SPAM: SPPM and MVR for Non-convex Cross-Device Federated Learning

Laboratoire des Signaux et Systèmes, 2025

Avetik Karagulyan (CNRS / L2S / Université Paris-Saclay)

Based on a joint paper with E. Shulgin (KAUST), A. Sadiev (KAUST), P. Richtárik (KAUST)

1. Federated Learning

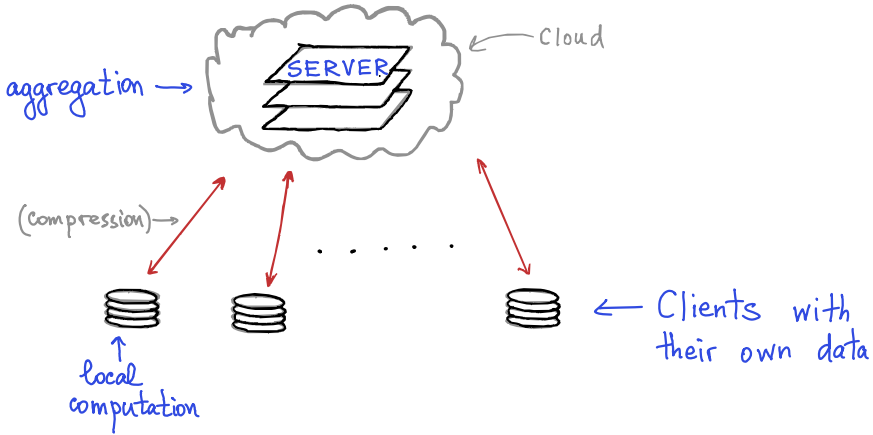
Federated learning

"Federated learning is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each client's raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective." - [KMA+21]

Nowadays, cross-device FL mechanisms are widely used:

- Medical research [CKLT18, BCM+18];
- Distributed systems [XIZ+23];
- Gboard mobile keyboard, Android messages, Apple's Siri [EPK14, Pic19].

Scheme


Cloud
SERVER
aggregation
(compression)
Clients with their own data
local computation

Structure of FL methods

Thus, federated learning consists of three key components:

- Server update
- Broadcasting (compression)
- Local computation

Structure of FL methods

Thus, federated learning consists of three key components:

- **Server update**
- Broadcasting (compression)
- **Local computation**

Simplest setting

Standard FL setting:

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

- 1 server, n clients
- Each f_i is stored on the client i.
- The clients in parallel do local computation (local `GD`) and broadcast to the server.
- Server aggregates and sends back the new iterate in parallel.

Distributed Gradient Descent

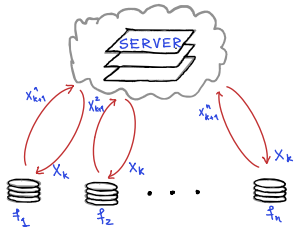
The famous gradient descent algorithm can be implemented in this way.

Algorithm 1: Distributed Gradient Descent

Input: Stepsizes $\gamma_k > 0$ for $k \geq 0$, starting point $x_0 \in \mathbb{R}^d$

for $k = 0, 1, 2, \ldots$ **do**

- The server broadcasts the current iterate x_k.
- The i-th client computes the local step: $x_{k+1}^i = x_k - \gamma_k \nabla f_i(x_k)$.
- Broadcasts to the server.
- The server aggregates the iterates: $x_{k+1} = \frac{1}{n} \sum_{i=1}^n x_{k+1}^i$.



When d is large, **communication complexity becomes the bottleneck.**

Communication complexity

Definition (Informal)

$$\texttt{Communication Complexity} = (\#\ \texttt{of bits}) \times (\#\ \texttt{of communication rounds}).$$

There are two general ways to reduce communication complexity:

- Compression of the gradients or the iterates: [RSF21, SHR21, SSR22].

  Examples: Rand-k, Top-k, quantization, sign compressors etc.;

- **Local methods:** Do more/better local computations, so that the number of communication rounds is reduced. [MMSR22, MSR22].

2. The mathematical framework of the cross-device setting

Cross-device setting

This paper focuses on cross-device training:

- Clients are mobile or IoT devices [KJK+21].
- The number of clients n is large (billions).
- Here, one cannot hope to have access to full gradient at any time.
- Thus, finite-sum formulation is not suitable for this setting. It is applicable to other FL settings, such as cross-silo training, where the number of clients is moderate.

Hardware Heterogeneity

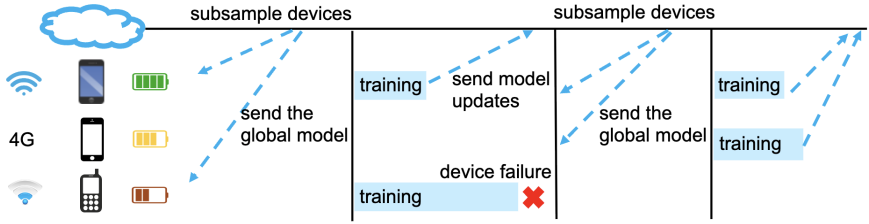


Figure: Data heterogeneity as depicted in [LSTS20].

- Asyncronous training

Data Heterogeneity

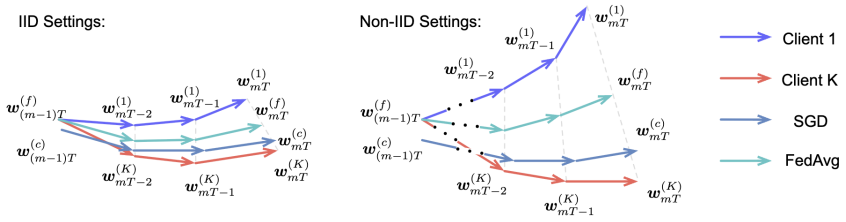


Figure: The divergence of `SGD` iterates for homogeneous and heterogeneous clients for the cross-entropy loss for neural networks. Courtesy of [ZLL+18].

Statistical/Data heterogeneity of the clients may affect learning [AASC19, SMAT22]:

- personalization, recommendation, fraud detection, etc.
- since traditional ML training algorithms are designed for central or distributed computation environments where data partitioning can be tightly controlled.

Notation

- ∇f for the gradient;
- $\|\cdot\|$ for the Euclidean norm;
- $\mathbb{E}[\cdot]$ for the expectation.
- $\mathrm{Unif}(S)$ denotes uniform distribution over the discrete set S.
- Index i is used for a non-random client
- $\xi \sim \mathcal{D}$ is used for a randomly selected client.

3. Assumptions and Background

Mathematical formulation

Instead, we study the following stochastic optimization problem:

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

where f_ξ may be non-convex.

- Here, f_ξ corresponds to the loss of client ξ on its local data [KJK+21].
- We cannot have access to the full function f, nor its gradient;
- Each client participates in the training process only a few times or maybe once.
- We assume that the gradient and the expectation are interchangeable, meaning

$$\mathbb{E}_{\xi \sim \mathcal{D}}[\nabla f_\xi(x)] = \nabla f(x), \quad \text{for} \quad \forall x \in \mathbb{R}^d.$$

.

Bounded variance

We aim to solve the stochastic optimization problem

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

Here, f_ξ and consequently f are potentially non-convex.

Assumption (Bounded variance)

We assume there exists $\sigma \geq 0$ such that for any $x \in \mathbb{R}^d$

$$\mathbb{E}_{\xi \sim \mathcal{D}}\left[\|\nabla f_\xi(x) - \nabla f(x)\|^2\right] \leq \sigma^2.$$

- This is a standard assumption in stochastic optimization.

Second-order heterogeneity

Assumption (Hessian similarity)

Assume there exists $\delta \geq 0$ such that for any i and $x, y \in \mathbb{R}^d$

$$\|\nabla f_i(x) - \nabla f(x) - \nabla f_i(y) + \nabla f(y)\| \leq \delta \|x - y\|. \tag{1}$$

- Some prior work consider $\|\nabla f_i(x) - \nabla f(x)\| \leq \delta$ instead of these two assumptions, but this is rather restrictive.
- [KJ22] showed that for $f_i \in C^2(\mathbb{R}^d)$, (1) is equivalent to $\left\|\nabla^2 f_i(x) - \nabla^2 f(x)\right\|_{op} \leq \delta$.
- (1) is satisfied for ridge regression with NNs [MLFV23].
- (1) replaces the smoothness assumption.

Proximal point method (PPM)

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

$$\text{prox}_g(x) := \arg\min_{y \in \mathbb{R}^d} \left\{ g(y) + \frac{1}{2}\|x - y\|^2 \right\}.$$

Algorithm 2: Stochastic Proximal Point Method (SPPM)

Input: Stepsizes $\gamma_k > 0$ for $k \geq 0$, starting point $x_0 \in \mathbb{R}^d$

for $k = 0, 1, 2, \ldots$ **do**

  The server: samples $\xi_k \sim \mathcal{D}$;

  The selected client:

    computes $x_{k+1} \in \text{prox}_{\gamma_k f_{\xi_k}}(x_k)$;

    sends x_{k+1} to the server;

For one client, using stationarity criterion, this is equivalent to implicit gradient descent:

$$x_{k+1} = x_k - \gamma_k \nabla f(x_{k+1}).$$

Convergence of SPPM in the strongly-convex regime

Theorem

Let f_ξ and f be μ-strongly convex. Define $\sigma_*^2 := \mathbb{E}_{\xi\sim\mathcal{D}}\left[\|\nabla f_\xi(x^*)\|^2\right]$, where x_* is the optimum. Let $x_0 \in \mathbb{R}^d$ be an arbitrary starting point. Then for any $k \geq 0$ and any $\gamma_k = \gamma > 0$, the iterates of SPPM satisfy

$$\mathbb{E}\left[\|x_k - x^*\|^2\right] \leq \left(\frac{1}{1+\gamma\mu}\right)^{2k}\left(\|x_0 - x^*\|^2 + \frac{\gamma\sigma_*^2}{\mu^2} + \frac{2}{\mu}\right). \tag{2}$$

- In order to remove dependence of σ_*, variance reduction mechanisms are applied:

$$x_{k+1} \in \operatorname{prox}_{\gamma_k f_{\xi_k}}(x_k - \gamma_k g_k),$$

  where g_k is an auxiliary sequence that "estimates" the gradient.

- If $g_k = \nabla f_{\xi_k}(x_*)$, then the σ_*^2 term in (2) vanishes. Thus, one would like g_k to gradually estimate $\nabla f_{\xi_k}(x_*)$.

Momentum-based Variance Reduction

MVR is a momentum-based variance reduction method for solving (Stoch.Opt.) that avoids large batch sizes [CO19].

Algorithm 3: MVR

Input: Stepsizes γ_k, momentum weight parameter p_k, starting point $x_0 \in \mathbb{R}^d$

for $k = 0, 1, 2, \ldots$ **do**

The server: samples $\xi_k \sim \mathcal{D}$;

sends x_k and g_k to the client ξ_k;

The selected client:

$x_{k+1} = x_k - \gamma_k g_k$;

$g_{k+1} = \nabla f_{\xi_k}(x_k) + (1 - p_k)(g_k - \nabla f_{\xi_k}(x_{k-1}))$;

sends x_{k+1} and g_{k+1} to the server;

- g_k is the momentum-based gradient estimate.
- Faster convergence than standard SGD.
- Adaptive variance reduction without large batch sizes.
- No extra memory overhead.

4. SPAM = SPPM + MVR

Pseudocode: SPAM

Algorithm 4: SPAM

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**

The server: ;

Samples $\xi_k \sim \mathcal{D}$;

Sends x_k, g_{k-1} to the client ξ_k;

The selected client: ;

$g_k = \nabla f_{\xi_k}(x_k) + (1 - p_k)(g_{k-1} - \nabla f_{\xi_k}(x_{k-1}))$ (MVR);

$x_{k+1} \in \operatorname{prox}_{\gamma_k f_{\xi_k}}\left(x_k + \gamma_k(\nabla f_{\xi_k}(x_k) - g_k)\right)$ (SPPM);

Sends x_{k+1}, g_k to the server;

- Notice that:

$$x_{k+1} = \arg\min_{y \in \mathbb{R}^d} \left\{ \gamma_k f_{\xi_k}(y) + \gamma_k \langle g_k - \nabla f_{\xi_k}(x_k), y - x_k \rangle + \frac{\|y - x_k\|^2}{2} \right\}.$$

Convergence result

Theorem (Fixed parameters)

Let, $\tilde{x}_{K+1} \sim \mathsf{Unif}(\{x_1, x_2, \ldots, x_{K+1}\})$ and $F := f(x_0) - f_{\inf}$. Then, for fixed stepsizes $\gamma_k = \gamma$ s.t. $\gamma \leq O(1/\delta)$ and momentum parameters $p_k = p$, we have

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

- More details on the conditions of parameters p, γ can be found in the paper.
- The last term does not depend on K. To remove the constant term, one can use decreasing stepsizes. See Theorem 2 of the paper.

Communication Complexity

Corollary

Choose

- $\gamma_k = \gamma = \min\left(\frac{1}{\delta}, \left(\frac{F}{2\delta^2\sigma^2 K}\right)^{1/3}\right)$,
- $p_k = p = \max(\gamma^2\delta^2, 1/K)$.

Then, the communication complexity of SPAM, to obtain

$$\mathbb{E}\left[\|\nabla f(\tilde{x}_{K+1})\|^2\right] \leq \varepsilon \quad \text{error is of order} \quad \mathcal{O}\left(\frac{\delta F + \sigma^2}{\varepsilon} + \frac{\delta\sigma F}{\varepsilon^{3/2}}\right).$$

- That is we can initialize $g_0 = \nabla f(x_0)$. In that case, we obtain the communication complexity of $\mathcal{O}\left(\frac{\delta F}{\varepsilon} + \frac{\delta\sigma F}{\varepsilon^{3/2}}\right)$.
- Suppose all the clients are the same. Then, $\delta = 0$ and each local gradient is the global one. Thus, we do not need to communicate.

Comparison of the settings

Algorithm	Paper	Hessian Sim.	PP	No Smoothness	Cross-Device	Server	Local
FedProx	[LSZ+20]	✗	✔	✔	✔	–	PPM
SABER	[MLFV23]	✔	✗	✔	✗	PAGE	PPM
MIME	[KJK+20]	✔	✗	✗	✔	MVR	SGD
CE-LSGD	[PWW+22]	✔	✔	✗	✔	MVR	SARAH
SPAM	This work	✔	✔	✔	✔	MVR	PPM

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

- Here PP means partial participation. That is when we sample several clients at each iteration. We will not discuss it here. See Section 5 of the paper.
- All the other methods for the Cross-device setting, require SGD-type methods for local computation. This requires smoothness of the objective.

Approximate proximal operator

- Computing the exact proximal operator in SPAM at every iteration is costly.

Definition (ϵ-$\mathrm{prox}_\epsilon(\cdot)$)

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 $\epsilon\text{-}\mathrm{prox}_\epsilon(x_k, g_k, \gamma_k, k)$ is the set of vectors y_{ap}, which satisfy

- decrease in function value: $\mathbb{E}[\phi_k(y_{\mathsf{ap}})] \leq \phi_k(x_k)$,
- approximate stationarity: $\mathbb{E}\left[\left\|\nabla \phi_k(y_{\mathsf{ap}})\right\|^2\right] \leq \epsilon^2$.

where ϕ_k is defined as

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

Pseudocode: SPAM–inexact

Algorithm 5: 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**

The server: ;

Samples $\xi_k \sim \mathcal{D}$;

Sends x_k, g_{k-1} to the client ξ_k;

The selected client: ;

$g_k = \nabla f_{\xi_k}(x_k) + (1 - p_k)(g_{k-1} - \nabla f_{\xi_k}(x_{k-1}))$ (MVR);

$x_{k+1} \in \text{a-prox}_{\epsilon}(x_k, g_k, \gamma_k, \xi_k)$ (a–prox);

The selected client: Sends x_{k+1}, g_k to the server;

Convergence of SPAM-inexact

Theorem (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 $\gamma_k^2 \leq \frac{1}{16\delta^2}$ and choose $p_k = \frac{96\delta^2\gamma_k^2}{96\delta^2\gamma_k^2+1}$. Then,

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

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

- We observe that the term with σ^2 depends on K.
- The approximation error ϵ of the inexact prox appears in the result.

5. Experiments and Conclusion

Plot: SPAM-inexact **vs** CE-LGD

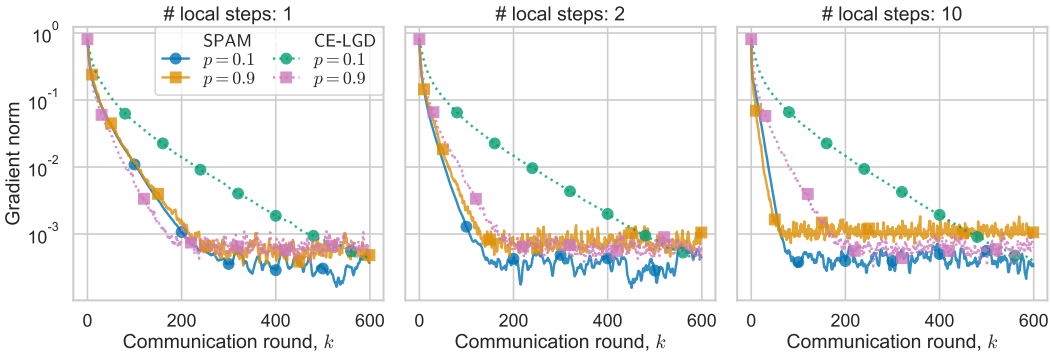


Figure: Comparison of `SPAM-inexact` ($\gamma = 5/\delta$) and `CE-LGD` with different p and number of local steps for distributed ridge regression.

Plot: SPAM–inexact

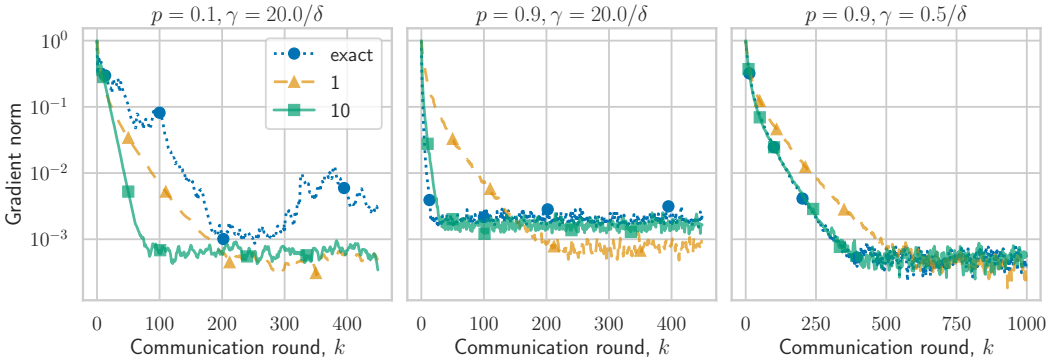


Figure: Convergence of `SPAM–inexact` with different p and γ for distributed ridge regression.

Conclusion

- `SPAM` is an algorithm for cross-device federated learning, which combines `SPPM` and `MVR`.
- Assuming second-order heterogeneity and bounded variance conditions, `SPAM` does not need smoothness of the objective.
- In its most general form, `SPAM` achieves faster communication complexity than its competitors.
- Furthermore, it does not prescribe a specific local method for analysis, providing practitioners with flexibility and responsibility in selecting suitable local solver.

Future work

- Assessing empirical performance on a real cross-device setting.
- Add local stochastic gradients.
- Design adaptive stepsize schedules that do not depend on δ and σ^2. This, however remains open also for `MVR`.

References

[AASC19] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. arXiv preprint arXiv:1912.00818, 2019.

[BCM+18] Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. International journal of medical informatics, 112:59–67, 2018.

[CKLT18] Rachel Cummings, Sara Krehbiel, Kevin A Lai, and Uthaipon Tantipongpipat. Differential privacy for growing databases. Advances in Neural Information Processing Systems, 31, 2018.

[CO19] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. Advances in Neural Information Processing Systems, 32, 2019.

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, pages 1054–1067, 2014.

[KJ22] Ahmed Khaled and Chi Jin. Faster federated optimization under second-order similarity. In The Eleventh International Conference on Learning Representations, 2022.

[KJK+20] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Mime: Mimicking centralized stochastic algorithms in federated learning. arXiv preprint arXiv:2008.03606, 2020.

[KJK+21] 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.

[KMA+21] 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, Zaid 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.

[LSTS20] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. IEEE signal processing magazine, 37(3):50–60, 2020.

[LSZ+20] 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.

[MLFV23] Konstantin Mishchenko, Rui Li, Hongxiang Fan, and Stylianos Venieris. Federated learning under second-order data heterogeneity. Openreview, https://openreview.net/forum?id=jkhVrIllKg, 2023.

[MMSR22] 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.

[MSR22] Artavazd Maranjyan, Mher Safaryan, and Peter Richtárik. Gradskip: Communication-accelerated local gradient methods with better computational complexity. arXiv preprint arXiv:2210.16402, 2022.

[Pic19] Sundar Pichai. Privacy should not be a luxury good. The New York Times, 8:25, 2019.

[PWW+22] 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.

[RSF21] Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. EF21: A new, simpler, theoretically better, and practically faster error feedback. Advances in Neural Information Processing Systems, 34:4384–4396, 2021.

[SHR21] Mher Safaryan, Filip Hanzely, and Peter Richtárik. Smoothness matrices beat smoothness constants: Better communication compression techniques for distributed optimization. Advances in Neural Information Processing Systems, 34:25688–25702, 2021.

[SMAT22] Andrew Silva, Katherine Metcalf, Nicholas Apostoloff, and Barry-John Theobald. Fedembed: Personalized private federated learning. arXiv preprint arXiv:2202.09472, 2022.

[SSR22] 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.

[XIZ+23] Jihao Xin, Ivan Ilin, Shunkang Zhang, Marco Canini, and Peter Richtárik. Kimad: Adaptive Gradient Compression with Bandwidth Awareness. In Proceedings of DistributedML'23, Dec 2023.

[ZLL+18] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. arXiv preprint arXiv:1806.00582, 2018.

This is the last slide.
Thank you!