x_{k+1}) + (1-p_k)\left(\nabla f(x_k) - \nabla f_{\xi_{k+1}}(x_k)\right)\right\|^2 \mid \mathcal{F}_k\right] \\
&\quad = \mathrm{E}\left[\|p_k\left(\nabla f_{\xi_{k+1}}(x_{k+1}) - \nabla f(x_{k+1})\right)\right. && (23)\\
&\qquad \left.+ (1-p_k)\left(\nabla f_{\xi_{k+1}}(x_{k+1}) - \nabla f(x_{k+1}) + \nabla f(x_k) - \nabla f_{\xi_{k+1}}(x_k)\right)\|^2 \mid \mathcal{F}_k\right] \\
&\quad \le (1+\alpha)p_k^2 \mathrm{E}\left[\left\|\nabla f_{\xi_{k+1}}(x_{k+1}) - \nabla f(x_{k+1})\right\|^2 \middle| \mathcal{F}_k\right] && (24)\\
&\qquad + (1+\alpha^{-1})(1-p_k)^2 \mathrm{E}\left[\left\|\nabla f_{\xi_{k+1}}(x_{k+1}) - \nabla f(x_{k+1}) + \nabla f(x_k) - \nabla f_{\xi_{k+1}}(x_k)\right\|^2 \middle| \mathcal{F}_k\right].
\end{aligned}$$

We apply (3) and (5) to bound, respectively, the first term and the second term on the right-hand side of (23):

$$\begin{aligned}
&\mathrm{E}\left[\left\|\nabla f_{\xi_{k+1}}(x_{k+1}) - \nabla f(x_{k+1}) + (1-p_k)\left(\nabla f(x_k) - \nabla f_{\xi_{k+1}}(x_k)\right)\right\|^2 \middle| \mathcal{F}_k\right] \\
\le \quad &(1+\alpha)p_k^2\sigma^2 + (1+\alpha^{-1})(1-p_k)^2\delta^2\|x_{k+1} - x_k\|^2.
\end{aligned}$$

Taking $\alpha = 1$, we obtain the following

$$\mathrm{E}\left[\|g_{k+1} - \nabla f(x_{k+1})\|^2 \middle| \mathcal{F}_k\right] \le (1-p_k)^2\|g_k - \nabla f(x_k)\|^2 + 2(1-p_k)^2\delta^2\|x_{k+1} - x_k\|^2 + 2p_k^2\sigma^2.$$

This concludes the proof of the lemma.

J.3. Proof of Lemma 12

By the definition of x_{k+1}, we have

$$\begin{aligned}
\|x_{k+1} - x_k\|^2 &= \gamma_k^2\|\nabla f_{\xi_k}(x_{k+1}) + g_k - \nabla f_{\xi_k}(x_k)\|^2 \\
&= \gamma_k^2\|\nabla f(x_{k+1}) + g_k - \nabla f(x_k) + \nabla f_{\xi_k}(x_{k+1}) - \nabla f(x_{k+1}) - \nabla f_{\xi_k}(x_k) + \nabla f(x_k)\|^2 \\
&\ge \frac{\gamma_k^2}{3}\|\nabla f(x_{k+1})\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2 \\
&\quad -\gamma_k^2\|\nabla f_{\xi_k}(x_{k+1}) - \nabla f(x_{k+1}) - \nabla f_{\xi_k}(x_k) + \nabla f(x_k)\|^2 \\
&\ge \frac{\gamma_k^2}{3}\|\nabla f(x_{k+1})\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2 - \gamma_k^2\delta^2\|x_{k+1} - x_k\|^2,
\end{aligned}$$

where we used a variant of Jensen's inequality $3(a^2+b^2+c^2) \ge (a+b+c)^2$, for $a, b, c > 0$. Therefore, we have

$$\begin{aligned}
\|x_{k+1} - x_k\|^2 &\ge \frac{1}{1+\gamma_k^2\delta^2}\left(\frac{\gamma_k^2}{3}\|\nabla f(x_{k+1})\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2\right) \\
&\ge \frac{16}{17}\left(\frac{\gamma_k^2}{3}\|\nabla f(x_{k+1})\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2\right) \\
&\ge \frac{\gamma_k^2}{4}\|\nabla f(x_{k+1})\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2.
\end{aligned}$$

Thus, we have

$$\mathrm{E}\left[\|x_{k+1} - x_k\|^2\right] \ge \frac{\gamma_k^2}{4}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] - \gamma_k^2\mathrm{E}\left[\|g_k - \nabla f(x_k)\|^2\right].$$

This concludes the proof of the lemma.

J.4. Proof of Lemma 13

Let $x_{k+1} = \text{a-prox}_\epsilon\left(x_k, g_k, \gamma_k, \xi_k\right)$. Then, from the definition of the function ϕ_k (7), we have

$$\begin{aligned}\|x_{k+1} - x_k\|^2 &= \gamma_k^2\|\nabla f_{\xi_k}(x_{k+1}) + g_k - \nabla f_{\xi_k}(x_k) - \nabla\phi_k(x_{k+1})\|^2 \\ &\geq \gamma_k^2\left(\frac{1}{4}\|\nabla f(x_{k+1})\|^2 - \|g_k - \nabla f(x_k)\|^2 - \delta^2\|x_{k+1} - x_k\|^2 - \|\nabla\phi_k(x_{k+1})\|^2\right).\end{aligned}$$

Since $\|a_1 + a_2 + a_3 + a_4\|^2 \leq 4\left(\|a_1\|^2 + \|a_2\|^2 + \|a_3\|^2 + \|a_4\|^2\right)$ for any vectors $a_i \in \mathbb{R}^d$, which implies $\|a_4\|^2 \geq \frac{1}{4}\|a_1 + a_2 + a_3 + a_4\|^2 - \|a_1\|^2 - \|a_2\|^2 - \|a_3\|^2$ and $\mathrm{E}\left[\|\nabla\phi_k(x_{k+1})\|^2\right] \leq \epsilon^2$, we deduce

$$\|x_{k+1} - x_k\|^2 \geq \frac{\gamma_k^2}{1 + \gamma_k^2\delta^2}\left(\frac{1}{4}\|\nabla f(x_{k+1})\|^2 - \|g_k - \nabla f(x_k)\|^2 - \epsilon\right).$$

Therefore, we have

$$\begin{aligned}\|x_{k+1} - x_k\|^2 &\geq \frac{1}{1 + \gamma_k^2\delta^2}\left(\frac{\gamma_k^2}{4}\|\nabla f(x_{k+1})\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2 - \gamma_k^2\epsilon^2\right) \\ &\geq \frac{16}{17}\left(\frac{\gamma_k^2}{4}\|\nabla f(x_{k+1})\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2 - \gamma_k^2\epsilon^2\right) \\ &\geq \frac{\gamma_k^2}{5}\|\nabla f(x_{k+1})\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2 - \gamma_k^2\epsilon^2.\end{aligned}$$

Taking expectations on both sides leads to

$$\mathrm{E}\left[\|x_{k+1} - x_k\|^2\right] \geq \frac{\gamma_k^2}{5}\mathrm{E}\left[\|\nabla f(x_{k+1})\|^2\right] - \gamma_k^2\mathrm{E}\left[\|g_k - \nabla f(x_k)\|^2\right] - \gamma_k^2\epsilon^2.$$

This concludes the proof of the lemma.

J.5. Proof of Lemma 15

Recalling that $x_{k+1}^{\xi_k} = \text{a-prox}_\epsilon\left(x_k, g_k, \gamma_k, \xi_k\right)$, we have

$$f_{\xi_k}(x_{k+1}^{\xi_k}) + \left\langle g_k - \nabla f_{\xi_k}(x_k), x_{k+1}^{\xi_k} - x_k\right\rangle + \frac{1}{2\gamma_k}\|x_{k+1}^{\xi_k} - x_k\|^2 \leq f_{\xi_k}(x_k).$$

Similar to the proof of Proposition 1 we start with

$$\begin{aligned}f(x_{k+1}) - f(x_k) &= \int_0^1 \left\langle \nabla f(\underbrace{x_k + \tau(x_{k+1} - x_k)}_{:=x(\tau)}), x_{k+1} - x_k\right\rangle d\tau, \\ f_{\xi_k}(x_{k+1}^{\xi_k}) - f_{\xi_k}(x_k) &= \int_0^1 \left\langle \nabla f_{\xi_k}(\underbrace{x_k + \tau(x_{k+1}^{\xi_k} - x_k)}_{:=x^{\xi_k}(\tau)}), x_{k+1}^{\xi_k} - x_k\right\rangle d\tau.\end{aligned}$$

Thus, we have

$$
\begin{aligned}
f(x_{k+1}) - f(x_k) &= f_{\xi_k}(x_{k+1}^{\xi_k}) - f_{\xi_k}(x_k) \\
&\quad + \int_0^1 \langle \nabla f(x(\tau)), x_{k+1} - x_k \rangle \, d\tau \\
&\quad + \int_0^1 \left\langle -\nabla f_{\xi_k}(x^{\xi_k}(\tau)), x_{k+1}^{\xi_k} - x_k \right\rangle d\tau \\
&= f_{\xi_k}(x_{k+1}^{\xi_k}) - f_{\xi_k}(x_k) + \left\langle g_k - \nabla f_{\xi_k}(x_k), x_{k+1}^{\xi_k} - x_k \right\rangle \\
&\quad + \int_0^1 \langle \nabla f(x(\tau)), x_{k+1} - x_k \rangle \, d\tau \\
&\quad + \int_0^1 \left\langle -\nabla f_{\xi_k}(x^{\xi_k}(\tau)) - g_k + \nabla f_{\xi_k}(x_k), x_{k+1}^{\xi_k} - x_k \right\rangle d\tau.
\end{aligned}
$$

Applying the descent property of a-prox (see Definition 1), we deduce the following:

$$
\begin{aligned}
f(x_{k+1}) - f(x_k) &\leq -\frac{1}{2\gamma_k}\left\| x_{k+1}^{\xi_k} - x_k \right\|^2 + \left\langle \nabla f(x_k) - g_k, x_{k+1}^{\xi_k} - x_k \right\rangle \\
&\quad + \int_0^1 \langle \nabla f(x(\tau)), x_{k+1} - x_k \rangle \, d\tau \\
&\quad + \int_0^1 \left\langle -\nabla f_{\xi_k}(x^{\xi_k}(\tau)) - \nabla f(x_k) + \nabla f_{\xi_k}(x_k), x_{k+1}^{\xi_k} - x_k \right\rangle d\tau.
\end{aligned}
$$

Let us take expectation from both sides conditioned to $\mathcal{G}_k = \{x_k, x_{k+1}, S_k, g_k\}$. In other words, we take expectation with respect to the random index ξ_k chosen uniformly from the already chosen S_k:

$$
\begin{aligned}
f(x_{k+1}) - f(x_k) &\leq \mathrm{E}\left[-\frac{1}{2\gamma_k}\left\| x_{k+1}^{\xi_k} - x_k \right\|^2 + \left\langle \nabla f(x_k) - g_k, x_{k+1}^{\xi_k} - x_k \right\rangle \mid \mathcal{G}_k \right] \\
&\quad + \mathrm{E}\left[\int_0^1 \langle \nabla f(x(\tau)), x_{k+1} - x_k \rangle \, d\tau \mid \mathcal{G}_k \right] \\
&\quad + \mathrm{E}\left[\int_0^1 \left\langle -\nabla f_{\xi_k}(x^{\xi_k}(\tau)) - \nabla f(x_k) + \nabla f_{\xi_k}(x_k), x_{k+1}^{\xi_k} - x_k \right\rangle d\tau \middle| \mathcal{G}_k \right] \\
&= \mathrm{E}\left[-\frac{1}{2\gamma_k}\left\| x_{k+1}^{\xi_k} - x_k \right\|^2 \mid \mathcal{G}_k \right] + \langle \nabla f(x_k) - g_k, x_{k+1} - x_k \rangle \\
&\quad + \mathrm{E}\left[\int_0^1 \left\langle \nabla f(x(\tau)) - \nabla f(x_k) - \nabla f_{\xi_k}(x^{\xi_k}(\tau)) + \nabla f_{\xi_k}(x_k), x_{k+1}^{\xi_k} - x_k \right\rangle d\tau \mid \mathcal{G}_k \right].
\end{aligned}
$$

Here the last equality is due to the fact that ξ_k is independent of $\mathcal{G}_k$ and $x_{k+1} = \mathrm{E}\left[x_{k+1}^{\xi_k} \mid \mathcal{G}_k\right]$. Therefore, applying Cauchy-Schwartz inequality

$$\begin{aligned}
f(x_{k+1}) - f(x_k) \le \mathrm{E}\left[-\frac{1}{2\gamma_k}\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right] + \langle \nabla f(x_k) - g_k, x_{k+1} - x_k\rangle \\
+ \mathrm{E}\left[\int_0^1 \left\langle \nabla f(x(\tau)) - \nabla f(x^{\xi_k}(\tau)), x_{k+1}^{\xi_k} - x_k \right\rangle d\tau \mid \mathcal{G}_k\right] \\
+ \mathrm{E}\left[\int_0^1 \left\langle \nabla f(x^{\xi_k}(\tau)) - \nabla f(x_k) - \nabla f_{\xi_k}(x^{\xi_k}(\tau)) + \nabla f_{\xi_k}(x_k), x_{k+1}^{\xi_k} - x_k \right\rangle d\tau \mid \mathcal{G}_k\right] \\
\le \mathrm{E}\left[-\frac{1}{2\gamma_k}\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right] + \langle \nabla f(x_k) - g_k, x_{k+1} - x_k\rangle \\
+ \mathrm{E}\left[\int_0^1 \left\|\nabla f(x(\tau)) - \nabla f(x^{\xi_k}(\tau))\right\| \left\|x_{k+1}^{\xi_k} - x_k\right\| d\tau \mid \mathcal{G}_k\right] \\
+ \mathrm{E}\left[\int_0^1 \left\|\nabla f(x^{\xi_k}(\tau)) - \nabla f(x_k) - \nabla f_{\xi_k}(x^{\xi_k}(\tau)) + \nabla f_{\xi_k}(x_k)\right\| \left\|x_{k+1}^{\xi_k} - x_k\right\| d\tau \mid \mathcal{G}_k\right].
\end{aligned}$$

Applying Cauchy-Schwartz inequality once again we deduce

$$\begin{aligned}
f(x_{k+1}) - f(x_k) \ \le\ & \mathrm{E}\left[-\frac{1}{2\gamma_k}\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right] + \frac{C}{2}\|\nabla f(x_k) - g_k\|^2 + \frac{1}{2C}\|x_{k+1} - x_k\|^2 \\
& + \mathrm{E}\left[\int_0^1 L\left\|x(\tau) - x^{\xi_k}(\tau)\right\| \left\|x_{k+1}^{\xi_k} - x_k\right\| d\tau \mid \mathcal{G}_k\right] \\
& + \mathrm{E}\left[\int_0^1 \delta\left\|x^{\xi_k}(\tau) - x_k\right\| \left\|x_{k+1}^{\xi_k} - x_k\right\| d\tau \mid \mathcal{G}_k\right] \\
\le\ & \mathrm{E}\left[-\frac{1}{2\gamma_k}\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right] + \frac{C}{2}\|\nabla f(x_k) - g_k\|^2 + \frac{1}{2C}\|x_{k+1} - x_k\|^2 \\
& + \mathrm{E}\left[\int_0^1 L\tau\left\|x_{k+1} - x_{k+1}^{\xi_k}\right\| \left\|x_{k+1}^{\xi_k} - x_k\right\| d\tau \mid \mathcal{G}_k\right] \\
& + \mathrm{E}\left[\int_0^1 \delta\tau\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 d\tau \mid \mathcal{G}_k\right].
\end{aligned}$$

Computing the integral with respect to τ we obtain

$$\begin{aligned}
f(x_{k+1}) - f(x_k) \ \le\ & \mathrm{E}\left[-\frac{1}{2\gamma_k}\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right] + \frac{C}{2}\|\nabla f(x_k) - g_k\|^2 + \frac{1}{2C}\|x_{k+1} - x_k\|^2 \\
& + \frac{L}{2}\mathrm{E}\left[\left\|x_{k+1} - x_{k+1}^{\xi_k}\right\| \left\|x_{k+1}^{\xi_k} - x_k\right\| \mid \mathcal{G}_k\right] + \frac{\delta}{2}\mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right] \\
\le\ & \mathrm{E}\left[-\frac{1}{2\gamma_k}\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right] + \frac{C}{2}\|\nabla f(x_k) - g_k\|^2 + \frac{1}{2C}\|x_{k+1} - x_k\|^2 \\
& + \frac{L}{4}\mathrm{E}\left[\left\|x_{k+1} - x_{k+1}^{\xi_k}\right\|^2 \mid \mathcal{G}_k\right] + \frac{2\delta + L}{4}\mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right].
\end{aligned}$$

Recall again that $x_{k+1} = \mathrm{E}\left[x_{k+1}^{\xi_k} \mid \mathcal{G}_k\right]$, thus $x_{k+1} = \arg\min_{a\in\mathbb{R}^d} \mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - a\right\|^2 \mid \mathcal{G}_k\right]$. Therefore,

$$\mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - x_{k+1}\right\|^2 \mid \mathcal{G}_k\right] \leq \mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right].$$

Furthermore,

$$\|x_{k+1} - x_k\|^2 = \left\|\mathrm{E}\left[x_{k+1}^{\xi_k} \mid \mathcal{G}_k\right] - x_k\right\|^2 \leq \mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right].$$

Combining these two bounds, we deduce

$$\begin{aligned} f(x_{k+1}) - f(x_k) \quad &\leq \quad \frac{C}{2}\|\nabla f(x_k) - g_k\|^2 \\ &\quad + \left(\frac{1}{2C} + \frac{\delta + L}{2} - \frac{1}{2\gamma_k}\right) \mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right]. \end{aligned}$$

The previous bound is true for every positive value of C. Thus, it is true also for $C = 4\gamma_k$. Taking into account that $\gamma_k < \frac{1}{4(L+\delta)}$, we get

$$\frac{1}{2C} + \frac{\delta + L}{2} - \frac{1}{2\gamma_k} \leq \frac{1}{8\gamma_k} + \frac{1}{8\gamma_k} - \frac{1}{2\gamma_k} = -\frac{1}{4\gamma_k}.$$

Therefore,

$$f(x_{k+1}) - f(x_k) \quad \leq \quad 2\gamma_k\|\nabla f(x_k) - g_k\|^2 - \frac{1}{4\gamma_k}\mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \mid \mathcal{G}_k\right].$$

Thus, taking full expectation on both sides we have

$$\mathrm{E}\left[f(x_{k+1}) - f_{\inf}\right] \leq \mathrm{E}\left[f(x_k) - f_{\inf}\right] + 2\gamma_k\mathrm{E}\left[\|\nabla f(x_k) - g_k\|^2\right] - \frac{1}{4\gamma_k}\mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - x_k\right\|^2\right].$$

This concludes the proof.

J.6. Proof of Lemma 16

By the definition of $x_{k+1}^{\xi_k}$, for every $\xi \in S_k$ we have

$$\begin{aligned} \left\|x_{k+1}^{\xi_k} - x_k\right\|^2 \quad &= \quad \gamma_k^2\left\|\nabla f_{\xi_k}(x_{k+1}^{\xi_k}) + g_k - \nabla f_{\xi_k}(x_k) - \nabla\phi_k(x_{k+1})\right\|^2 \\ &= \quad \gamma_k^2\left\|\nabla f(x_{k+1}^{\xi_k}) + g_k - \nabla f(x_k) + \nabla f_{\xi_k}(x_{k+1}^{\xi_k}) \right. \\ &\qquad \left. -\nabla f(x_{k+1}^{\xi_k}) - \nabla f_{\xi_k}(x_k) + \nabla f(x_k) - \nabla\phi_k(x_{k+1})\right\|^2 \\ &\geq \quad \frac{\gamma_k^2}{4}\left\|\nabla f(x_{k+1}^{\xi_k})\right\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2 \\ &\qquad -\gamma_k^2\left\|\nabla f_{\xi_k}(x_{k+1}^{\xi_k}) - \nabla f(x_{k+1}^{\xi_k}) - \nabla f_{\xi_k}(x_k) + \nabla f(x_k)\right\|^2 - \gamma_k^2\epsilon^2 \\ &\geq \quad \frac{\gamma_k^2}{4}\left\|\nabla f(x_{k+1}^{\xi_k})\right\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2 - \gamma_k^2\delta^2\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 - \gamma_k^2\epsilon^2. \end{aligned}$$

The third inequality is due to Cauchy-Schwartz and the second property of the approximate proximal operator (See Definition 1). Therefore, we have

$$
\begin{aligned}
\left\|x_{k+1}^{\xi_k} - x_k\right\|^2 &\geq \frac{1}{1+\gamma_k^2\delta^2}\left(\frac{\gamma_k^2}{4}\left\|\nabla f(x_{k+1}^{\xi_k})\right\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2 - \gamma_k^2\epsilon^2\right) \\
&\geq \frac{16}{17}\left(\frac{\gamma_k^2}{4}\left\|\nabla f(x_{k+1}^{\xi_k})\right\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2 - \gamma_k^2\epsilon^2\right) \\
&\geq \frac{\gamma_k^2}{5}\left\|\nabla f(x_{k+1}^{\xi_k})\right\|^2 - \gamma_k^2\|g_k - \nabla f(x_k)\|^2 - \gamma_k^2\epsilon^2.
\end{aligned}
$$

We deduce

$$
\mathrm{E}\left[\left\|x_{k+1}^{\xi_k} - x_k\right\|^2\right] \geq \frac{\gamma_k^2}{5}\mathrm{E}\left[\left\|\nabla f(x_{k+1}^{\xi_k})\right\|^2\right] - \gamma_k^2\mathrm{E}\left[\|g_k - \nabla f(x_k)\|^2\right] - \gamma_k^2\epsilon^2.
$$

This concludes the proof of the lemma.

Appendix K. Experimental details

We provide additional details on the experimental settings from Section E.

Consider a distributed ridge regression problem defined as

$$
f(x) = \mathrm{E}_\xi\left[\|A_\xi x - y_\xi\|^2\right] + \frac{\lambda}{2}\|x\|^2, \tag{25}
$$

where ξ is uniform random variable over $\{1,\ldots,n\}$ for $n = 10, \lambda = 0.1$. We follow a similar to [21] procedure for synthetic data generation, which allows us to calculate and control Hessian similarity δ. Namely, a random matrix $A_0 \in \mathbb{R}^{d\times d}$ ($d = 100$) is generated with entries from a standard Gaussian distribution $\mathcal{N}(0,1)$. Then we obtain $A = A_0A_0^\top$ (to ensure symmetry) and set $A'_\xi = A + B_\xi$ by adding a random symmetric matrix B_ξ (generated similarly to A). Afterwards we modify $A_\xi = A'_\xi + I\lambda_{\min}(A'_\xi)$ by adding an identity matrix I times minimum eigenvalue to guarantee $A_\xi \succeq 0$. Entries of vectors $y_\xi \in \mathbb{R}^d$, and initialization $x_0 \in \mathbb{R}^d$ are generated from a standard Gaussian distribution $\mathcal{N}(0,1)$.

In case of inexact proximal point computation (1/10 local steps) local subproblems (7) are solved with gradient descent.

Simulations were performed on a machine with 24 Intel(R) Xeon(R) Gold 6246 CPU @ 3.30 GHz.