# Sign SGD with Heavy-Tailed Noise and Differential Privacy

Alexey Kravatskiy
kravtskii.aiu@phystech.edu

Anton Plusnin
plusnin.aa@phystech.edu

Savelii Chezhegov
chezhegov.sa@phystech.edu

Alexander Beznosikov
beznosikov.an@phystech.edu

April 17, 2025

**Abstract**

In the era of large models, federated learning has become indispensable like never before. Sound modern federated learning must meet three key requirements. First, the method must process correctly real-world data, which, in case of Large Language Models, means the algorithm must tolerate noise with unbounded variance. Second, to ensure applicability, the algorithm must converge under these conditions with high probability, not only in expectation. Third, the whole procedure must not jeopardize user data. To address these natural requirements, we have constructed a novel modification of Sign version of Stochastic Gradient Descent. In this paper, we demonstrate that it meets all three earlier stated requisites. We start with proving algorithm's high-probability convergence on data with heavy-tailed noise. Then, we prove its differential privacy. Finally, we show the superior performance of the algorithm in training Large Language Models.

**Keywords:** Sign SGD, differential privacy, high-probability convergence, federated learning, heavy-tailed noise.

**Highlights:**

1. Sign Stochastic Gradient Descent can be used to train LLMs on real data.

2. Our modification of Sign Stochastic Gradient Descent keeps user data private.

3. Our modification of Sign Stochastic Gradient Descent does not require tuning.

# 1 Introduction

Federated Learning is a useful method to train models that require large amounts of data. Indeed, it is often the case that the data is distributed across multiple devices, like mobile phones [9], and it is not only costly to collect all the data in one place, but often also unacceptable due to the requirements of privacy. On the other hand, training the model only on the local data for a particular user is impossible, as the model is large. Hence, a need for a joint training procedure arises. We come to a setting with a server and a number of workers, where each worker has a local dataset. The goal is to train a model on the data from all the workers, without sharing the data between the workers or with the server.

The most obvious way to train a model in a federated setting is to use Stochastic Gradient Descent (SGD) [15] by passing the gradient to the server (add citation). However, when trasmitting the gradient itself, the communication cost is unaffordably high and the privacy of the data is compromised. To address this issue, one can use the Sign Stochastic Gradient Descent algorithm [1]. This algorithm transmits only the signs of the coordinates of the gradients, which is much cheaper in terms of communication.

Clipping the norm of gradient estimate before SGD step, which comprises the method called ClipSGD, is an another great idea that demonstrates positive empirical results [13, 4]. Nonetheless, the clipping requires meticulous tuning of the clipping threshold, which was shown to depend on not only iteration number but also the objective function and the noise [16, Theorem 3.1]. In case of federated learning, the search of the threshold is even more complicated, as the data is distributed across the workers and cannot be used for tuning, and the objective function is more complex due to sophistication and size of the model.

The natural simplification of the clipping is normalization of the gradient, which lies at the core of NSGD [5]. NSGD outperformed ClipSGD on the task of sequence labeling via LSTM Language Models [10]. Despite this fact, NSGD, unlike ClipSGD, requires larged batch sizes for convergence, which though can be mended by applying momentum [2]. The major drawback of NSGD is the absence of proofs of convergence with high probability. Moreover, the method seems to be not private.

As to guarantee the privacy is our priority, we opted to base our algorithm on SignSGD.

The convergence with high probability of SignSGD for the heavy-tailed noise was recently proved [8], which makes SignSGD a perfect candidate for federated learning. Simultaneously, SignSGD was already used as a base to create a differentially private algorithm [7]. However, in both cases, no proofs for the convergence of the algorithm in the modern federated learning setting were provided.

In this paper, we propose a modification of SignSGD that can be used to train LLMs on real data. We show that our algorithm converges with high probability, even in the presence of heavy-tailed noise, and does not require tuning. We also show that our algorithm is differentially private. Finally, we test the algorithm on the training of LLMs.

# 2 Problem statement.

To-do: what kind of distribution $S$ is. Change the abstract. It should start with narrow problem. Remove "can" and variants from the paper altogether!

First, we need to state that we work with stochastic optimization.

**Stochastic optimization problem.** The stochastic optimization problem for a smooth non-convex function $f : \mathbb{R}^d \to \mathbb{R}$ is:

$$\min_{x \in \mathbb{R}^d} f(x) := \mathbb{E}_{\xi \sim \mathcal{S}}[f(x, \xi)], \tag{1}$$

where random variable $\xi$ is sampled from an unknown distribution $\mathcal{S}$. The gradient oracle returns unbiased gradient estimate $\nabla f(x, \xi) \in \mathbb{R}^d$. In machine learning, for instance, $f(x, \xi)$ is a loss function on a sample $\xi$ [17].

The most popular algorithm to solve (1) is Stochastic Gradient Descent (SGD) [15]:

$$x^{k+1} = x^k - \gamma_k \cdot g^k, \quad g^k := \nabla f(x^k, \xi^k).$$

For non-convex functions, the algorithm must stop at the point with sufficiently small gradient norm. We will apply the algorithm to the federated optimization problem.

**Federated optimization problem.** Let $I = X \times Y$ be a sample space, where $X$ is a space of feature vectors and $Y$ is a label space. For the hypothesis space $\mathcal{W} \subseteq \mathbb{R}^d$, a loss function is defined as $l : \mathcal{W} \times I \to \mathbb{R}$ which measures the loss of the prediction on the data point

$(x, y) \in I$ based on the hypothesis $w \in \mathcal{W}$. For a dataset $D \subset I$, the global loss function $F : \mathcal{W} \to \mathbb{R}$ is defined as

$$F(w) = \frac{1}{|D|} \sum_{(x,y) \in D} l(w; (x, y)). \tag{2}$$

In case of distributed optimization, the dataset is split between $M$ workers. Each worker $m$ has a local dataset $D_m \subset I$ and a local function $f_m : \mathbb{R}^d \to \mathbb{R}$ defined as

$$f_m(w) = \frac{1}{|D_m|} \sum_{(x_n,y_n) \in D_m} l(w; (x_n, y_n)), \tag{3}$$

where $|D_m|$ is the size of worker $m$'s local dataset $D_m$.

Thus, our goal is to solve the following federated optimization problem:

$$\min_{w \in \mathbb{R}^d} F(w) \quad \text{where} \quad F(w) \stackrel{\text{def}}{=} \frac{1}{M} \sum_{m=1}^{M} f_m(w). \tag{4}$$

We assume that the data are distributed over the workers uniformly, consequently, $\mathbb{E}[f_m(w)] = F(w)$ for workers' data distribution.

Now, let us introduce the requirements for the algorithm to solve the federated optimization problem.

**Heavy-tailed noise.** Noise has bounded $\kappa$-th moment for some $\kappa \in (1, 2]$, i.e. $\mathbb{E}_\xi[\|\nabla f(x, \xi) - \nabla f(x)\|_2^\kappa] \leq \sigma^\kappa$. In particular, the noise can have unbounded variance, i.e. $\kappa < 2$.

**Differential privacy.** Additionally, the algorithm must be private, which means it must satisfy $(\epsilon, \delta)$-local differential privacy [3].

**High-probability convergence.** The algorithm must have convergence guarantees which hold true with probability at least $1 - \delta, \delta \in (0, 1)$.

# 3 Theory

## 3.1 The Algorithm and the compressor

The algorithm we are working with can be defined generally as follows:

**Algorithm 1** Stochastic-Sign SGD with majority vote

**Input**: learning rate $\eta$, current hypothesis vector $w^{(t)}$, $M$ workers each with an independent gradient $\boldsymbol{g}_m^{(t)}$, the 1-bit compressor $q(\cdot)$.
**on server:**
  **pull** $q(\boldsymbol{g}_m^{(t)})$ **from** worker $m$.
  **push** $\tilde{\boldsymbol{g}}^{(t)} = sign\Big(\frac{1}{M}\sum_{m=1}^{M} q(\boldsymbol{g}_m^{(t)})\Big)$ **to** all the workers.
**on each worker:**
  **update** $w^{(t+1)} = w^{(t)} - \eta\tilde{\boldsymbol{g}}^{(t)}$.

As the algorithm must be differentially private, we use as a 1-bit compressor dp-sign compressor [7]:

**Definition 1.** *For any given gradient $\boldsymbol{g}_m^{(t)}$, the compressor dp-sign outputs dp-sign$(\boldsymbol{g}_m^{(t)}, \epsilon, \delta)$. The i-th entry of dp-sign$(\boldsymbol{g}_m^{(t)}, \epsilon, \delta)$ is given by*

$$dp\text{-}sign(\boldsymbol{g}_m^{(t)}, \epsilon, \delta)_i = \begin{cases} 1, & \textit{with probability } \Phi\Big(\frac{(\boldsymbol{g}_m^{(t)})_i}{\sigma}\Big) \\ -1, & \textit{with probability } 1 - \Phi\Big(\frac{(\boldsymbol{g}_m^{(t)})_i}{\sigma}\Big) \end{cases} \tag{5}$$

*where $\Phi(\cdot)$ is the cumulative distribution function of the normalized Gaussian distribution; $\sigma = \frac{\Delta_2}{\epsilon}\sqrt{2\ln(\frac{1.25}{\delta})}$, where $\epsilon$ and $\delta$ are the differential privacy parameters and $\Delta_2$ is the sensitivity measure.*

As stated in Theorem 5 from [7], the *dp-sign* mechanism is $(\epsilon, \delta)$-differentially private for any $\epsilon$ and $\delta \in (0, 1)$.

## 3.2 Convergence for dp-sign

Theorem 6 from [7] should establish probability of the dissimilarity of majority sign of the gradients and majority sign of the dp-signs:

**Theorem 1.** *Let $u_1, u_2, \cdots, u_M$ be $M$ known and fixed real numbers. Further define random variables $\hat{u}_i = dp\text{-}sign(u_i, \epsilon, \delta), \forall 1 \le i \le M$. Then there always exist a constant $\sigma_0$ such that when $\sigma \ge \sigma_0$, $P(sign(\frac{1}{M}\sum_{m=1}^{M}\hat{u}_i) \ne sign(\frac{1}{M}\sum_{m=1}^{M}u_i)) < \Big(1 - x^2\Big)^{\frac{M}{2}}$, where $x = \frac{|\sum_{m=1}^{M}u_m|}{2\sigma M}$.*

However, as we recently found out, the theorem is fundamentally flawed, as it makes $\sigma$ a parameter, while it follow from the definition of the *dp-sign* compressor that $\sigma$ is a function of $\epsilon$ and $\delta$.

Let us suppose that there exist $\epsilon$ and $\delta$ such that for $\sigma(\epsilon, \delta)$, the inequality on probability from the theorem holds true. Then, let us walk through the proof to find a precise lower bound for $\sigma$. If this bound is lower than $\sigma(\epsilon, \delta)$, the theorem is indeed correct.

In the proof of Theorem 6, the authors needed to find the lower bound for the following expression:

$$\frac{1}{\sqrt{2\pi}\sigma}\left[\left|\sum_{m=1}^{M} u_m\right| e^{-\frac{u_1^2}{2\sigma^2}} + \left|\sum_{m=2}^{M} u_m\right|\left[e^{-\frac{(\sum_{m=2}^{M} u_m)^2}{2\sigma^2}} - 1\right]\right]$$

Instead of taking limit $\sigma \to \infty$ as they did, we will use the well known relation $e^{-x^2} \geq 1 - x^2$. After applying it to the earlier mentioned expression and making some trivial transformations, we get the following:

$$\sigma^2 \geq \frac{7}{5}\left(u_1^2 + \left|\sum_{m=1}^{M} u_m\right|^2 + \frac{|u_1|^3}{\left|\sum_{m=1}^{M} u_m\right|}\right).$$

It is a sufficient condition, not a necessary one. However, it reflects the key features of the condition on $\sigma$. $\frac{|u_1|^3}{\left|\sum_{m=1}^{M} u_m\right|}$ is an extremely unreliable term, as $|u1|$ may be high, especially for heavy-tailed noise, while $\left|\sum_{m=1}^{M} u_m\right|$ may be small (it is easy to construct an appropriate example, with 3 workers). Hence, the bound on $\sigma$ is not only high, but also unstable. Consequently, we have no proofs whatsoever of the convergence of the algorithm.

Moreover, our misgivings are supported by the fact that in an updated version [6] of the article [7], the authors have removed all mentions of dp-sign. Right now, we are facing the problem that the algorithm for which we sought proofs for more general type of noise, might not make sense at all. Currently we are focused on improving our understanding of differential privacy, which will help us construct our own

The problems with convergence of dp-sign are not limited by the incorrect Theorem 6. The authors have overlooked another issue of utmost importance. Proving the $(\epsilon, \delta)$ differential privacy of dp-sign, they did so for only one call of dp-sign operator. However, according to [3] (Theorem 3.16), the overall privacy of running dp-sign $T$ times is $(T\epsilon, T\delta)$. Consequently, when we set target $\epsilon \approx 10$, $\delta \approx 10^-5$, as is recommended for machine learning purposes [14], for $T = 100$ we get 100 times lower $\epsilon$ and $\delta$ for dp-sign. Judging by our experiments, this destroys convergence. That is intuitive: high noise for low $(\epsilon, \delta)$ leads to poorer convergence, and this forces us to use still lower $(\epsilon, \delta)$ to preserve the required level of privacy during the run of the algorithm.

### 3.3 Rényi Differntial privacy

**Definition 2** (Rényi divergence [12]). *Let $P$ and $Q$ be two distributions on $\mathcal{X}$ defined over the same probability space, and let $p$ and $q$ be their respective densities. The Rényi divergence of a finite order $\alpha \neq 1$ between $P$ and $Q$ is defined as*

$$\mathrm{D}_\alpha(P \parallel Q) \triangleq \frac{1}{\alpha - 1} \ln \int_{\mathcal{X}} q(x) \left(\frac{p(x)}{q(x)}\right)^\alpha \mathrm{d}x.$$

*Rényi divergence at orders $\alpha = 1, \infty$ are defined by continuity.*

**Definition 3** (Rényi differential privacy (RDP)). *We say that a randomized mechanism $\mathcal{M}\colon \mathcal{S} \to \mathcal{R}$ satisfies $(\alpha, \varepsilon)$-Rényi differential privacy (RDP) if for any two* adjacent *inputs $S, S' \in \mathcal{S}$ it holds that*

$$\mathrm{D}_\alpha(\mathcal{M}(S) \parallel \mathcal{M}(S')) \leq \varepsilon.$$

The notion of adjacency between input datasets is domain-specific and is usually taken to mean that the inputs differ in contributions of a single individual. In this work, we will use this definition and call two datasets $S, S'$ to be adjacent if $S' = S \cup \{x\}$ for some $x$ (or vice versa).

**Definition 4** (Sampled Gaussian Mechanism (SGM)). *Let $f$ be a function mapping subsets of $\mathcal{S}$ to $\mathbb{R}^d$. We define the Sampled Gaussian mechanism (SGM) parameterized with the sampling rate $0 < q \leq 1$ and the noise $\sigma > 0$ as*

$$\mathrm{SG}_{q,\sigma}(S) \triangleq f(\{x\colon x \in S \text{ is sampled with probability } q\}) + \mathcal{N}(0, \sigma^2 \mathbb{I}^d),$$

*where each element of $S$ is sampled independently at random with probability $q$ without replacement, and $\mathcal{N}(0, \sigma^2 \mathbb{I}^d)$ is spherical $d$-dimensional Gaussian noise with per-coordinate variance $\sigma^2$.*

From [12], we get the guarantee for integer $\alpha$:

$$\varepsilon_R \leq \frac{1}{\alpha - 1} \log \left( \sum_{k=0}^{\alpha} \binom{\alpha}{k} (1-q)^{\alpha-k} q^k \exp\left(\frac{k^2 - k}{2\sigma^2}\right)\right)$$

From [11], we relate $\varepsilon_R$ with $\varepsilon, \delta$ for the whole process with $T$ iterations. If this relation is satisfied, the algorithm is $(\varepsilon, \delta)$-private:

$$\varepsilon_R \le \varepsilon/T - \frac{\log 1/\delta}{T(\alpha-1)}$$

.

This is all theory we need to conduct experiments.

# 4 Experiments

In this section, we present the experimental results for the methods we discussed in Sections 2 and 3. First, we applied the algorithms to a classic machine learning problem. Then, we tested the algorithms on the training of Large Language Models.

## 4.1 Synthetic noise.

We test our algorithm on the method of logistic regression for UCI Mushroom Dataset. The dataset consists of 6,449 training samples and 1,625 testing samples. Each sample has $d = 112$ features, and represents a mushroom either poisonous or edible. We apply different algorithms to train the logistic regression model. The algorithms are run on 20 workers, with training data distributed between the workers equally. We compare the cases when there is no noise, when the noise is normal with $\sigma^2 = 1/4$, and when the noise is Levy stable with standard deviation $= 1/4$. We set the learning rate $\gamma = 1/\sqrt{dT}$ for SGD, sign-SGD, and dp-signSGD, and there are 2000 iterations for each algorithm. The results are presented in fig. 1.

As we see, dp-signSGD does converge, even under heavy-tailed noise.

## 4.2 DP-Sign

Having verified Sign-SGD, we test dp-sign on the same dataset. We present three variants of the dp-sign:

1. DP-SignSGD from [7] with unrectified $(\epsilon, \delta)$. The algorithm converges, but is not sufficiently private, as we highlighted earlier.

2. DP-SignSGD from [7] with rectified $(\epsilon/T, \delta/T)$, where $T$ is 1000. The algorithm is now private, but does not converge.

3. Our modification of DP-SignSGD. We used Rényi differential privacy [3]. This type of privacy can be easily converted to classical differential privacy, but has a more temperate

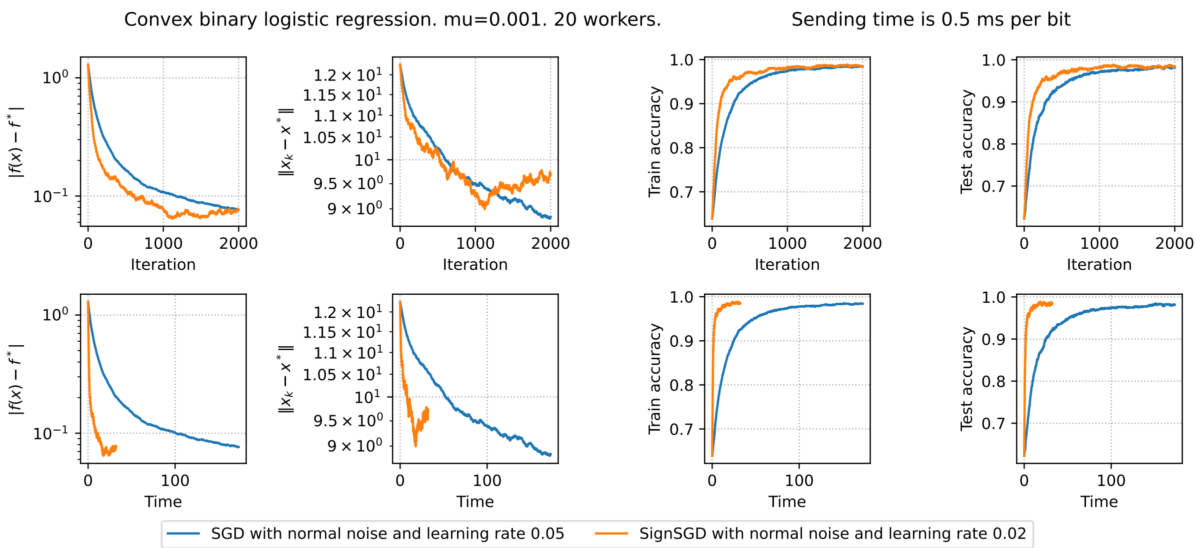


Figure 1: Logistic regression on UCI Mushroom Dataset.
Performance of SGD and SignSGD with majority voting.

Table 1: DP-Sign SGD methods

| Method | Citation | Is $(\epsilon, \delta)$ private | Converges |
|---|---|---|---|
| ChineseDP-SignSGD-cheat | [7] | No | Yes |
| ChineseDP-SignSGD rectified | [7] | Yes | No |
| Rényi DP-SignSGD | **This paper** | Yes | Yes |

deterioration of privacy during the sequential composition: the noise $\sigma \sim \sqrt{T}$ instead of $\sigma \sim T$. We will expand Theory and introduce there this algorithm, when it will be ready. For now, it as dp-sign does not converge.

Currently, we have the following table:

# 5 Conclusion

We have highlighted the deficiencies of the previous approach of working with dp-sign-SGD. With a use of Rényi differential privacy and advanced numerically-produced bounds for Sampled Gaussian Mechanism, we have consturcted a privacy accountant that makes dp-sign-SGD converge and private, which, to the best of our knowledge, has never been done before.

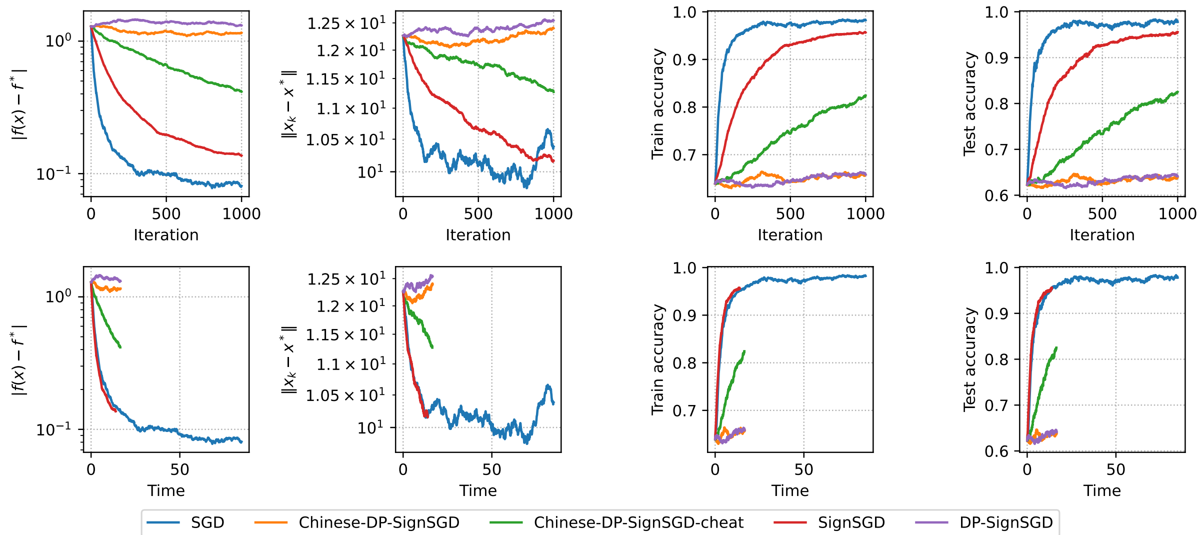


Figure 2: Logistic regression on UCI Mushroom Dataset.
Performance of SGD, SignSGD and different variants of DP-SignSGD with majority voting.

# 6 Acknowledgments

We will fill this section when we are done with the article.

# A Additional Proofs

We will present them after we complete the tests and fix the algorithm.

# References


[1] Jeremy Bernstein et al. "signSGD: Compressed Optimisation for Non-Convex Problems". In: *International Conference on Machine Learning*. 2018, pp. 560–569.

[2] Ashok Cutkosky and Harsh Mehta. "Momentum improves normalized sgd". In: *International conference on machine learning*. PMLR. 2020, pp. 2260–2268.

[3] C. Dwork, A. Roth, et al. "The algorithmic foundations of differential privacy". In: *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[5] Elad Hazan, Kfir Levy, and Shai Shalev-Shwartz. “Beyond convexity: Stochastic quasi-convex optimization”. In: *Advances in Neural Information Processing Systems*. 2015, pp. 1594–1602.

[6] Richeng Jin et al. “Sign-Based Gradient Descent With Heterogeneous Data: Convergence and Byzantine Resilience”. In: *IEEE Transactions on Neural Networks and Learning Systems* 36.2 (2025), pp. 3834–3846. DOI: 10.1109/TNNLS.2023.3345367.

[7] Richeng Jin et al. “Stochastic-Sign SGD for Federated Learning with Theoretical Guarantees”. In: *Part of this work is published in IEEE Transactions on Neural Networks and Learning Systems, 2024* 36.2 (Feb. 25, 2020), pp. 3834–3846. ISSN: 2162-2388. DOI: 10.1109/tnnls.2023.3345367. arXiv: 2002.10940 [cs.LG].

[8] Nikita Kornilov et al. *Sign Operator for Coping with Heavy-Tailed Noise: High Probability Convergence Bounds with Extensions to Distributed Optimization and Comparison Oracle*. 2025. DOI: 10.48550/ARXIV.2502.07923.

[9] Brendan McMahan et al. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. In: *Artificial Intelligence and Statistics*. 2017, pp. 1273–1282.

[10] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. “Regularizing and optimizing LSTM language models”. In: *arXiv preprint arXiv:1708.02182* (2017).

[11] Ilya Mironov. “Rényi Differential Privacy”. In: *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, Aug. 2017, pp. 263–275. DOI: 10.1109/csf.2017.11. URL: http://dx.doi.org/10.1109/CSF.2017.11.

[12] Ilya Mironov, Kunal Talwar, and Li Zhang. *Rényi Differential Privacy of the Sampled Gaussian Mechanism*. 2019. arXiv: 1908.10530 [cs.LG]. URL: https://arxiv.org/abs/1908.10530.

[13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. “On the difficulty of training recurrent neural networks”. In: *International conference on machine learning*. 2013, pp. 1310–1318.

[14] Natalia Ponomareva et al. “How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy”. In: *Journal of Artificial Intelligence Research* 77 (July 2023). arXiv:2303.00654 [cs], pp. 1113–1201. ISSN: 1076-9757. DOI: 10.1613/jair.1.14649. URL: http://arxiv.org/abs/2303.00654 (visited on 04/09/2025).

[15] Herbert Robbins and Sutton Monro. "A stochastic approximation method". In: *The annals of mathematical statistics* (1951), pp. 400–407.

[16] Abdurakhmon Sadiev et al. "High-Probability Bounds for Stochastic Optimization and Variational Inequalities: the Case of Unbounded Variance". In: *arXiv preprint arXiv:2302.00999* (2023).

[17] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.