Sign-SGD via Parameter-free optimization

Daniil Medyakov[1 *], Sergey Stanko[1], Gleb Molodtsov[1], Philip Zmushko[1,2], Grigoriy Evseev[1], Egor Petrov[1,2], Aleksandr Beznosikov[1,3]

*Corresponding author: medyakovd3@gmail.com [1]BRAIn Lab [2]Yandex Research [3]Innopolis University

Problem setup and why parameter-free SIGN-SGD matters

Optimization problem. We study classic unconstrained minimization problem:

$$\min_{x\in\mathbb{R}^d} f(x),$$

and classic SIGN-SGD method:

$$x^{t+1} = x^t - \gamma\,\mathrm{sign}(\nabla f(x^t)).$$

The main goal of this work is to develop a **parameter-free** method based on the SIGN-SGD optimizer.

Why parameter-free?

- Theoretically optimal stepsizes **depend on global problem constants**, such as the L-smoothness of the objective f, the M-Lipschitzness of f, and the initial distance to the solution.
- In practice, however, these constants are typically **impossible to compute**.
- As a result, deploying an optimizer often requires **manual stepsize tuning** for **each individual task**.
- **Parameter-free** approaches address this issue by selecting the step size adaptively at each iteration, using computable approximations of the global problem constants along the optimization trajectory.
- This adaptive strategy **substantially reduces** end-to-end training time. *For example*, if one evaluates 5-10 candidate step sizes, each over a horizon of 10% of the total training budget, a parameter-free method can provide an overall $1.5\times$ **to** $2\times$ **speedup in total training time**. For large-scale model training, this constitutes a significant practical improvement.

Why SIGN-SGD?

- SIGN-SGD is a **memory-efficient** optimizer, as it only requires storing the sign of the gradient of the objective f.
- SIGN-SGD can be used effectively for **training large-scale models**.
- SIGN-SGD exhibits **step geometry** similar to that of ADAM.
- SIGN-SGD is applicable for **distributed optimization**, as it corresponds to 1-bit gradient compression.

SIGN-SGD parameter-free objectives. For smooth optimization, the standard analysis for the deterministic variant of SIGN-SGD yields the following bound:

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(x^t)\|_1 \leqslant \frac{\Delta^*}{\gamma T} + \frac{\gamma L_\infty}{2}, \tag{1}$$

where $\Delta^* = f(x^0) - f(x^*)$ — initial distance to the solution, L_∞ — smoothness constant of the problem with respect to l_∞-norm. A special choice of the stepsize in (1) yields the optimal convergence bound for the method. Specifically,

$$\text{choosing}\quad \gamma = \frac{\sqrt{\Delta^*}}{\sqrt{L_\infty T}},\quad \text{we obtain}\quad \mathcal{O}\left(\frac{\sqrt{\Delta^* L_\infty}}{\sqrt{T}}\right)\quad \text{convergence rate.}$$

Thus, designing a parameter-free optimizer based on SIGN-SGD amounts to achieving adaptivity to the constants L_∞, and Δ^*.

Convergence properties of classic SIGN-SGD:

- SIGN-SGD does not achieve regret convergence with a constant step size, even for linear objectives.
- SIGN-SGD does not converge for non-smooth objectives, even with an adaptive stepsize.
- SIGN-SGD with a batch size of 1 and stochastic gradients does not converge.

Full Article

Find BRAIn Lab on X

Contributions

- **Parameter-free SIGN-SGD.** We introduce a parameter-free deterministic SIGN-SGD method. The core idea involves per-iteration step-size adaptation. Every iteration, we choose estimators of L_∞ and $f(x^0) - f(x^*)$ using the current gradient information. This design is practical, as it requires no additional hyperparameter search or restarts.
- **Stochastic and distributed settings.** We study our algorithm in the distributed setting and the case of stochastic gradient oracles. A lack of stochastic analysis presents a significant drawback in parameter-free optimization. Our work addresses this limitation.
- **Practical extensions.** We extend our approach in two important directions:
  - We incorporate momentum to improve practical performance.
  - We provide a memory-efficient parameter-free version. It stores only the sign of the gradient from the previous step while maintaining adaptivity to the problem properties.
- **Empirical validation.** We demonstrate that our methods are competitive in practical tasks, including LLM and ViT training. A momentum variant further improves performance across both language and vision benchmarks. Empirically, parameter-free training matches or is slightly below tuned SIGN-SGD and ADAMW with cosine schedules, while achieving appreciably better overall training time.

Algorithm design and theoretical analysis

We impose the following assumptions on the objective function f.

Assumption 1: Smoothness.

The function f is L_∞-smooth, i.e., it satisfies $\|\nabla f(x) - \nabla f(y)\|_1 \leqslant L_\infty\|x-y\|_\infty$ for any $x, y \in \mathbb{R}^d$.

We consider a smoothness constant with respect to the l_∞-norm to highlight the theoretical improvement in the convergence of SIGN-SGD relative to SGD. Under this assumption, the criterion is given by the l_1-norm of the gradient, which is strictly larger than the l_2-norm used in the analysis of SGD. At the same time, the L_∞ constant in over-parameterized neural networks is often practically comparable to the L_2 smoothness constant, since most of the spectrum is concentrated near zero.

Assumption 2: Convexity.

The function f is convex, i.e., it satisfies $f(x) \leqslant f(y) + \langle\nabla f(x), x - y\rangle$ for any $x, y \in \mathbb{R}^d$.

Although neural networks are inherently non-convex, theoretical analysis under convexity assumptions remains relevant. Recent studies suggest that deep neural networks often exhibit properties similar to convexity in certain regions, making insights from convex analysis applicable. Moreover, convex optimization serves as a theoretical foundation for the design of optimization algorithms. For example, ADAGRAD algorithm was initially developed and analyzed for convex problems.

Assumption 3: Existence of a finite minimum value

The function f has a (maybe not unique) finite minimum, i.e., $f(x^*) = \inf_{x\in\mathbb{R}^d} f(x) > -\infty$.

PF-Sign idea: per-iteration approximation of $\Delta^\star, L_\infty$

Approximation of $\Delta^\star$. We maintains a non-decreasing estimator $d^t \leqslant D^t$ via the observed progress:

$$\widetilde{d}^t = \sum_{i=0}^{t-1}\gamma^i\,\langle\nabla f(x^{i+1}), \mathrm{sign}(\nabla f(x^i))\rangle, \qquad d^t = \max\{d^{t-1}, \widetilde{d}^t\}.$$

When a lower bound $\widetilde{f} \leqslant f(x^\star)$ is known, we can also use the simpler surrogate

$$f(x^0) - \widetilde{f}.$$

Approximation of L_∞. The local smoothness estimator accumulates coordinate-wise gradient variation:

$$\eta^t = \eta^{t-1} + \frac{\|\nabla f(x^t) - \nabla f(x^{t-1})\|_1}{\|x^t - x^{t-1}\|_\infty}, \qquad \lambda^t = \frac{1}{\sqrt{\eta^t}}.$$

Then the effective stepsize becomes either

$$\gamma^t = \lambda^t\sqrt{d^t} \qquad \text{or} \qquad \gamma^t = \lambda^t\sqrt{f(x^0) - \widetilde{f}}.$$

Algorithm ALIAS

Algorithm 1: ALIAS

Input: Starting point $x^0 \in \mathbb{R}^d$, initial L_∞-approximation $\eta^{-1} = 0$, initial Δ^*-approximation $d^0 \in \mathbb{R}_+$, lower bound $\widetilde{f}$ on $f(x^*)$, number of iterations T
for $t = 0, \dots, T-1$ **do**
  Compute gradient $\nabla f(x^t)$
  $\eta^t = \eta^{t-1} + \frac{\|\nabla f(x^t) - \nabla f(x^{t-1})\|_1}{\|x^t - x^{t-1}\|_\infty}$; $\lambda^t = \frac{1}{\sqrt{\eta^t}}$
  if $t \neq 0$ **then**
    $\widetilde{d}^t = \sum_{i=0}^{t-1}\gamma^i\langle\nabla f(x^{i+1}), \mathrm{sign}(\nabla f(x^i))\rangle$
    $d^t = \max\left(d^{t-1}, \widetilde{d}^t\right)$
  end if
  Option I: $\gamma^t = \lambda^t\sqrt{d^t}$
  Option II: $\gamma^t = \lambda^t\sqrt{f(x^0) - \widetilde{f}}$
  $x^{t+1} = x^t - \gamma^t\mathrm{sign}(\nabla f(x^t))$
end for

- **No restarts, no grid search:** All parameters are updated online.
- **Sign-based geometry:** ALIAS adapts to the SIGN-SGD parameters D^t and L_∞, yielding stepsize for sign-based update.
- **Extensions:** The same idea transfers to compressed multi-node and stochastic settings.

Deterministic and stochastic theorems

Theorem 1: Deterministic case

Assume L_∞-smoothness, convexity, finite minimum, and access to exact gradients. Let

$$\varepsilon \geqslant \frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(x^t)\|_1, \qquad L_\infty^0 = \frac{\|\nabla f(x^1) - \nabla f(x^0)\|_1}{\|x^1 - x^0\|_\infty}.$$

Then ALIAS reaches ε-accuracy in

$$\mathcal{O}\left(\frac{(\Delta^*)^2(L_\infty)^3}{d^0(L_\infty^0)^2\,\varepsilon^2}\right)$$

iterations with Option I and in

$$\mathcal{O}\left(\frac{\Delta^*(L_\infty)^3}{(L_\infty^0)^2\,\varepsilon^2}\right)$$

iterations with Option II.

Theorem 2: Stochastic case

Assume stochastic gradients with coordinate-wise bounded variance, increasing batch-size and sample-wise L_∞-smoothness. Then the stochastic ALIAS variant with Option II reaches near-stationarity with complexity of the form

$$\mathcal{O}\left(\frac{\Delta^*}{\varepsilon^2}\left[(L_\infty)^3\mathbb{E}\left(\frac{1}{L_\infty^{0,\xi^1}}\right)^2 + \|\sigma\|_1^2 L_\infty\,\mathbb{E}\frac{1}{\min\limits_{0\leqslant t\leqslant T-1} L_\infty^{t,\xi^{t+1}}}\right]\right).$$

Discussion of bounds

- We provide a guarantee in terms of **finding a stationary point** even under convexity assumptions on the objective, since SIGN-SGD does not admit regret convergence.
- In the exact-gradient theorem, ALIAS pays an **extra factor** $(\frac{L_\infty}{L_\infty^0})^2$. This additional factor arises from the inexact approximation of the constant L_∞, but it **does not affect the asymptotic convergence rate**.
- If L_∞ is known, replacing λ^t by $\frac{1}{\sqrt{L_\infty + \sum_i L_\infty^i}}$ **improves the rate** to roughly $\mathcal{O}(\frac{\Delta^* L_\infty}{\varepsilon^2})$.
- In the stochastic case, convergence is to a **variance-dependent neighborhood**; **only increasing mini-batching** restores the standard variance control.
- To construct **memory-efficient version**, we swaps the (ℓ_∞, ℓ_1)-norms in smoothness approximation and works with L_1-smoothness.

Experiments

LLaMA pre-training. We pre-train `LLaMA`-style models on `C4` dataset and compare tuned SIGN-SGD, ADAMW, DOG, D-ADAPTATION, MOMO, PRODIGY baselines, and ALIAS with/without added momentum. The main take-away is that **basic ALIAS is close to tuned cosine-scheduled Sign-SGD**, while **ALIAS with added momentum surpasses tuned baselines** without learning-rate search.

Table 1: Comparion of methods on LLAMA (130M) pre-training.

Algorithm	Validation loss ↓	Perplexity ↓
SIGN-SGD (wd, lr)	3.041	20.923
SIGN-SGD (wd, lr, cosine sc)	**2.980**	**19.693**
STEEPEST DESCENT (wd, lr, cosine sc)	3.022	20.537
NORMALIZED SGD (wd, lr, cosine sc)	*3.006*	*20.169*
ALIAS (wd)	*3.006*	*20.169*
SIGN-SGD (wd, β, lr)	2.968	19.459
SIGN-SGD (wd, β, lr, cosine sc)	*2.923*	*18.596*
STEEPEST DESC. (wd, β, lr, cosine sc)	2.932	18.765
NORM. SGD (wd, β, lr, cosine sc)	2.934	18.803
ADAMW (wd, β, lr, cosine sc)	2.929	18.698
DOG (wd, β, cosine sc)	2.939	18.897
D-ADAPTATION (wd, β, cosine sc)	2.927	18.672
MOMO (wd, β, cosine sc)	2.925	18.634
PRODIGY (wd, β)	3.003	20.145
PRODIGY (wd, β, cosine sc)	2.930	18.727
ALIAS (wd, β)	2.976	19.609
ALIAS (wd, β, cosine sc)	**2.918**	**18.504**

Cosine behavior emerges automatically. A striking qualitative result is that the effective ALIAS with momentum **stepsize closely matches the tuned cosine schedule**.

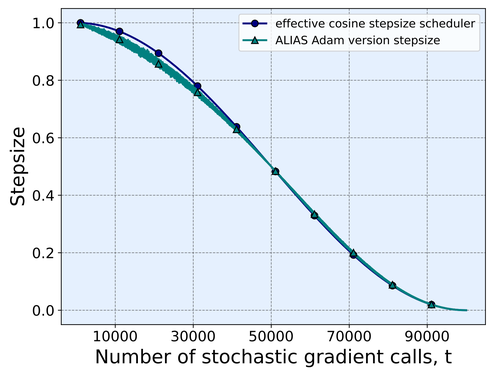

Figure 1: The ALIAS with momentum effective stepsize naturally tracks the cosine scheduler.

AlgoPerf benchmark. We evaluate parameter-free methods on AlgoPerf benchmark. We choose MRI reconstruction (MRI) and molecular property prediction (MPP) tasks. The main highlight is that **ALIAS with added momentum achieves better metrics** than all competing parameter-free methods, and on the MRI reconstruction task it even **outperforms a tuned ADAMW**.

Table 2: Comparison of methods on AlgoPerf benchmark.

Algorithm	MRI, SSIM ↑	MPP, mAP ↑
AdamW (wd, β, lr, cosine sc)	*0.723*	**0.254**
DOG (wd, β)	0.714	0.231
D-Adaptation (wd, β)	0.722	0.221
MoMo (wd, β)	*0.723*	0.221
Prodigy (wd, β)	*0.723*	0.212
ALIAS (wd, β)	**0.724**	*0.242*

Experimental conclusion

- ALIAS **removes the learning-rate grid search** while **remaining competitive** with tuned Sign-SGD.
- ALIAS with added momentum is **the strongest practical variant**.
- ALIAS with added momentum **automatically reproduces the behavior of a cosine scheduler**.
- On AlgoPerf, the method is **competitive with strong parameter-free baselines** and even **outperforms tuned ADAMW** on MRI reconstruction task.
- Proposed ALIAS algorithm is a strong general-purpose method that does not require stepsize tuning, allowing it to **accelerate the end-to-end training pipeline by** $1.5-2\times$ **without sacrificing performance**.