# Muon Does Not Converge on Convex Lipschitz Functions

**Tetiana Parshakova**[*] **Ahmed Khaled**[†] **Michael Crawshaw**[*]
**Guillaume Garrigos**[‡] **Robert M. Gower**[*]

[*]CCM, Flatiron Institute, Simons Foundation [†]Google DeepMind [‡]Université Paris Cité

## Abstract

Muon and its variants have shown strong empirical performance in a variety of deep learning tasks. Existing convergence analyses of Muon rely on smoothness assumptions, though arguably the most successful function class for developing deep learning methods (such as AdaGrad, Shampoo, Schedule-Free and more) has been the class of convex and Lipschitz functions. In this paper we question whether the classical convex Lipschitz model is a useful one for understanding Muon.

Our answer is no. We show that Muon does not converge on the class of convex and Lipschitz functions, regardless of the choice of learning rate schedule. We also show that error feedback restores convergence of regularized Muon and all the non-Euclidean subgradient methods with momentum. However, error feedback does not consistently improve performance in two representative settings for image classification (CIFAR-10) and language modeling (nanoGPT on FineWeb-Edu 10B). Our conclusion is that convex Lipschitz theory, despite having a prominent role in the design of practical methods for deep learning, is not the most suited one for Muon. This suggests that Muon's success must come from structure absent from this model, most plausibly related to smoothness.

## 1 Introduction

The Muon optimizer [Jor+24b; BN24], together with related variants [LLS25; Pet+25], has recently emerged as a highly competitive optimization approach in a growing range of applications. These include training nanoGPT models at the 124M scale [Jor+24a], the nanochat challenge at 1.5B parameters [Kar26], and diffusion models [Sch25]. Muon has also been used in large-scale language model training, for example in the Moonlight 16B (3B active) MoE model [Liu+25], the 1T (32B active) parameters Kimi frontier model [Tea+25] and very recently the DeepSeek-V4 model [Dee26] with 1.6T (49B active) parameters. These empirical results have generated substantial interest in understanding the theoretical properties of Muon and related spectral methods.

Muon combines spectral descent [CCC15; Car+15] with momentum and a GPU-efficient approximation of the polar factor [Ams+26; Jor+24b]. Existing convergence analyses of Muon and related methods predominantly assume some form of smoothness; see Section A for a detailed review.

By contrast, one of the most influential theoretical settings for the design and analysis of optimization methods in deep learning has been the class of convex Lipschitz objectives. Adaptive methods such as AdaGrad [DHS11] were designed for convex and Lipschitz functions, which later gave rise to RMSProp and Adam [KB14]. It also extends naturally to matrix-valued methods such as Shampoo [GKS18], which can be viewed as a layerwise Kronecker approximation of full-matrix AdaGrad while still enjoying convergence guarantees in the convex Lipschitz setting. More recently, Schaipp et al. [Sch+25] showed that convex Lipschitz analyses can be surprisingly predictive of loss

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

dynamics in modern deep learning practice. In the inaugural AlgoPerf competition [Dah+23], the winning methods in the two rulesets were distributed Shampoo and Schedule-Free AdamW [Kas+25], both of which are closely tied to convex Lipschitz theory. In particular, Schedule-Free is motivated by optimal last-iterate convergence guarantees in the convex Lipschitz setting. Yet, despite the close conceptual connection between spectral descent and matrix-preconditioned methods, there is currently no convergence theory for Muon or spectral descent in the convex Lipschitz setting.

A central reason for this gap is that Muon may fail to converge on this class. We construct a convex Lipschitz objective, adapted from [Kar+19], for which the iterates of Muon remain bounded away from the solution. This lack of convergence holds even with a vanishing sequence of stepsizes, but requires an initialization in a narrow set. We then strengthen this construction and show that even local convergence can fail for almost every initialization, provided the momentum parameter $\beta$ is chosen in $[0, 1/2)$. This stands in sharp contrast both to the non-Euclidean smooth setting [She+25, Theorem 4.6] and to the standard Euclidean nonsmooth setting [GG23, Theorem 9.9], for which anytime convergence rates are known when $\beta = 0$.

Our analysis relies on the observation that, on diagonal matrices, Muon is equivalent to the signed momentum method. We prove this reduction in Lemma B.2. This reduction is especially notable because signed momentum has also been studied as a simple proxy for Adam [BH18; OG25]. Thus for diagonal matrices, Muon and Adam are in fact closely related methods. With this reduction, signed momentum becomes an exact specialization of Muon and therefore provides a tractable route to analysis. Building on this observation, we extend the counterexample of [Kar+19] for signed subgradient descent to prove non-convergence of signed momentum, and hence of Muon.

We then show that there is a modification of Muon that converges in the convex Lipschitz setting. We propose EF-MuonMax, a combination of norm-scaled spectral descent with error feedback [Kar+19], and provide an anytime convergence rate guarantee for this method. This EF modification can also be combined with any non-Euclidean descent method, resulting in a convergent method. Empirically, however, this fix does not appear to help in the regimes where Muon is typically used. Experiments on image classification (WideResNet-28-10 with 36M parameters on CIFAR-10 [KH+09]) and language modeling (a nanoGPT-like transformer with 206M parameters on FineWeb-Edu [Pen+24]) tasks show that EF-MuonMax underperforms MuonMax on both tasks, while EF-Muon improves over Muon on image classification but not language modeling. These practical problems lie outside the convex Lipschitz class.

Taken together, our results suggest that the convex Lipschitz class might not be the right theoretical lens for understanding Muon. In this regime, much like Adam [RKK18], Muon itself can fail to converge, and modifying it to recover convergence does not consistently improve practical performance. This is striking because closely related optimization ideas, from AdaGrad and Shampoo to Schedule-Free, are all meaningfully informed by convex Lipschitz theory. For Muon, by contrast, the standard convex Lipschitz framework does not yield a straightforward positive theory for unmodified Muon, unlike for AdaGrad/Shampoo-style methods. Any convex Lipschitz theory for Muon must either modify the algorithm, restrict the problem class, or exploit assumptions beyond worst-case Lipschitz convexity.

### 1.1 Contributions

Our contributions can be summarized as follows.

**Reduction of Muon to signed momentum.** We exhibit a reduction of Muon to signed momentum in Proposition 3.2 for all functions that rely only on the diagonal of a matrix.

**Counterexamples for Muon.** We give Counterexamples 3.3 and 3.4 that show that Muon does not converge on the class of convex and Lipschitz functions.

**Non-Euclidean updates are compressions.** We show that the Muon update (specifically: regularized Muon) can be viewed as a compression operator, and that the update of all regularized non-Euclidean subgradient methods are compression operators (Proposition 4.1). In the notation of Nesterov [Nes12], this means that all sharp operators are, up to a multiplicative constant, *compression* operators [Bez+23].

**Error feedback with momentum.** Using this interpretation as a compression, we develop the EF-M (error feedback with momentum) algorithm, which when combined with any non-Euclidean subgradient method results in a method that converges for convex and Lipschitz functions (Theorem 4.2).

## 2 Non-Euclidean subgradient methods

To simplify the discussion, we will consider a loss over a single linear layer. That is, let $W \in \mathbb{R}^{m\times n}$ be a matrix of parameters, and suppose our loss $f(W) \in \mathbb{R}$ is a function of this matrix, and our objective is to find $W$ that minimizes $f$. Extensions to the setting such as a Cartesian product of matrices, which is the parameter space for deep learning, can be done by using an appropriate max norm over the Cartesian product space, see Section C.3 and [BN24; Cra+25].

To define the local linearization of $f(W)$ we use the trace inner-product given by

$$\langle W, G\rangle = \operatorname{tr}\left(W^\top G\right) = \sum_{ij} W_{ij}G_{ij}.$$

This trace inner-product is equivalent to flattening the matrices $W$ and $G$ into vectors, and then using the standard inner-product over vectors.

### 2.1 Two flavors of subgradient descent

Let $W_t \in \mathbb{R}^{m\times n}$ be a given iterate at the $t$-th step. For a given norm $\|\cdot\|$ over matrices, we will consider two variants of subgradient descent for minimizing $f(W)$. *Regularized subgradient descent* is given by minimizing the local linearization with an additive regularizer

$$W_{t+1} \in \operatorname*{argmin}_{W\in\mathbb{R}^{m\times n}} \left(f(W_t) + \langle G_t, W - W_t\rangle + \frac{1}{2\lambda_t}\|W - W_t\|^2\right), \tag{1}$$

whereas *constrained subgradient descent* is given by minimizing the local linearization subject to a constraint

$$W_{t+1} \in \operatorname*{argmin}_{W\in\mathbb{R}^{m\times n}} \left(f(W_t) + \langle G_t, W - W_t\rangle\right) \quad \text{subject to} \quad \|W - W_t\| \le \lambda_t, \tag{2}$$

where $\lambda_t > 0$ is the learning rate, and $G_t \in \partial f(W_t)$. Note that (1) and (2) may have more than one solution, so to specify an algorithm one has to choose a particular solution $W_{t+1}$.

The question is, which norm over matrices should we use? In neural networks, matrix parameters $W$ represent linear layers that apply the transformation $x \mapsto Wx$, where $x$ denotes the current embedding or input vector. When stacking multiple linear layers with nonlinear activations, stable training often requires controlling the scale of the activations. In particular, the output $x' = Wx$ should remain bounded so that $\|x'\|_2 = O(1)$ [YSB24]. One way to enforce this is to control both the weight matrix and the input embedding using

$$\|Wx\|_2 \le \|W\|_{\mathrm{op}}\|x\|_2,$$

where $\|W\|_{\mathrm{op}}$ denotes the operator (spectral) norm of $W$, defined as

$$\|W\|_{\mathrm{op}} = \sup_{\|x\|_2=1} \|Wx\|_2. \tag{3}$$

The norm of input $\|x\|_2$ is often bounded due to some normalization layer, such as Layernorm [BKH16] or RMSnorm [ZS19]. To incentivize the operator norm to remain bounded, we can use it to regularize (1), or constrain (2), the update in subgradient descent. This leads to spectral subgradient descent.

### 2.2 Muon and spectral subgradient descent

Given a matrix $W$ we define its *polar factor* as $\operatorname{polar}(W) := UV^\top$, where $W = U\Sigma V^\top$ is the reduced SVD decomposition of $W$. If we use the operator norm $\|\cdot\| = \|\cdot\|_{\mathrm{op}}$ in constrained subgradient descent (2), then one of the solutions [Pet+25] is given by

$$W_{t+1} = W_t - \lambda_t \operatorname{polar}(G_t), \tag{specGD}$$

where $\lambda_t$ is the learning rate. The above method is known as *spectral subgradient descent*, and can be combined with momentum to give rise to the *Muon* algorithm[1]:

$$\begin{aligned} M_t &= \beta M_{t-1} + (1-\beta)G_t, \quad G_t \in \partial f(W_t), \quad M_{-1} = 0, \\ W_{t+1} &= W_t - \lambda_t \operatorname{polar}(M_t), \end{aligned} \tag{Muon}$$

where $\beta \in [0, 1)$ is the momentum coefficient. If we consider the regularized subgradient descent (1) instead of (2), then one obtains a similar algorithm [BN24; Cra+25] dubbed *regularized Muon* where the descent direction $\operatorname{polar}(M_t)$ is rescaled with the nuclear norm $\|M_t\|_{\mathrm{nuc}}$:

$$\begin{aligned} M_t &= \beta M_{t-1} + (1-\beta)G_t, \quad G_t \in \partial f(W_t), \quad M_{-1} = 0, \\ W_{t+1} &= W_t - \lambda_t \|M_t\|_{\mathrm{nuc}} \operatorname{polar}(M_t). \end{aligned} \tag{regMuon}$$

Note that computing the polar factor in (Muon) can be done efficiently on GPUs without needing to compute the SVD. Indeed the polar factor can be approximated by a matrix polynomial [Jor+24b; Ams+26], which is how[2] Muon is implemented in practice [Jor+24a; Kar26].

## 3 Counterexamples for Muon on nonsmooth convex problems

We are going to show that Muon cannot converge in the convex nonsmooth setting. More precisely, that it is not possible to derive an anytime convergence result for the Muon algorithm over the class of convex Lipschitz functions. The proofs of our claims can be found in Section B.4. Our main tool is the following convex Lipschitz function, which will serve as our main counterexample throughout.

**Definition 3.1.** Fix $m, n \geq 2$ and $c \in (0, 1)$. Define $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ by

$$f(W) = c|W_{11} + W_{22}| + |W_{11} - W_{22}|. \tag{4}$$

This is a diagonal function: $f(W)$ depends only on the (first two) diagonal entries of $W$, and as such can naturally be identified with a function over $\mathbb{R}^2$. In this setting, the Muon update simplifies considerably. Indeed the polar factor of its subgradients $\operatorname{polar}(G_t)$ in (Muon) is obtained by taking the elementwise sign of $G_t$. Consequently, Muon reduces exactly to signed momentum on this class of problems, as stated below; see Section B.2 for the proofs.

**Proposition 3.2** (Informal)**.** If $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ depends only on the main diagonal entries of its argument, then Muon is equivalent to the signed momentum algorithm.

As a consequence, for losses $f$ that only depend on the diagonal of $W$, we can use signed momentum to study the convergence of Muon. In particular, if we want to establish convergence of Muon on any class of functions, we should first study signed momentum since it is simpler.

In [Kar+19] the authors study signed subgradient descent applied to the function $\hat{f} : \mathbb{R}^2 \to \mathbb{R}$, $\hat{f}(x, y) = c|x+y| + |x-y|$ which inspired our Definition 3.1. They show in [Kar+19, Counterexample 2] that signed subgradient descent without momentum ($\beta = 0$) may fail to converge, regardless of how the stepsize schedule is tuned. Keeping in mind that spectral subgradient descent applied to our function $f$ is equivalent to signed subgradient descent on its diagonal elements, we can immediately deduce that Muon with no momentum ($\beta = 0$) may fail to converge too. But their counterexample suffers from two drawbacks, which we will improve upon.

First, their counterexample does not permit the use of *momentum*, which is standard in the implementation of Muon. We will show that the obstruction to convergence remains *for every choice of momentum parameter* $\beta \in [0, 1)$.

Second, their counterexample relies on using a subgradient that is not used in practice. To be precise, for the subgradient of the absolute value $|\cdot|$ evaluated at 0 they use $+1$ or $-1$. Though this is a

[1] We note that sometimes Muon uses a Nesterov style momentum [Jor+24b], though for simplicity we consider only the Polyak style momentum.

[2] As accessed on May 1, 2026.

valid subgradient, it goes against standard implementations in PyTorch or TensorFlow, in which the returned subgradient is simply 0. We get rid of this technical assumption by showing that Muon will fail to converge *no matter which subgradient is chosen*.

Our first Counterexample 3.3 below shows that Muon cannot converge globally on the class of convex and Lipschitz functions when using an *offline* decreasing stepsize schedule. See Figure 1 for an illustration of how the algorithm gets stuck, and see Section B.4 for the proofs.

**Counterexample 3.3** (Muon cannot converge: offline stepsizes)**.** Let $\beta \in [0, 1)$ and let $\{\lambda_t\}_{t=0}^{\infty}$ be a nonincreasing positive sequence chosen offline. Let $f$ be as in Definition 3.1 with $c = \frac{1-\beta}{2}$, and consider the iterates $W_t$ of (Muon) applied to $f$. Let $W^\star$ be any minimizer of $f$.

- If $\lim_{t\to\infty} \lambda_t > 0$, then there exists an open set of initializations such that

$$\text{for every } t \geq 0, \quad f(W_t) - \inf f \geq 1 - \beta \quad \text{and} \quad \|W_t - W^\star\|_F \geq \frac{1-\beta}{2}. \tag{5}$$

- If $\lambda_t$ is strictly decreasing to 0, then there exists an initialization such that (5) holds.

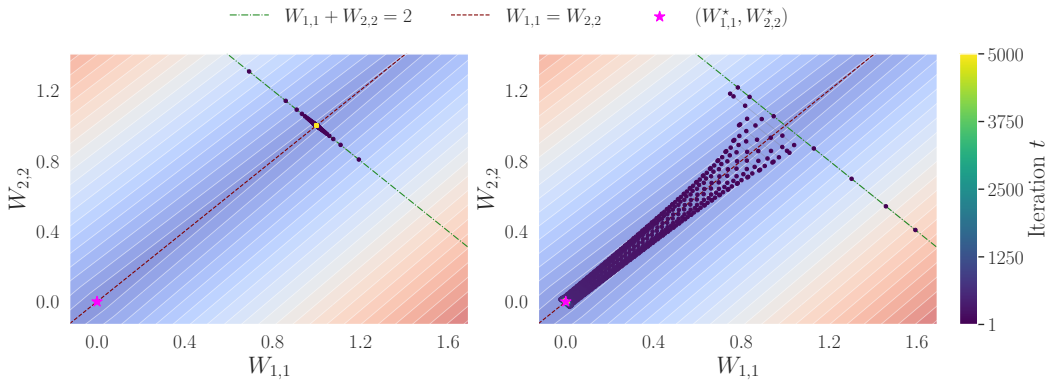


Figure 1: Level sets of $f$ given in Definition 3.1 in the $((W_t)_{1,1}, (W_t)_{2,2})$ plane. Left: Muon iterates. Right: EF-MuonMax iterates. The background shows function level sets: dark blue denotes low values, orange higher values, and red the highest values. The color bar applies only to the trajectory dots and indicates the iteration $t$.

We then relax our assumptions on the stepsizes in our second Counterexample 3.4, in exchange for a restricted use of momentum: $\beta \in [0, \frac{1}{2})$. More precisely we make no monotonicity assumption on the stepsizes, allowing as well for adaptive schedules depending on the past subgradients. This includes for instance stepsizes depending on $\|M_t\|_{\text{nuc}}$, such as in (regMuon). Further, this second result stands for almost every starting point $W_0$, and is for instance true with probability one when the initialization is drawn from an absolutely continuous distribution.

**Counterexample 3.4** (Muon cannot converge: adaptive stepsize and low momentum)**.** Consider $\beta \in [0, \frac{1}{2})$. Let $f$ be as in Definition 3.1 with $c = \frac{1}{2} - \beta$. Consider the iterates $W_t$ of (Muon) or (regMuon) applied to $f$ where the stepsizes $\lambda_t$ are chosen as an offline function of the previous subgradients $G_0, \ldots, G_t$. Then there exists a set of zero measure such that, for any initialization outside of this set, the iterates cannot converge to a minimizer of $f$.

Counterexample 3.4 shows that even a local convergence theory for Muon is not possible for convex and Lipschitz functions. More precisely, when $\beta \in [0, \frac{1}{2})$, almost every initialization in a ball centered at a minimizer $W^\star$ will fail to converge, no matter how small is the radius of this ball. This applies in particular to (specGD), which corresponds to the case $\beta = 0$. There is a striking difference with the smooth setting, where (specGD) is known to converge with a suitable choice of stepsizes.

## 4 Fixing Muon on Lipschitz nonsmooth functions

Counterexample 3.3 and Counterexample 3.4 show that Muon does not converge on the class of convex and Lipschitz functions. Here we propose to modify Muon by incorporating an error feedback structure, which leads to an algorithm (named EF-MuonMax) enjoying global convergence for convex and Lipschitz functions. Our approach has an even broader scope: we show that error feedback allows us to guarantee convergence *for any non-Euclidean* subgradient descent scheme incorporating momentum.

### 4.1 From subgradient descent to compressed gradients via LMOs

For any chosen norm $\|\cdot\|$, the regularized subgradient descent (1) combined with momentum can be equivalently rewritten [BN24, Proposition 1] in compact form

$$\begin{aligned} M_t &= \beta M_{t-1} + (1-\beta)G_t, \quad G_t \in \partial f(W_t), \\ W_{t+1} &\in W_t - \lambda_t \|M_t\|_* \mathsf{LMO}_{\|\cdot\|}(M_t), \end{aligned} \tag{6}$$

where the *linear maximization oracle* (LMO) and the *dual norm* are respectively defined as

$$\begin{aligned} \mathsf{LMO}_{\|\cdot\|}(M) &:= \operatorname*{argmax}_{X\in\mathbb{R}^{m\times n}:\|X\|\le 1} \langle X, M\rangle, \\ \|M\|_* &:= \max_{X\in\mathbb{R}^{m\times n}:\|X\|\le 1} \langle X, M\rangle. \end{aligned}$$

Formally the LMO is a set-valued operator, so in practice one must consider a particular selection in $\mathsf{LMO}_{\|\cdot\|}(M)$, a standard choice being the element with the least norm. For instance, when considering the operator norm, the least norm element of $\mathsf{LMO}_{\|\cdot\|_{\mathrm{op}}}(M)$ is exactly the polar factor $\mathrm{polar}(M)$, see Lemma C.2. This illustrates that (regMuon) is a particular case of (6) for the operator norm.

The main operation in (6) is the computation of

$$W \mapsto \|W\|_* \mathsf{LMO}_{\|\cdot\|}(W). \tag{7}$$

This operator is denoted $W^\#$ by Nesterov [Nes12, Eq. (1.8)] and called the *sharp operator* in later work, including [YTC15]. We show next that this sharp operator is, up to a multiplicative constant depending on the dimension of the problem, a *compression* operator [Bez+23]. This fact has been well-known for various choices of norms, but seems to be new when considering any arbitrary norm on a vector space.

**Proposition 4.1** (Dual subgradient compression operator)**.** Let $\|\cdot\|$ be any norm on $\mathbb{R}^{m\times n}$ with dual norm $\|\cdot\|_*$. Let $\alpha, \beta > 0$ satisfy for all $W \in \mathbb{R}^{m\times n}$

$$\alpha\|W\|_F \le \|W\| \le \beta\|W\|_F. \tag{8}$$

Then the operator $\mathcal{C}(W) := \alpha^2\|W\|_* \mathsf{LMO}_{\|\cdot\|}(W)$ is a compression operator, that is

$$\|W - \mathcal{C}(W)\|_F^2 \le \big(1-\delta\big)\|W\|_F^2, \qquad \delta = \frac{\alpha^2}{\beta^2}. \tag{9}$$

In particular with the operator norm $\|\cdot\|_{\mathrm{op}}$ we can take $\alpha^2 = \delta = 1/r$ with $r = \min\{m, n\}$.

The above lemma allows us to draw a connection between Muon and the literature on compressed algorithms [Bez+23; Dem+23]. Compressed algorithms are designed for solving optimization problems on distributed systems, which is a completely separate problem from ours. This connection is nevertheless useful because this literature shares a similar point with us: it is known that some compressors may introduce such a bias that convergence may fail [Bez+23], reminiscent of what we have learned in Section 3. More interesting strategies were proposed for restoring convergence by modifying the algorithm. One of them is the use of *error feedback* [Sei+14; SCJ18; Bez+23] which we detail and use next.

### 4.2 Error feedback on top of non-Euclidean SGD with momentum

Roughly speaking, error feedback consists in keeping track of the discrepancy between a true direction $M_t$ and its compression $\mathcal{C}(M_t)$, and to use it in a later iteration so as to rectify the trajectory of the iterates. We propose to incorporate error feedback into Muon, and more generally into (6) for any choice of norm, which gives rise to the EF-M algorithm below. We do so following the ideas from [Kar+19] where error feedback was proposed to restore the convergence of sign subgradient descent, which corresponds in our framework to the choices $\|\cdot\| = \|\cdot\|_\infty$ and $\beta = 0$.

**Algorithm 1** Error feedback with momentum (EF-M)

**Input:** stepsizes $\lambda_t > 0$, momentum parameter $\beta \in [0, 1)$, compressor $\mathcal{C} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$

Initialize $W_0 \in \mathbb{R}^{m\times n}$, $E_0 = 0$, and $M_{-1} = 0$
**for** $t = 0, 1, \ldots, T$ **do**
  $G_t \in \mathbb{R}^{m\times n}$ ▷ Sample a stochastic subgradient
  $M_t = \beta M_{t-1} + (1-\beta)G_t$ ▷ Update the momentum
  $P_t = E_t + \lambda_t M_t$ ▷ Form the error-corrected message
  $W_{t+1} = W_t - \mathcal{C}(P_t)$ ▷ Update the iterate with compressed weights
  $E_{t+1} = P_t - \mathcal{C}(P_t)$ ▷ Update the error feedback memory
**end for**

We can use Proposition 4.1 to instantiate Algorithm 1 for various choices of norm. When considering the operator norm $\|\cdot\|_{\text{op}}$ on matrices, taking

$$\mathcal{C}(W) = \frac{1}{\min\{m,n\}}\|W\|_{\text{nuc}}\,\text{polar}(W)$$

gives the single-matrix version of EF-MuonMax in Algorithm 2. The choice of the norm $\|\cdot\|_\infty$ on vectors gives EF-signSGD [Kar+19] when $\beta = 0$, and its momentum extension when $\beta > 0$. Since our framework is flexible enough to handle any normed vector space, we can also very easily specify Algorithm 1 for product spaces of matrices associated with different layers. The experiments in Section 5 use the product-space version of EF-MuonMax (Algorithm 4); the corresponding technical details can be found in Section C.3 and Section C.4.

Thanks to the error feedback mechanism in Algorithm 1, we are able to restore the convergence of every non-Euclidean stochastic momentum method (e.g., spectral descent and Muon), on the class of convex and Lipschitz functions. In Figure 1 we illustrate the convergence of EF-MuonMax on our counterexample from Definition 3.1.

**Theorem 4.2** (Anytime convex Lipschitz convergence of EF-M)**.** Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be convex and let $W^\star$ be a minimizer. Let $\mathcal{C}$ be a compression operator, meaning that for all $W \in \mathbb{R}^{m\times n}$,

$$\|\mathcal{C}(W) - W\|_F^2 \le (1-\delta)\|W\|_F^2, \qquad \delta \in (0, 1].$$

Consider the iterates of Algorithm 1, and assume that $G_t$ is a stochastic subgradient satisfying

$$\mathbb{E}_t[G_t] \in \partial f(W_t), \qquad \mathbb{E}_t\|G_t\|_F^2 \le \sigma^2.$$

If we consider a vanishing schedule for the stepsizes $\lambda_t = 1/\sqrt{t+1}$, then

$$\mathbb{E}\left[f(\bar{W}_T)\right] - f(W^\star) \le \frac{\|W_0 - W^\star\|_F^2}{2\sqrt{T+1}} + \sigma^2\left(\frac{2\sqrt{1-\delta}}{\delta} + \frac{\beta}{1-\beta} + \frac{1}{2}\right)\frac{1+\log(T+1)}{\sqrt{T+1}},$$

where $\bar{W}_T = \frac{1}{T+1}\sum_{t=0}^T W_t$.

In the Euclidean setting (i.e., $\mathcal{C} =$ identity and $\delta = 1$), Theorem 4.2 recovers the standard convergence rate of SGD. For general compressors, but turning off the momentum ($\beta = 0$), Theorem 4.2 recovers the convergence rate from [Kar+19, Theorem III]. A more general result can be found in the appendix, allowing for any choice of stepsizes $\lambda_t$. For EF-MuonMax with a single matrix, Proposition 4.1 gives a compression scaling of $1/r$, where $r = \min\{m, n\}$. This is in line with [She+25, Theorem 4.6], which is the only known convergence result for (specGD), under the assumption that the stepsizes scale like $1/r$.

## 5 Experiments

In this section we study the effects of error feedback on MuonMax and Muon. The results in the preceding sections were stated for a single parameter matrix. In applications, however, the loss function depends on a product space of parameters typically matrices and vectors associated with different layers. We choose a norm over this product space (see Section C.3 for details) so that the resulting regularized subgradient method (6) gives a variant of the MuonMax method [Cra+25; BN24]. MuonMax was in fact the first proposed variant of Muon [BN24], and we choose this variant because in a systematic study over all variants of Muon, MuonMax was found to perform best among the regularized variants of Muon [Cra+25].

We compare MuonMax to EF-MuonMax, the specialization of Algorithm 1 to the same product norm; see Section C.4. We also compare Muon, given by (Muon) on the matrix layers, to its error-feedback variant EF-Muon. EF-Muon uses the unscaled polar map, which is not a contractive compressor, so Theorem 4.2 does not cover this variant. All experiments use the `init2winit` [Gil+26] codebase on TPU v5 lite pods with $2 \times 4$ topology. We briefly describe both experiments below and leave more details to Section F in the appendix. Each variant of Muon also needs to be combined with a *paired optimizer*, which is used for updating the non-matrix parameters (e.g., biases). The choice of the paired optimizer depends on the task.

**Image classification.** We train a WideResNet-28-10 with 36M parameters on CIFAR-10 dataset. MuonMax and EF-MuonMax use signed momentum as the paired optimizer and 60 hyperparameter configurations per method. Muon and EF-Muon use Adam as the paired optimizer and the same 96-point grid per method. We select each method by its lowest final validation cross-entropy (CE). See Section F.1 for details.

**Language modeling.** We train a nanoGPT model with 206M parameters on the FineWeb-Edu 10B dataset [Pen+24]. MuonMax and EF-MuonMax use NAdamW as the paired optimizer and 30 hyperparameter configurations per method. Muon and EF-Muon use Adam and the same 96-point grid as in image classification. We again select by final validation CE; see Section F.2 for details.

**Results.** The results are shown in Table 1 and Figure 2. On image classification, EF-Muon achieves the lowest final validation CE of 0.2574, followed by MuonMax (0.2651), Muon (0.2746), and EF-MuonMax (0.3884). On language modeling, Muon achieves the lowest final validation CE of 3.0073, followed by MuonMax (3.2697), EF-MuonMax (3.4295), and EF-Muon (3.4858). Error feedback worsens MuonMax on both tasks; for Muon, it improves image classification but worsens language modeling.

| Task | Method | Learning rate | LR multiplier | Validation CE |
|---|---|---|---|---|
| Image classification | MuonMax | $5\times10^{-4}$ | 100 | 0.2651 |
| | EF-MuonMax | $5\times10^{-3}$ | 30 | 0.3884 |
| | Muon | 0.01 | 0.5 | 0.2746 |
| | EF-Muon | 0.05 | 30 | **0.2574** |
| Language modeling | MuonMax | 0.01 | 30 | 3.2697 |
| | EF-MuonMax | 0.01 | 30 | 3.4295 |
| | Muon | 0.002 | 1 | **3.0073** |
| | EF-Muon | 0.002 | 30 | 3.4858 |

Table 1: Configurations with the lowest final validation CE for each method and task.

**Discussion.** Error feedback worsens MuonMax on both tasks, while its effect on Muon depends on the task. Thus restoring convergence in the convex Lipschitz setting does not ensure better practical performance. The Muon comparison is empirical: its unscaled polar map lies outside the compressor assumptions of Theorem 4.2.

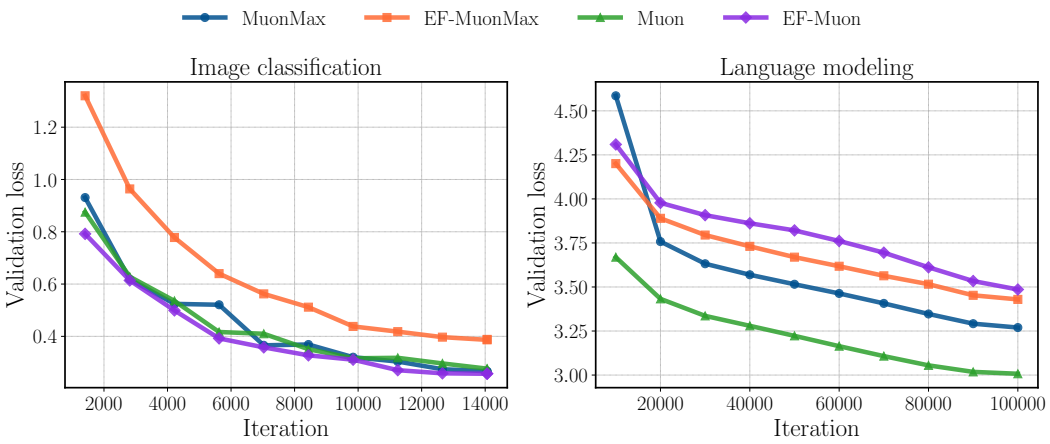


Figure 2: Validation CE loss of MuonMax, EF-MuonMax, Muon, and EF-Muon, each at the configuration with the lowest final validation CE. Left: WideResNet-28-10 with 36M parameters on CIFAR-10. Right: nanoGPT with 206M parameters on FineWeb-Edu 10B.

## 6 Conclusions and future directions

Our work is analogous in spirit to the early analysis of Adam by [RKK18], which showed that Adam can fail to converge on convex Lipschitz functions and proposed AMSGrad as a modification that restores convergence guarantees. In practice, however, AMSGrad has not displaced Adam, plausibly because its theoretical advantage does not consistently translate into empirical gains. We find a similar phenomenon for Muon: standard Muon can fail to converge on convex Lipschitz objectives, while a scaled error-feedback modification restores a convergence guarantee but does not consistently improve practical performance. This suggests that worst-case convex Lipschitz theory, although useful as a diagnostic, is not by itself the right model for explaining Muon's empirical success.

Our negative results therefore point to the need for a different theoretical testbed for Muon. Convex smoothness is one natural candidate, and has already led to useful insights [DD25]; alternatively, the relevant assumptions may involve other forms of structure that are absent from the nonsmooth Lipschitz model. Regardless of the eventual model class, the reduction in Proposition 3.2 provides a useful diagnostic tool: any proposed explanation of Muon should be consistent with its exact specialization to signed momentum on diagonal matrices.

## 7 Limitations

Each counterexample comes with certain limitations. Counterexample 3.3 requires decreasing stepsizes. This is fine for the variant of Muon in (Muon), which is the version most used in practice [Jor+24a], but it excludes the regularized variant of Muon (regMuon). Whereas our Counterexample 3.4 holds for adaptive stepsize sequences, including (regMuon), but not for the most used setting of $\beta$, since it only considers $\beta \in [0, \frac{1}{2})$. A further limitation is that neither counterexample considers a scheduler for $\beta$. Some of the most interesting proofs of convergence for momentum in the stochastic setting hold for $\beta_t \uparrow 1$, see [GG23]. Furthermore, though our counterexamples rule out a possible convergence result, they do not exclude the possibility of a complexity result, see Section D for details of this distinction. Finally, our theory does not cover approximate polar factors [Ams+26; Jor+24b] or Nesterov momentum, both of which are used in the experiments.

## Acknowledgments and disclosure of funding

The authors have no funding or competing interests to disclose.

## References


[Ams+26] N. Amsel et al. "The Polar Express: Optimal Matrix Sign Methods and their Application to the Muon Algorithm". In: *The Fourteenth International Conference on Learning Representations*. 2026.

[Bez+23] A. Beznosikov et al. "On Biased Compression for Distributed Learning". In: *Journal of Machine Learning Research* 24.276 (2023), pp. 1–50.

[BH18] L. Balles and P. Hennig. "Dissecting Adam: The Sign, Magnitude and Variance of Stochastic Gradients". In: *International Conference on Learning Representations*. 2018.

[BKH16] J. L. Ba, J. R. Kiros, and G. E. Hinton. *Layer Normalization*. 2016. arXiv: 1607.06450 [stat.ML].

[BN24] J. Bernstein and L. Newhouse. *Old Optimizer, New Norm: An Anthology*. 2024. arXiv: 2409.20325 [cs.LG].

[Car+15] D. E. Carlson et al. "Preconditioned Spectral Descent for Deep Learning". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015.

[CCC15] D. Carlson, V. Cevher, and L. Carin. "Stochastic Spectral Descent for Restricted Boltzmann Machines". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. Ed. by G. Lebanon and S. V. N. Vishwanathan. Vol. 38. Proceedings of Machine Learning Research. San Diego, California, USA: PMLR, May 2015, pp. 111–119.

[CLY25] D. Chang, Y. Liu, and G. Yuan. *On the Convergence of Muon and Beyond*. 2025. arXiv: 2509.15816 [cs.LG].

[Cra+25] M. T. Crawshaw et al. "An Exploration of Non-Euclidean Gradient Descent: Muon and its many Variants". In: *arXiv:2510.09827v1* (2025).

[Dah+23] G. E. Dahl et al. *Benchmarking Neural Network Training Algorithms*. 2023. arXiv: 2306.07179.

[DD25] D. Davis and D. Drusvyatskiy. "When do spectral gradient updates help in deep learning?" In: *arXiv preprint arXiv:2512.04299* (2025).

[Dee26] DeepSeek-AI. *DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence*. https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf. Technical report, hosted on Hugging Face. 2026.

[Dem+23] Y. Demidovich et al. "A Guide through the Zoo of Biased SGD". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 23158–23171.

[DHS11] J. Duchi, E. Hazan, and Y. Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: *J. Mach. Learn. Res.* 12 (July 2011), pp. 2121–2159.

[Doz16] T. Dozat. "Incorporating Nesterov Momentum into Adam". In: *ICLR 2016 Workshop Track* (2016).

[GG23] G. Garrigos and R. M. Gower. *Handbook of Convergence Theorems for (Stochastic) Gradient Methods*. Jan. 2023. arXiv: 2301.11235 [math].

[Gil+26] J. M. Gilmer et al. *init2winit: a JAX codebase for initialization, optimization, and tuning research*. Version 0.0.3. 2026.

[GKS18] V. Gupta, T. Koren, and Y. Singer. "Shampoo: Preconditioned Stochastic Tensor Optimization". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 1842–1850.

[GL13] G. H. Golub and C. F. van Loan. *Matrix Computations*. Fourth. JHU Press, 2013.

[Gru+25] K. Gruntkowska et al. *Error Feedback for Muon and Friends*. 2025. arXiv: 2510.00643 [cs.LG].

[Haz16] E. Hazan. "Introduction to online convex optimization". In: *Foundations and Trends in Optimization* 2.3-4 (2016), pp. 157–325.

[Jia+26] R. Jiang et al. "Adaptive Matrix Online Learning through Smoothing with Guarantees for Nonsmooth Nonconvex Optimization". In: *arXiv preprint arXiv:2602.08232* (2026).

[Jor+24a] K. Jordan et al. *Modded-nanoGPT: Speedrunning the nanoGPT baseline*. https://github.com/KellerJordan/modded-nanogpt. GitHub repository. 2024.

[Jor+24b] K. Jordan et al. "Muon: An optimizer for hidden layers in neural networks". In: *Blog post, retrieved May 2026* (2024).

[Kar+19] S. P. Karimireddy et al. "Error Feedback Fixes SignSGD and other Gradient Compression Schemes". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 3252–3261.

[Kar26] A. Karpathy. *NanoChat Challenge*. https://github.com/karpathy/nanochat/. 2026.

[Kas+25] P. Kasimbeg et al. "Accelerating neural network training: An analysis of the AlgoPerf competition". In: *The Thirteenth International Conference on Learning Representations*. 2025.

[KB14] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[KH+09] A. Krizhevsky, G. Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[Kov25] D. Kovalev. "Understanding gradient orthogonalization for deep learning via non-euclidean trust-region optimization". In: *arXiv preprint arXiv:2503.12645* (2025).

[LH25] J. Li and M. Hong. *A Note on the Convergence of Muon*. 2025. arXiv: 2502.02900 [math.OC].

[Liu+25] J. Liu et al. "Muon is scalable for LLM training". In: *arXiv preprint arXiv:2502.16982* (2025).

[LLS25] T. T.-K. Lau, Q. Long, and W. Su. "PolarGrad: A Class of Matrix-Gradient Optimizers from a Unifying Preconditioning Perspective". In: *arXiv preprint arXiv:2505.21799* (2025).

[Med+25] S. Medapati et al. "Training neural networks faster with minimal tuning using pre-computed lists of hyperparameters for NAdamW". In: *arXiv preprint arXiv:2503.03986* (2025). arXiv: 2503.03986.

[MG15] J. Martens and R. Grosse. "Optimizing neural networks with kronecker-factored approximate curvature". In: *International conference on machine learning*. PMLR. 2015, pp. 2408–2417.

[Nes12] Y. Nesterov. "Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems". In: *SIAM Journal on Optimization* 22.2 (Jan. 2012), pp. 341–362.

[OG25] A. Orvieto and R. M. Gower. "In Search of Adam's Secret Sauce". In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2025.

[Pen+24] G. Penedo et al. "The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale". In: *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2024.

[Pet+25] T. Pethick et al. "Training deep learning models with norm-constrained lmos". In: *arXiv preprint arXiv:2502.07529* (2025).

[Pey15] J. Peypouquet. *Convex Optimization in Normed Spaces*. SpringerBriefs in Optimization. Cham: Springer International Publishing, 2015.

[Ria+26] A. Riabinin et al. "Where Does Warm-Up Come From? Adaptive Scheduling for Norm-Constrained Optimizers". In: *arXiv preprint arXiv:2602.05813* (2026).

[RKK18] S. J. Reddi, S. Kale, and S. Kumar. "On the Convergence of Adam and Beyond." In: *ICLR*. 2018.

[Sch+25] F. Schaipp et al. "The Surprising Agreement Between Convex Optimization Theory and Learning-Rate Scheduling for Large Model Training". In: *Proceedings of the 42nd International Conference on Machine Learning*. Ed. by A. Singh et al. Vol. 267. Proceedings of Machine Learning Research. PMLR, July 2025, pp. 53267–53294.

[Sch25] F. Schaipp. *Optimization Benchmark for Diffusion Models on Dynamical Systems*. 2025. arXiv: 2510.19376 [cs.LG].

[SCJ18] S. U. Stich, J.-B. Cordonnier, and M. Jaggi. "Sparsified SGD with Memory". In: *Advances in neural information processing systems* 31 (2018).

[Sei+14] F. Seide et al. "1-Bit Stochastic Gradient Descent and Its Application to Data-Parallel Distributed Training of Speech DNNs." In: *Interspeech*. Vol. 2014. Singapore, 2014, pp. 1058–1062.

[She+25] W. Shen et al. *On the Convergence Analysis of Muon*. 2025. arXiv: 2505.23737 [stat.ML].

[SNI25] N. Sato, H. Naganuma, and H. Iiduka. *Convergence Bound and Critical Batch Size of Muon Optimizer*. 2025. arXiv: 2507.01598 [cs.LG].

[Su25] W. Su. *Isotropic Curvature Model for Understanding Deep Learning Optimization: Is Gradient Orthogonalization Optimal?* 2025. arXiv: 2511.00674 [math.OC].

[SW25] M.-E. Sfyraki and J.-K. Wang. *Lions and Muons: Optimization via Stochastic Frank–Wolfe*. 2025. arXiv: 2506.04192 [math.OC].

[Tea+25] K. Team et al. "Kimi K2: Open agentic intelligence". In: *arXiv preprint arXiv:2507.20534* (2025).

[TH12] T. Tieleman and G. Hinton. "Lecture 6.5-RMSProp, coursera: Neural networks for machine learning". In: *University of Toronto, Technical Report* 6 (2012), p. 2012.

[Vya+24] N. Vyas et al. "Soap: Improving and stabilizing shampoo using adam". In: *arXiv preprint arXiv:2409.11321* (2024).

[Wat92] G. A. Watson. "Characterization of the subdifferential of some matrix norms". In: *Linear Algebra Appl* 170.1 (1992), pp. 33–45.

[Xie+25] S. Xie et al. "A Tale of Two Geometries: Adaptive Optimizers and Non-Euclidean Descent". In: *arXiv preprint arXiv:2511.20584* (2025).

[YSB24] G. Yang, J. B. Simon, and J. Bernstein. *A Spectral Condition for Feature Learning*. 2024. arXiv: 2310.17813 [cs.LG].

[YTC15] A. Yurtsever, Q. Tran-Dinh, and V. Cevher. "A Universal Primal-Dual Convex Optimization Framework". In: *Advances in Neural Information Processing Systems*. Vol. 28. 2015.

[ZS19] B. Zhang and R. Sennrich. "Root Mean Square Layer Normalization". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

## Contents


**1 Introduction** **1**

1.1 Contributions . . . . . 2

**2 Non-Euclidean subgradient methods** **3**

2.1 Two flavors of subgradient descent . . . . . 3

2.2 Muon and spectral subgradient descent . . . . . 3

**3 Counterexamples for Muon on nonsmooth convex problems** **4**

**4 Fixing Muon on Lipschitz nonsmooth functions** **6**

4.1 From subgradient descent to compressed gradients via LMOs . . . . . 6

4.2 Error feedback on top of non-Euclidean SGD with momentum . . . . . 7

**5 Experiments** **8**

**6 Conclusions and future directions** **9**

**7 Limitations** **9**

**A Prior work** **14**

**B Proofs in Section 3** **15**

B.1 Signed momentum . . . . . 15

B.2 A reduction of Muon to signed momentum for diagonal functions . . . . . 15

B.3 Basic results on our counterexample function from Definition 3.1 . . . . . 17

B.4 Proofs of Counterexamples 3.3 and 3.4 . . . . . 19

B.5 Non-diagonal counterexamples . . . . . 24

**C Proofs in Section 4** **24**

C.1 LMOs and compression operators . . . . . 24

C.2 Basic choices of LMO-based compressors and EF-MuonMax . . . . . 25

C.3 LMO-based compression operator over layers . . . . . 27

C.4 MuonMax . . . . . 29

C.5 Convergence bounds for error feedback with momentum . . . . . 29

**D Convergence, complexity and limitations that depend on $\beta$** **34**

**E Experiments on counterexample function** **35**

**F Deep learning experiments** **36**

F.1 Image classification . . . . . 37

F.2 Language modeling . . . . . 38

# A Prior work

Here we provide a more extensive review of prior work on Muon and spectral descent.

**Spectral descent and Muon.** General idea behind Muon [Jor+24b] is based on spectral descent, which corresponds to taking the update using steepest descent over spectral norm ball instead of Euclidean norm [CCC15; Car+15]. Recent work constructs modular norms which allow one to apply steepest descent under different norms to neural networks with many layers [BN24]. Muon and its variants, e.g., Scion and PolarGrad, can also be interpreted through the same lens [Jor+24b; Pet+25; LLS25].

**Smooth analyses.** Existing convergence results for Muon rely on smoothness assumption. In [DD25] authors study conditions under which a spectral update gives a larger one-step decrease than a Euclidean update. Su [Su25] analyzes orthogonalized updates through an isotropic curvature model and shows when spectrum homogenization and full orthogonalization are favored. Kovalev [Kov25] interprets subgradient orthogonalization as a non-Euclidean trust-region method and analyzes stochastic momentum variants that recover Muon.

Several papers also prove convergence guarantees for smooth Muon variants. Li et al. [LH25] study both constrained and regularized formulations. Shen et al. [She+25] analyze constrained Muon with and without momentum, including a star-convex setting. Sato et al. [SNI25] consider momentum, Nesterov acceleration, and weight decay, with an emphasis on batch-size effects. Sfyraki et al. [SW25] study constrained Muon together with weight decay, gradient clipping, and variance reduction. Chang et al. [CLY25] analyze variance-reduced Muon and also obtain stronger guarantees under additional growth conditions. Xie et al. [Xie+25] consider adaptive smoothness and adaptive variance in a non-Euclidean setting. Under a generalized smoothness assumption Takáč et al. [Ria+26] examined adaptive scheduling for norm-constrained optimizers, including Muon, and its implications for warm-up strategies. One of the few references that consider the nonsmooth setting is [Jia+26] where they analyzed smoothed variants of spectral descent with guarantees for nonsmooth and nonconvex optimization.

These works are complementary to ours. They explain why Muon can perform well in smooth regimes; our focus is the convex and Lipschitz function class, which also includes the nonsmooth functions.

**Convex nonsmooth analyses and adaptive methods.** The convex Lipschitz model is one of the fundamental models that gave rise to the development and analysis of modern adaptive first-order methods. AdaGrad [DHS11] is an adaptive subgradient method for online learning and stochastic optimization based on follow-the-regularized-leader family of algorithms [Haz16]. Adam [KB14] combines ideas from AdaGrad and RMSProp [TH12], and has become a standard optimizer choice in deep learning.

This perspective also extends to matrix-based optimizers. Shampoo [GKS18] can be viewed as a matrix version of AdaGrad which replaces full-matrix preconditioner by a Kronecker product, with regret-based convergence analysis. K-FAC [MG15] is derived by approximating natural-gradient methods: it uses Kronecker-factored approximations of Fisher (or Gauss-Newton) curvature blocks to define a preconditioned gradient step. SOAP [Vya+24] combines these ideas with Adam-style adaptivity, using Shampoo's eigenspaces while applying coordinatewise scaling in the rotated basis.

Thus, although Shampoo, K-FAC, and SOAP are matrix-based, they remain adaptive preconditioning methods: they change the metric in which gradients are measured. Muon is different, as it relies on the orthogonalization of the update direction.

More recently, Schaipp et al. [Sch+25] showed that analyses under the convex and Lipschitz model are particularly informative for understanding training dynamics in deep learning. Therefore, it is also important to study Muon under the model that led to the development of widely used adaptive optimizers.

**Error feedback.** In [Kar+19] authors gave convex and Lipschitz counterexample for sign subgradient descent, and proved that sign subgradient descent with error feedback mechanism restores convergence. More recently, [Gru+25] extended error feedback analysis to Muon and LMO-based methods to distributed setting under the generalized smoothness assumptions. In particular, they use compression operator for communication between workers and error feedback to guarantee convergence in distributed setting. In contrast, we interpret LMO itself as compression operator and use error feedback to guarantee convergence in the nonsmooth setting.

# B Proofs in Section 3

## B.1 Signed momentum

When our parameters are vectors $w \in \mathbb{R}^d$, a special case of non-Euclidean subgradient descent is given by *signed subgradient descent*. Indeed, when we use the infinity norm $\|w\|_\infty := \max_{i=1,\dots,d} |w_i|$ in (2), then one realization of the algorithm is given by

$$w_{t+1} = w_t - \lambda_t \operatorname{sign}(g_t), \qquad g_t \in \partial f(w_t). \tag{signGD}$$

Here $\operatorname{sign}$ is defined as the standard sign operator

$$\operatorname{sign}(x) = \begin{cases} +1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0, \end{cases} \tag{10}$$

and in what follows we will allow ourselves to apply it elementwise to vectors or matrices while keeping the same overloaded notation. Signed subgradient descent can be combined with momentum, giving rise to the *signed momentum* algorithm (or Signum), defined for some momentum parameter $\beta \in [0, 1)$ as

$$\begin{aligned} m_t &= \beta m_{t-1} + (1-\beta) g_t, \quad g_t \in \partial f(w_t), \quad m_{-1} = 0, \\ w_{t+1} &= w_t - \lambda_t \operatorname{sign}(m_t). \end{aligned} \tag{signMomentum}$$

Note that (signMomentum) can also be applied to matrices $W \in \mathbb{R}^{m\times n}$ seen as vectors: in this case the sign operation $\operatorname{sign}(M_t)$ is applied on every coefficient of the matrix.

## B.2 A reduction of Muon to signed momentum for diagonal functions

Before talking about diagonal functions, let us set some definitions regarding diagonal rectangular matrices. Given a vector $a \in \mathbb{R}^{\min\{n,m\}}$, we define the diagonal matrix $\operatorname{Diag}(a) \in \mathbb{R}^{m\times n}$ via

$$\operatorname{Diag}(a)_{ij} = \begin{cases} 0 & i \neq j \\ a_i & i = j, \end{cases} \qquad i = 1, \dots, m, \quad j = 1, \dots, n.$$

We further say that a rectangular matrix $A \in \mathbb{R}^{m\times n}$ is *diagonal* if $A = \operatorname{Diag}(a)$ for some vector $a \in \mathbb{R}^{\min\{n,m\}}$. We also introduce the adjoint operator $\operatorname{diag} : \mathbb{R}^{m\times n} \to \mathbb{R}^{\min\{m,n\}}$, where $\operatorname{diag}(A)$ returns the vector composed of the diagonal elements of $A$. A first important fact is that computing the polar factor of a diagonal matrix is the same as applying the $\operatorname{sign}$ operator component-wise to this matrix.

**Lemma B.1** (Polar factor of a rectangular diagonal matrix)**.** Let $A \in \mathbb{R}^{m\times n}$ be a diagonal matrix. Then

$$\operatorname{polar}(A) = \operatorname{sign}(A).$$

*Proof of Lemma B.1.* Let $p = \min\{m, n\}$ and $A = \operatorname{Diag}(a)$ for some $a \in \mathbb{R}^p$. Let $I = \{i \in \{1, \dots, p\} \mid a_i \neq 0\}$ and $r = |I| = \operatorname{rank}(A)$. Index $I = \{i_1, \dots, i_r\}$. Let $e'_1, \dots, e'_m$ be the canonical basis of $\mathbb{R}^m$, and $e_1, \dots, e_n$ be the canonical basis of $\mathbb{R}^n$. Define $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ via their columns:

$$u_j = \operatorname{sign}(a_{i_j}) e'_{i_j} \in \mathbb{R}^m, \qquad v_j = e_{i_j} \in \mathbb{R}^n, \qquad j = 1, \dots, r,$$

and set $\Sigma = \operatorname{diag}(|a_{i_1}|, \dots, |a_{i_r}|) \in \mathbb{R}^{r\times r}$. Then

$$U\Sigma V^\top = \sum_{j=1}^{r} \operatorname{sign}(a_{i_j}) |a_{i_j}| e'_{i_j} e_{i_j}^\top = \sum_{j=1}^{r} a_{i_j} e'_{i_j} e_{i_j}^\top = A,$$

so $A = U\Sigma V^\top$ is a reduced SVD of $A$. Therefore, since coordinates with $a_i = 0$ do not appear in the reduced SVD

$$\begin{aligned}
\operatorname{polar}(A) &= UV^\top \\
&= \sum_{j=1}^{r} \operatorname{sign}(a_{i_j}) e'_{i_j} e_{i_j}^\top \\
&= \sum_{i=1}^{p} \operatorname{sign}(a_i) e'_i e_i^\top \\
&= \operatorname{Diag}(\operatorname{sign}(a_1), \dots, \operatorname{sign}(a_p)) \\
&= \operatorname{sign}(A),
\end{aligned}$$

where the last equality comes from the fact that $A$ has zero coefficients outside its diagonal, and that $\operatorname{sign}(0) = 0$. □

Our main reduction result will show that (Muon) and (signMomentum) are equivalent for diagonal functions. This is the formal statement corresponding to our informal Proposition 3.2.

**Lemma B.2** (Muon reduction)**.** Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a *diagonal function*, in the sense that

$$f(W) = f_{\text{diag}}(\operatorname{diag}(W))$$

for some function $f_{\text{diag}} : \mathbb{R}^{\min\{m,n\}} \to \mathbb{R}$. Assume that $f_{\text{diag}}$ is a convex and continuous function. If $W_t$ are the iterates of (Muon) applied to $f$, then they also follow the (signMomentum) algorithm applied to $f$.

*Proof.* Let us first establish that every subgradient $G \in \partial f(W)$ is a diagonal matrix. To see this, use the fact that $f = f_{\text{diag}} \circ \operatorname{diag}$ where $f_{\text{diag}}$ is convex continuous and $\operatorname{diag} : \mathbb{R}^{m\times n} \to \mathbb{R}^{\min\{m,n\}}$ is a linear operator. That allows to write [Pey15, Proposition 3.28]

$$\partial f(W) = \operatorname{diag}^*(\partial f_{\text{diag}}(\operatorname{diag}(W))),$$

where the adjoint operator $\operatorname{diag}^*$ is exactly Diag. In other words the subgradients of $f$ belong to the range of Diag, so they are indeed diagonal matrices.

At every iteration, the momentum update reads

$$M_t = \beta M_{t-1} + (1-\beta) G_t,$$

where $G_t \in \partial f(W_t)$ is a diagonal matrix, and the momentum matrix is initialized with $M_{-1} = 0$. By induction, $M_t$ is diagonal for every $t = 0, 1, \dots$. Define $m_t := \operatorname{diag}(M_t)$ which verifies $M_t = \operatorname{Diag}(m_t)$. We can then use Lemma B.1 to establish

$$\operatorname{polar}(M_t) = \operatorname{sign}(M_t).$$

Consequently the iterates $W_t$ of (Muon) verify

$$\begin{aligned} M_t &= \beta M_{t-1} + (1-\beta) G_t, \quad G_t \in \partial f(W_t), \\ W_{t+1} &= W_t - \lambda_t \operatorname{sign}(M_t), \end{aligned}$$

which means that indeed $W_t$ follows the (signMomentum) algorithm applied to $f$.

□

### B.3 Basic results on our counterexample function from Definition 3.1

Recall that our counterexample function in Definition 3.1 is defined as

$$f(W) = c|W_{11} + W_{22}| + |W_{11} - W_{22}|, \quad c \in (0,1).$$

It is not only a diagonal function, but only depends on the first two diagonal elements of $W$. In view to exploit this sparse dependency on the coefficients of $W$, and for simplifying subsequent notations, we introduce

$$\operatorname{diag}_2 : \mathbb{R}^{m\times n} \to \mathbb{R}^2, \quad \operatorname{diag}_2(W) = (W_{11}, W_{22}),$$

which is a linear operator returning the first two diagonal elements of a matrix. Note also that $f$ is nonsmooth due to the presence of absolute values, so to compute its subgradients we will need a notation for the subgradient of the absolute value, which is the set-valued sign function

$$\operatorname{SIGN}(x) := \begin{cases} +1 & x > 0 \\ [-1,+1] & x = 0 \\ -1 & x < 0. \end{cases}$$

We now collect a few facts about this function $f$.

**Lemma B.3.** Let $f$ be the function from Definition 3.1.

1. $f(W) = f_{\text{diag}}(\operatorname{diag}_2(W))$ for $f_{\text{diag}} : \mathbb{R}^2 \to \mathbb{R}$ and
$$f_{\text{diag}}(w_1, w_2) = c|w_1 + w_2| + |w_1 - w_2|.$$
2. $f_{\text{diag}}$ and $f$ are convex, continuous and nonsmooth.
3. $\inf f = \inf f_{\text{diag}} = 0$.
4. $\operatorname{argmin} f_{\text{diag}} = \{0\}$ and $W^\star$ minimizes $f$ if and only if $W^\star_{1,1} = W^\star_{2,2} = 0$.
5. The subdifferential of $f_{\text{diag}}$ is
$$\partial f_{\text{diag}}(w) = c \operatorname{SIGN}(w_1 + w_2) \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \operatorname{SIGN}(w_1 - w_2) \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{11}$$
6. The functions $f_{\text{diag}} : (\mathbb{R}^2, \|\cdot\|_2) \to (\mathbb{R}, |\cdot|)$ and $f : (\mathbb{R}^{m\times n}, \|\cdot\|_F) \to (\mathbb{R}, |\cdot|)$ are $G$-Lipschitz continuous, with $G = \sqrt{2(1+c^2)} \le 2$.

*Proof.*

1. Immediate.
2. Trivial.
3. $f(W) \ge 0$, and $f(0) = 0$, therefore $\inf_W f(W) = 0$. The same holds for $f_{\text{diag}}$.
4. Immediate.

5. This follows from basic sum and composition rules for convex continuous functions and linear operators [Pey15, Proposition 3.28 & Theorem 3.30], and $\partial|x| = \mathrm{SIGN}(x)$.

6. The Lipschitz constant of the convex function $f_{\mathrm{diag}}$ is equal to $\sup_{w\in\mathbb{R}^2} \sup_{g\in\partial f_{\mathrm{diag}}(w)} \|g\|_2$. From the previous point, we know that

$$g \in \partial f_{\mathrm{diag}}(w) \implies g = (cs_1 + s_2, cs_1 - s_2),$$

for some $s_1 \in \mathrm{SIGN}(w_1 + w_2)$ and $s_2 \in \mathrm{SIGN}(w_1 - w_2)$. Then

$$\|g\|_2^2 = 2c^2 s_1^2 + 2s_2^2 \leq 2(c^2 + 1).$$

It follows that $f_{\mathrm{diag}}$ is $\sqrt{2(c^2+1)}$-Lipschitz continuous, with $\sqrt{2(c^2+1)} \leq 2$ coming from the fact that $c \leq 1$.

The Lipschitz constant of the convex function $f$ is equal to $\sup_{W\in\mathbb{R}^{m\times n}} \sup_{G\in\partial f(W)} \|G\|_F$. Using a chain rule as before leads to

$$G \in \partial f(W) \implies G = \mathrm{diag}_2^*(g),$$

where $g \in \partial f_{\mathrm{diag}}(\mathrm{diag}_2(W))$. Because $\mathrm{diag}_2$ is a projection between Euclidean spaces, we have $\|\mathrm{diag}_2\|_{\mathrm{op}} \leq 1$. And because we just saw that $\|g\| \leq \sqrt{2(c^2+1)}$, we conclude that $\|G\|_F \leq \sqrt{2(c^2+1)}$ as well.

□

Now we present an analog to the reduction Lemma B.2, which shows that studying (Muon) on our counterexample is equivalent to studying the two-dimensional iterates of (signMomentum).

**Lemma B.4** (Muon reduction on our counterexample)**.** Let $f$ be the function from Definition 3.1, that we decompose as

$$f(W) = f_{\mathrm{diag}}(\mathrm{diag}_2(W)), \qquad f_{\mathrm{diag}}(w_1, w_2) = c|w_1 + w_2| + |w_1 - w_2|.$$

If $W_t \in \mathbb{R}^{m\times n}$ are the iterates of (Muon) applied to $f$, then the iterates $w_t = \mathrm{diag}_2(W_t) \in \mathbb{R}^2$ follow the (signMomentum) algorithm applied to $f_{\mathrm{diag}}$, and we have $f(W_t) = f_{\mathrm{diag}}(w_t)$.

*Proof.* Let us first establish that every subgradient $G \in \partial f(W)$ is a diagonal matrix. To see this, use the fact that $f = f_{\mathrm{diag}} \circ \mathrm{diag}_2$ where $f_{\mathrm{diag}}$ is convex continuous and $\mathrm{diag}_2 : \mathbb{R}^{m\times n} \to \mathbb{R}^2$ is a linear operator. That allows to write [Pey15, Proposition 3.28]

$$\partial f(W) = \mathrm{diag}_2^*(\partial f_{\mathrm{diag}}(\mathrm{diag}_2(W))).$$

It is a simple exercise to see that $\mathrm{diag}_2^*$ is exactly

$$\mathrm{diag}_2^* : \mathbb{R}^2 \to \mathbb{R}^{m\times n}, \quad \mathrm{diag}_2^*(w_1, w_2) = \mathrm{Diag}(w_1, w_2, 0, \ldots, 0).$$

In particular we see that every subgradient of $f$ is a diagonal matrix.

Let us now consider the iterates $W_t$ of (Muon). At every iteration, the momentum update reads

$$M_t = \beta M_{t-1} + (1-\beta) G_t,$$

where $G_t \in \partial f(W_t)$ is a diagonal matrix. We further know that the momentum matrix is initialized with $M_{-1} = 0$ which is also diagonal. By induction, we deduce that $M_t$ is diagonal for every $t \geq 0$. We can then use Lemma B.1 to establish that

$$\mathrm{polar}(M_t) = \mathrm{sign}(M_t).$$

Consider now $w_t := \mathrm{diag}_2(W_t)$, and use the linearity of $\mathrm{diag}_2$ to write

$$\begin{aligned} w_{t+1} &= \mathrm{diag}_2(W_{t+1}) \\ &= \mathrm{diag}_2(W_t) - \lambda_t \,\mathrm{diag}_2(\mathrm{polar}(M_t)) \\ &= w_t - \lambda_t \,\mathrm{diag}_2(\mathrm{sign}(M_t)). \end{aligned}$$

Because $\mathrm{diag}_2 \circ \mathrm{sign} = \mathrm{sign} \circ \mathrm{diag}_2$, we deduce that

$$w_{t+1} = w_t - \lambda_t \,\mathrm{sign}(\mathrm{diag}_2(M_t)) = w_t - \lambda_t \,\mathrm{sign}(m_t)$$

after defining $m_t := \mathrm{diag}_2(M_t)$. Moreover we have seen earlier that $G_t = \mathrm{diag}_2^*(g_t)$, where $g_t \in \partial f_{\mathrm{diag}}(\mathrm{diag}_2(W_t)) = \partial f_{\mathrm{diag}}(w_t)$. This means that $\mathrm{diag}_2(G_t) = g_t$. We can therefore write

$$m_t = \mathrm{diag}_2(M_t) = \beta \,\mathrm{diag}_2(M_{t-1}) + (1-\beta)\,\mathrm{diag}_2(G_t) = \beta m_{t-1} + (1-\beta) g_t.$$

We finally have verified that

$$\begin{aligned} m_t &= \beta m_{t-1} + (1-\beta) g_t, \quad g_t \in \partial f_{\mathrm{diag}}(w_t), \\ w_{t+1} &= w_t - \lambda_t \,\mathrm{sign}(m_t), \end{aligned}$$

meaning that $w_t$ follows the (signMomentum) algorithm applied to $f_{\mathrm{diag}}$.

□

### B.4 Proofs of Counterexamples 3.3 and 3.4

**Counterexample 3.3** (Muon cannot converge: offline stepsizes)**.** Let $\beta \in [0,1)$ and let $\{\lambda_t\}_{t=0}^{\infty}$ be a nonincreasing positive sequence chosen offline. Let $f$ be as in Definition 3.1 with $c = \frac{1-\beta}{2}$, and consider the iterates $W_t$ of (Muon) applied to $f$. Let $W^\star$ be any minimizer of $f$.

- If $\lim_{t\to\infty} \lambda_t > 0$, then there exists an open set of initializations such that

$$\text{for every } t \geq 0, \quad f(W_t) - \inf f \geq 1 - \beta \quad \text{and} \quad \|W_t - W^\star\|_F \geq \frac{1-\beta}{2}. \tag{5}$$

- If $\lambda_t$ is strictly decreasing to 0, then there exists an initialization such that (5) holds.

*Proof of Counterexample 3.3.* First of all, we know thanks to Lemma B.4 that we mostly need to examine the iterates of (signMomentum) applied to the function $f_{\mathrm{diag}} : \mathbb{R}^2 \to \mathbb{R}$,

$$f_{\mathrm{diag}}(w) = c|w_1 + w_2| + |w_1 - w_2|.$$

The iterates of (signMomentum) verify in this case

$$g_t \in \partial f_{\mathrm{diag}}(w_t), \qquad m_t = \beta m_{t-1} + (1-\beta) g_t, \qquad w_{t+1} = w_t - \lambda_t \,\mathrm{sign}(m_t).$$

Before starting our analysis of this algorithm, we need to focus on the following quantities, that we will later need to track along the iterations:

$$\lambda := \lim_{t\to\infty} \lambda_t, \quad R_t := \frac{\lambda}{2} + S_t, \quad S_t := \sum_{s=0}^{\infty} (-1)^s (\lambda_{t+s} - \lambda), \quad t = 0, 1, \ldots.$$

Our assumptions that $\lambda_t$ is nonincreasing and converging to $\lambda$ imply, via the alternating-series test, that $S_t$ is a convergent series, and therefore is well defined. We can further infer that $S_t \geq 0$, and even that $S_t > 0$ in the case that $\lambda_t$ is strictly decreasing (which we assume when $\lambda = 0$). This implies that $R_t > 0$ holds true, whether $\lambda = 0$ or not. Another point to observe is that $S_t$ is, up to a sign change, the tail of the series $S_0$:

$$S_t = \sum_{k=t}^{\infty} (-1)^{k-t} (\lambda_k - \lambda) = (-1)^t \sum_{k=t}^{\infty} (-1)^k (\lambda_k - \lambda).$$

Because we know that $S_0$ is a converging series, the above means that $S_t$ tends to zero when $t \to +\infty$. A last point to be made is that $R_t$ can be equivalently defined through the recursion $R_{t+1} = \lambda_t - R_t$.

Indeed

$$\begin{aligned} R_{t+1} &= \frac{\lambda}{2} + S_{t+1} \\ &= \frac{\lambda}{2} + \sum_{s=0}^{\infty} (-1)^s (\lambda_{t+s+1} - \lambda) \\ &= \frac{\lambda}{2} - \sum_{s=1}^{\infty} (-1)^s (\lambda_{t+s} - \lambda) \\ &= \frac{\lambda}{2} + (\lambda_t - \lambda) - \sum_{s=0}^{\infty} (-1)^s (\lambda_{t+s} - \lambda) \\ &= \lambda_t - \frac{\lambda}{2} - S_t = \lambda_t - R_t. \end{aligned}$$

Let us now start the analysis of the iterates $w_t$ of (signMomentum). Consider initializing the algorithm with $w_0 \in \mathcal{W}$, where the set of initializations $\mathcal{W}$ is defined by

$$\mathcal{W} := \left\{ r \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (R_0 + \delta) \begin{bmatrix} 1 \\ -1 \end{bmatrix} \;\middle|\; r \geq 1,\ |\delta| < R_t,\ t = 0, 1, \dots \right\}.$$

Here a couple of observations need to be made, depending on the value of $\lambda$. If $\lambda = 0$ then $R_t > 0$ but also $R_t = S_t \to 0$. In this case $\delta$ must be equal to zero, and $\mathcal{W}$ is nonempty. If $\lambda > 0$ then $R_t = \frac{\lambda}{2} + S_t \geq \frac{\lambda}{2}$. In this case any choice $|\delta| < \frac{\lambda}{2}$ is valid, and $\mathcal{W}$ has a nonempty interior.

Consider now that we take $w_0 \in \mathcal{W}$ with parameters $r, \delta$. We are going to prove by induction that, for every $t = 0, 1, \dots$

$$w_t = r \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (\delta + (-1)^t R_t) \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{12}$$

It is clear that (12) is true at $t = 0$, from our choice of $w_0$. Assume now that (12) holds for all $0, \dots, t$, and let us show that it also holds for $t + 1$. To compute $\partial f_{\text{diag}}(w_t)$, observe that

$$\operatorname{sign}(w_{t,1} + w_{t,2}) = \operatorname{sign}(2r) = +1, \qquad \operatorname{sign}(w_{t,1} - w_{t,2}) = \operatorname{sign}(2(\delta + (-1)^t R_t)) = (-1)^t,$$

where the last equality stems from the fact that $|\delta| < R_t$. Using Lemma B.3 we understand that $\partial f_{\text{diag}}(w_t)$ is reduced to a unique subgradient, that we note $g_t$, and whose expression is given by

$$g_t = c \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (-1)^t \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

We can now unroll the recursion defining the momentum sequence $m_t$, and write

$$\begin{aligned} m_t &= (1 - \beta) \sum_{s=0}^{t} \beta^{t-s} g_s \\ &= c(1 - \beta) \sum_{s=0}^{t} \beta^{t-s} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (1 - \beta) \sum_{s=0}^{t} \beta^{t-s} (-1)^s \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ &= c(1 - \beta) \sum_{s=0}^{t} \beta^{s} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (1 - \beta)(-1)^t \sum_{s=0}^{t} (-\beta)^s \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ &= c(1 - \beta) \frac{1 - \beta^{t+1}}{1 - \beta} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (1 - \beta)(-1)^t \frac{1 - (-\beta)^{t+1}}{1 + \beta} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ &= \underbrace{c\big(1 - \beta^{t+1}\big)}_{=: a_t} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (-1)^t \underbrace{\frac{1 - \beta}{1 + \beta}\big(1 - (-\beta)^{t+1}\big)}_{=: b_t} \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \end{aligned}$$

Since $\beta \in [0, 1)$ we know that $a_t \geq 0$. We can also see that $b_t > a_t$ due to the fact that

$$c = \frac{1 - \beta}{2} < \frac{1 - \beta}{1 + \beta}, \qquad 1 - \beta^{t+1} \leq 1 - (-\beta)^{t+1}.$$

This means that the sign of $m_t$ will be governed by the coefficient of the $\left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right]$-term, that is

$$\operatorname{sign}(m_t) = (-1)^t \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Hence

$$\begin{aligned} w_{t+1} &= w_t - \lambda_t \operatorname{sign}(m_t) \\ &= r \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (\delta + (-1)^t R_t) \begin{bmatrix} 1 \\ -1 \end{bmatrix} - \lambda_t (-1)^t \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ &= r \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (\delta + (-1)^t (R_t - \lambda_t)) \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ &= r \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (\delta + (-1)^{t+1} R_{t+1}) \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \end{aligned}$$

where the last step uses $R_{t+1} = \lambda_t - R_t$. This proves that (12) holds for $t+1$.

Now we simply write that

$$f_{\text{diag}}(w_t) - \inf f_{\text{diag}} = c|w_{t,1} + w_{t,2}| + |w_{t,1} - w_{t,2}| \geq c|w_{t,1} + w_{t,2}| = 2cr \geq 2c = 1 - \beta.$$

To conclude the proof, it remains to formally use Lemma B.4 to convert this lower bound into one for (Muon). Consider the function $f(W) = f_{\text{diag}}(\operatorname{diag}_2(W))$, and define the following set of initializations

$$\Omega = \{W \in \mathbb{R}^{m \times n} \mid \operatorname{diag}_2(W) \in \mathcal{W}\}.$$

From what we know about $\mathcal{W}$, we can say that $\Omega$ is nonempty and has a nonempty interior when $\lambda > 0$. Consider now the iterates of (Muon) applied to $f$ and initialized with $W_0 \in \Omega$. Defining $\hat{w}_t := \operatorname{diag}_2(W_t)$, we know by definition that $\hat{w}_0 \in \mathcal{W}$. Using Lemma B.4, we also know that $\hat{w}_t$ follow the (signMomentum) algorithm applied to $f_{\text{diag}}$, with $f_{\text{diag}}(\hat{w}_t) = f(W_t)$. Combining what precedes, we finally obtain that

$$f(W_t) = f_{\text{diag}}(\hat{w}_t) \geq 1 - \beta.$$

It remains to use the Lipschitzness of $f$ to convert this into a bound on the iterates themselves. For every minimizer $W^\star$ of $f$, we can use the fact that $f$ is 2-Lipschitz continuous (see Lemma B.3) to write

$$1 - \beta \leq f(W_t) = |f(W_t) - f(W^\star)| \leq 2\|W_t - W^\star\|_F.$$

To conclude the proof, we simply need to replace $\Omega$ with its interior when $\lambda > 0$ to obtain an open set of initializations. □

Equation (12) places all iterates in the affine hyperplane $\{W \mid W_{11} + W_{22} = 2r\}$, where $r \geq 1$. Every convex combination $\bar{W}_T$ of these iterates lies in the same hyperplane, so $f(\bar{W}_T) \geq 2cr \geq 1 - \beta$. The same Lipschitz argument gives the distance bound in (5).

**Counterexample 3.4** (Muon cannot converge: adaptive stepsize and low momentum)**.** Consider $\beta \in \left[0, \frac{1}{2}\right)$. Let $f$ be as in Definition 3.1 with $c = \frac{1}{2} - \beta$. Consider the iterates $W_t$ of (Muon) or (regMuon) applied to $f$ where the stepsizes $\lambda_t$ are chosen as an offline function of the previous subgradients $G_0, \ldots, G_t$. Then there exists a set of zero measure such that, for any initialization outside of this set, the iterates cannot converge to a minimizer of $f$.

*Proof of Counterexample 3.4.* Let $W_t$ be the iterates of (Muon) applied to $f$. Recall that we assumed the stepsize at each step to be chosen as a function of the previously encountered gradients. That is, there exist functions $\phi_t : (\mathbb{R}^{m \times n})^{t+1} \to \mathbb{R}$ defined offline such that $\lambda_t = \phi_t(G_0, G_1, \ldots, G_t)$. In the light of Lemma B.4, let us define $f_{\text{diag}}(w_1, w_2) = c|w_1 + w_2| + |w_1 - w_2|$ so that $f(W) = f_{\text{diag}}(\operatorname{diag}_2(W))$. Let us also define $w_t := \operatorname{diag}_2(W_t)$ which satisfy $f(W_t) = f_{\text{diag}}(w_t)$ and follow the (signMomentum) algorithm

$$\begin{aligned} m_t &= \beta m_{t-1} + (1 - \beta) g_t, \quad g_t \in \partial f_{\text{diag}}(w_t), \\ w_{t+1} &= w_t - \lambda_t \operatorname{sign}(m_t), \end{aligned}$$

where $G_t = \operatorname{diag}_2^*(g_t)$. In particular we see that $\lambda_t$ can be written as an offline function $\hat{\phi}_t$ of the gradients $g_0, \dots, g_t$.

Our main task is now to provide a lower bound for this (signMomentum) algorithm. A key goal is to show that for almost every initialization, the subgradient $g_t \in \partial f_{\text{diag}}(w_t)$ can only take on finitely many values. From the subdifferential formula (11), we define this finite set of values as

$$\mathcal{G} = \left\{ cs_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + s_2 \begin{bmatrix} 1 \\ -1 \end{bmatrix} \;\middle|\; s_1, s_2 \in \{\pm 1\} \right\}.$$

Define now

$$E_\phi := \bigcup_{T \in \mathbb{N}_0} \left\{ 2 \sum_{t=0}^{T-1} \hat{\phi}_t(h_0, \dots, h_t)\delta_t \;\middle|\; \delta_t \in \{\pm 1\},\ h_t \in \mathcal{G},\ t = 0, \dots, T-1 \right\}, \tag{13}$$

and

$$\hat{\mathcal{W}} := \{w \in \mathbb{R}^2 \mid w_1 + w_2 = 0\} \cup \{w \in \mathbb{R}^2 \mid w_1 - w_2 \in E_\phi\}. \tag{14}$$

The set $E_\phi$ is countable: for each fixed $T$, the displayed set is finite, and $E_\phi$ is a countable union of such finite sets. Hence $E_\phi$ has measure zero. Since $w \mapsto w_1 - w_2$ is a nonzero linear functional,

$$\{w \mid w_1 - w_2 \in E_\phi\} = \bigcup_{x \in E_\phi} \{w \mid w_1 - w_2 = x\}$$

is a countable union of affine hyperplanes, hence has measure zero. Taking the union with $\{w \mid w_1 + w_2 = 0\}$ also gives a measure zero set, thus $\hat{\mathcal{W}}$ has measure zero. We finally define

$$\mathcal{W} := \{W \in \mathbb{R}^{m \times n} \mid \operatorname{diag}_2(W) \in \hat{\mathcal{W}}\}$$

which also has measure zero. Indeed,

$$\mathcal{W} = \{W \mid W_{11} + W_{22} = 0\} \cup \bigcup_{x \in E_\phi} \{W \mid W_{11} - W_{22} = x\},$$

which is a countable union of affine hyperplanes in $\mathbb{R}^{m \times n}$. We will impose from now on that $W_0 \notin \mathcal{W}$. Note that this is equivalent to saying that $w_0 = \operatorname{diag}_2(W_0) \notin \hat{\mathcal{W}}$.

Let

$$p_t := w_{t,1} + w_{t,2} = \left\langle w_t, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\rangle, \qquad q_t := w_{t,1} - w_{t,2} = \left\langle w_t, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\rangle. \tag{15}$$

Our initialization choice $w_0 \notin \hat{\mathcal{W}}$ implies that $p_0 \neq 0$ and $q_0 \notin E_\phi$. Since $0 \in E_\phi$ (take $T = 0$ in (13)), we also have $q_0 \neq 0$.

We show by induction that, for all $t = 0, 1, \dots$,

$$p_t = p_0, \qquad q_t = q_0 - 2 \sum_{s=0}^{t-1} \lambda_s \operatorname{sign}(q_s), \qquad q_t \neq 0. \tag{16}$$

The claim is immediate at $t = 0$. Assume (16) holds at time $t$. By induction we have that $p_t \neq 0$ and $q_t \neq 0$, thus

$$\operatorname{sign}(p_0) \in \{\pm 1\}, \qquad \operatorname{sign}(q_t) \in \{\pm 1\}.$$

Since $p_t \neq 0$ and $q_t \neq 0$, the subgradient (11) is unique, and is given by

$$g_t = c \operatorname{sign}(p_0) \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \operatorname{sign}(q_t) \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Since $m_{-1} = 0$ and $g_t$ is a weighted sum of $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ for all $t$, we have

$$m_t = \alpha_t \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \eta_t \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{17}$$

for some scalars $\alpha_t, \eta_t$. The momentum recursion gives

$$\alpha_t = \beta\alpha_{t-1} + (1-\beta)c\,\mathrm{sign}(p_0), \qquad \eta_t = \beta\eta_{t-1} + (1-\beta)\,\mathrm{sign}(q_t), \qquad \alpha_{-1} = \eta_{-1} = 0.$$

Hence

$$\alpha_t = c(1-\beta^{t+1})\,\mathrm{sign}(p_0) \qquad \Longrightarrow \qquad |\alpha_t| \le c.$$

Also, since $|\,\mathrm{sign}(q_j)| = 1$ for all $j = 0, \dots, t$ and $\eta_{-1} = 0$, we have

$$|\eta_t| = |(1-\beta)\sum_{j=0}^{t}\beta^j\,\mathrm{sign}(q_j)| \;\le\; (1-\beta)\sum_{j=0}^{t}\beta^j = 1-\beta^{t+1} \le 1.$$

Consequently

$$\mathrm{sign}(q_t)\eta_t = (1-\beta) + \beta\,\mathrm{sign}(q_t)\eta_{t-1} \ge (1-\beta) - \beta|\eta_{t-1}| \ge 1-2\beta.$$

Since $1-2\beta > c$, the above gives

$$\mathrm{sign}(q_t)\eta_t > c \ge |\alpha_t| \ge 0.$$

As a consequence of the above we have that

$$\mathrm{sign}(\eta_t) = \mathrm{sign}(q_t), \qquad |\eta_t| > |\alpha_t|. \tag{18}$$

Consequently $m_t$ in (17) is given by

$$m_t = (\alpha_t + \eta_t, \alpha_t - \eta_t).$$

By (18),

$$\mathrm{sign}(\alpha_t + \eta_t) = \mathrm{sign}(q_t), \qquad \mathrm{sign}(\alpha_t - \eta_t) = -\,\mathrm{sign}(q_t).$$

Hence

$$\mathrm{sign}(m_t) = \begin{bmatrix} \mathrm{sign}(q_t) \\ -\,\mathrm{sign}(q_t) \end{bmatrix} = \mathrm{sign}(q_t)\begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

The update becomes

$$w_{t+1} = w_t - \lambda_t\,\mathrm{sign}(q_t)\begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{19}$$

Hence by definition of $p_t$ in (15) we have that

$$p_{t+1} = \left\langle w_{t+1}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\rangle = p_t - \lambda_t\,\mathrm{sign}(q_t)\left\langle \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\rangle \;=\; p_t,$$

which shows the first part of (16) for $t+1$. For $\delta_t = -\,\mathrm{sign}(q_t) \in \{\pm 1\}$, by definition of $q_t$ in (15) we have that

$$\begin{aligned} q_{t+1} &= \left\langle w_{t+1}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\rangle \\ &= q_t - \lambda_t\,\mathrm{sign}(q_t)\left\langle \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\rangle \\ &= q_t - 2\lambda_t\,\mathrm{sign}(q_t), \end{aligned}$$

and unrolling over $t$ yields

$$q_{t+1} = q_0 - 2\sum_{s=0}^{t}\lambda_s\,\mathrm{sign}(q_s),$$

which shows the second part of (16) for $t+1$. For the last part of (16), notice that

$$\begin{aligned} q_{t+1} = 0 \quad &\Longleftrightarrow \quad q_0 = 2\sum_{s=0}^{t}\lambda_s\,\mathrm{sign}(q_s) \\ &\Longleftrightarrow \quad q_0 = 2\sum_{s=0}^{t}\hat{\phi}_s(g_0, \dots, g_s)\,\mathrm{sign}(q_s). \end{aligned}$$

Since $g_s \in \mathcal{G}$ and $\operatorname{sign}(q_s) \in \{\pm 1\}$ for all $s$, it follows that $q_0 \in E_\phi$. By construction $q_0 \notin E_\phi$, hence $q_{t+1} \neq 0$. This proves the claim for $t+1$.

Therefore $p_t = p_0 \neq 0$ for all $t$, and

$$f_{\text{diag}}(w_t) \geq c|w_{t,1} + w_{t,2}| = c|p_t| = c|p_0| = \frac{1-2\beta}{2}|w_{0,1} + w_{0,2}| > 0,$$

which in turn means that

$$f(W_t) - \inf f \geq c|p_0| > 0.$$

So we conclude that the iterates cannot converge to a minimizer.

For (regMuon), set $\tilde{\lambda}_t = \lambda_t \|M_t\|_{\text{nuc}}$. Since $M_t$ is determined by $G_0, \ldots, G_t$, $\tilde{\lambda}_t$ is covered by the same argument. □

### B.5 Non-diagonal counterexamples

**Proposition B.5** (Non-diagonal counterexamples)**.** Let $U \in \mathbb{R}^{m\times 2}$ and $V \in \mathbb{R}^{n\times 2}$ have orthonormal columns. Define

$$K = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}, \qquad A = UKV^\top, \quad B = UHV^\top,$$

and $f(W) = c|\langle A, W\rangle| + |\langle B, W\rangle|$. The conclusions of Counterexamples 3.3 and 3.4 hold for this $f$, with their respective choices of $c$, momentum, and stepsizes. In each case, the nonconvergent trajectories satisfy, for every $t \geq 0$,

$$\langle A, W_t\rangle = \langle A, W_0\rangle \neq 0, \qquad f(W_t) \geq c|\langle A, W_0\rangle| > 0 = \inf f.$$

*Proof.* Repeat the proofs of Counterexamples 3.3 and 3.4 with $p_t = \langle A, W_t\rangle$ and $q_t = \langle B, W_t\rangle$. The momentum coefficient recursions are unchanged, giving $M_t = U(\alpha_t K + \eta_t H)V^\top$ with $|\alpha_t| < |\eta_t|$ and $\operatorname{sign}(\eta_t) = \operatorname{sign}(q_t)$. Since $H \pm K \succ 0$, we obtain

$$\operatorname{polar}(M_t) = \operatorname{sign}(q_t)UV^\top, \qquad p_{t+1} = p_t, \qquad q_{t+1} = q_t - 2\lambda_t \operatorname{sign}(q_t),$$

where the last two identities use $\operatorname{tr} K = 0$ and $\operatorname{tr} H = 2$. Thus the scalar dynamics are unchanged. The initialization arguments also apply, since $W \mapsto (p, q)$ is surjective. The distance bounds follow as before, since $f$ is convex, $\inf f = 0$, and its Frobenius Lipschitz constant is $\sqrt{(c^2+5)/2} < 2$. For (regMuon), use the same rescaling $\lambda_t \mapsto \lambda_t \|M_t\|_{\text{nuc}}$ as in the original proof. □

Generic choices of $U, V$ make both $A$ and $B$ dense. Moreover, $KH \neq HK$, so $A, B$ cannot be simultaneously diagonalized by orthogonal left and right transformations. Each update leaves $\langle A, W_t\rangle$ unchanged. Since this value is nonzero at initialization and zero at every minimizer, the iterates cannot converge to a minimizer.

## C Proofs in Section 4

### C.1 LMOs and compression operators

**Proposition 4.1** (Dual subgradient compression operator)**.** Let $\|\cdot\|$ be any norm on $\mathbb{R}^{m\times n}$ with dual norm $\|\cdot\|_*$. Let $\alpha, \beta > 0$ satisfy for all $W \in \mathbb{R}^{m\times n}$

$$\alpha\|W\|_F \leq \|W\| \leq \beta\|W\|_F. \tag{8}$$

Then the operator $\mathcal{C}(W) := \alpha^2\|W\|_* \mathsf{LMO}_{\|\cdot\|}(W)$ is a compression operator, that is

$$\|W - \mathcal{C}(W)\|_F^2 \leq (1-\delta)\|W\|_F^2, \qquad \delta = \frac{\alpha^2}{\beta^2}. \tag{9}$$

In particular with the operator norm $\|\cdot\|_{\text{op}}$ we can take $\alpha^2 = \delta = 1/r$ with $r = \min\{m, n\}$.

*Proof.* For the proof, let $\hat{\mathcal{C}}(W) = \|W\|_* \mathsf{LMO}(W)$, where we have dropped the $\alpha^2$ re-scaling. Expand the squares

$$\|W - \alpha^2 \hat{\mathcal{C}}(W)\|_F^2 = \|W\|_F^2 + \alpha^4 \|\hat{\mathcal{C}}(W)\|_F^2 - 2\alpha^2 \langle W, \hat{\mathcal{C}}(W)\rangle.$$

By definition of $\mathsf{LMO}$ and the dual norm,

$$\langle W, \mathsf{LMO}(W)\rangle = \|W\|_*,$$

hence

$$\langle W, \hat{\mathcal{C}}(W)\rangle = \|W\|_*^2.$$

Also

$$\|\hat{\mathcal{C}}(W)\|_F = \|W\|_* \|\mathsf{LMO}(W)\|_F,$$

and since $\|\mathsf{LMO}(W)\| \leq 1$ and $\|U\| \geq \alpha \|U\|_F$ for every $U \in \mathbb{R}^{m\times n}$, we have

$$\|\mathsf{LMO}(W)\|_F \leq \frac{1}{\alpha} \|\mathsf{LMO}(W)\| \leq \frac{1}{\alpha}.$$

Therefore,

$$\|\hat{\mathcal{C}}(W)\|_F^2 \leq \frac{1}{\alpha^2} \|W\|_*^2.$$

Substituting into the expansion gives

$$\|W - \alpha^2 \hat{\mathcal{C}}(W)\|_F^2 \leq \|W\|_F^2 + \left(\frac{\alpha^4}{\alpha^2} - 2\alpha^2\right) \|W\|_*^2,$$

which becomes

$$\|W - \alpha^2 \hat{\mathcal{C}}(W)\|_F^2 \leq \|W\|_F^2 - \alpha^2 \|W\|_*^2.$$

Finally, $\|W\| \leq \beta \|W\|_F$ implies by duality that

$$\|W\|_* \geq \frac{1}{\beta} \|W\|_F.$$

Hence

$$\|W - \alpha^2 \hat{\mathcal{C}}(W)\|_F^2 \leq \|W\|_F^2 - \frac{\alpha^2}{\beta^2} \|W\|_F^2 = \left(1 - \frac{\alpha^2}{\beta^2}\right) \|W\|_F^2.$$

This proves that $\mathcal{C}(W) = \alpha^2 \hat{\mathcal{C}}(W)$ is a $\delta = \frac{\alpha^2}{\beta^2}$ compression operator.

To conclude the proof, we provide an explicit computation of the constants $\alpha, \beta$ when considering the operator norm. Consider any $W \in \mathbb{R}^{m\times n}$ and let $r$ be its rank, with of course $r \leq \min\{m, n\}$. Consider $W = U\Sigma V^\top$ its reduced SVD where $\Sigma = \text{Diag}(\sigma) \in \mathbb{R}^{r\times r}$, with $\sigma \in \mathbb{R}^r$. Then

$$\|W\|_F = \|\sigma\|_2, \qquad \|W\|_{\text{op}} = \|\sigma\|_\infty.$$

We can then use classic bounds in $\mathbb{R}^r$ to write

$$\frac{1}{\sqrt{r}} \|\sigma\|_2 \leq \|\sigma\|_\infty \leq \|\sigma\|_2,$$

and the conclusion follows from $\frac{1}{\sqrt{r}} \geq \frac{1}{\sqrt{\min\{m,n\}}}$. □

### C.2 Basic choices of LMO-based compressors and EF-MuonMax

As stated in Proposition 4.1, for any given norm $\|\cdot\|$, the corresponding LMO is a compression operator up to a multiplicative factor. Below we list explicit choices of norms and their corresponding multiplicative factors, whose proof can be found in standard textbooks such as [GL13].

**Remark C.1** (Basic LMO-based compressors)**.** For any of the following norms $\|\cdot\|$, the operator

$$\mathcal{C}(W) = \alpha^2 \|W\|_* \mathsf{LMO}_{\|\cdot\|}(W)$$

is a $\delta$-compression operator in the sense of Proposition 4.1, where

- if $\|\cdot\| = \|\cdot\|_1$ on $\mathbb{R}^d$, then $\|\cdot\|_* = \|\cdot\|_\infty$, $\alpha = 1$ and $\delta = 1/d$.
- if $\|\cdot\| = \|\cdot\|_2$ on $\mathbb{R}^d$, then $\|\cdot\|_* = \|\cdot\|_2$, $\alpha = 1$ and $\delta = 1$.
- if $\|\cdot\| = \|\cdot\|_\infty$ on $\mathbb{R}^d$, then $\|\cdot\|_* = \|\cdot\|_1$, $\alpha = 1/\sqrt{d}$ and $\delta = 1/d$.
- if $\|\cdot\| = \|\cdot\|_p$ on $\mathbb{R}^d$, then $\|\cdot\|_* = \|\cdot\|_q$, $\alpha = d^{\min\left\{0, \frac{1}{p} - \frac{1}{2}\right\}}$, and $\delta = d^{-2\left|\frac{1}{p} - \frac{1}{2}\right|}$.
- if $\|\cdot\| = \|\cdot\|_{\text{nuc}}$ on $\mathbb{R}^{m\times n}$, then $\|\cdot\|_* = \|\cdot\|_{\text{op}}$, $\alpha = 1$, and $\delta = 1/\min\{m, n\}$.
- if $\|\cdot\| = \|\cdot\|_{\text{op}}$ on $\mathbb{R}^{m\times n}$, then $\|\cdot\|_* = \|\cdot\|_{\text{nuc}}$, and $\alpha^2 = \delta = 1/\min\{m, n\}$.

These are worst-case bounds; for structured vectors (e.g., sparse or low rank gradients) the effective compression can be much better.

Computing an element in the $\mathsf{LMO}_{\|\cdot\|}(W)$ can be done in closed form for the norms $\|\cdot\|_1$, $\|\cdot\|_2$, $\|\cdot\|_\infty$, $\|\cdot\|_{\text{nuc}}$, see [Gru+25, Section D.1] for more details. For the operator norm $\|\cdot\|_{\text{op}}$, we prove here that the polar factor belongs to the LMO.

**Lemma C.2.** For any $A \in \mathbb{R}^{m\times n}$, $\text{polar}(A)$ is the unique least Frobenius norm element of $\mathsf{LMO}_{\|\cdot\|_{\text{op}}}(A)$.

*Proof.* By duality of the spectral and nuclear norms [Wat92],

$$\mathsf{LMO}_{\|\cdot\|_{\text{op}}}(A) = \partial \|A\|_{\text{nuc}}.$$

Let $r$ be the rank of $A$, and $A = U\Sigma V^\top$ be full SVD of $A$. Write

$$U = [U_r \quad U_\perp], \qquad V = [V_r \quad V_\perp],$$

where $U_r \Sigma_r V_r^\top$ is the reduced SVD of $A$.

By [Wat92], the subdifferential is equal to

$$\partial \|A\|_{\text{nuc}} = U_r V_r^\top + \{U_\perp E V_\perp^\top \mid E \in \mathbb{R}^{(m-r)\times(n-r)},\ \|E\|_{\text{op}} \le 1\}.$$

Moreover, $\text{polar}(A) = U_r V_r^\top$.

Consider any $E \in \mathbb{R}^{(m-r)\times(n-r)}$ with $\|E\|_{\text{op}} \le 1$, then

$$U_r V_r^\top + U_\perp E V_\perp^\top \in \partial \|A\|_{\text{nuc}}.$$

By orthogonal invariance of Frobenius norm,

$$\|U_r V_r^\top + U_\perp E V_\perp^\top\|_F^2 = \left\| U \begin{bmatrix} I_r & 0 \\ 0 & E \end{bmatrix} V^\top \right\|_F^2 = r + \|E\|_F^2 \ge r = \|\text{polar}(A)\|_F^2.$$

Equality is attained if and only if $E = 0$, which concludes the result. □

We can now specialize Algorithm 1 to the operator-norm compressor on a single matrix, giving the single-matrix version of EF-MuonMax below. The product-space version, which uses a shared scale across layers, is given in Algorithm 4.

**Algorithm 2** EF-MuonMax (single matrix)

**Input:** stepsizes $\lambda_t > 0$, momentum parameter $\beta \in [0, 1)$, $r = \min\{m, n\}$

Initialize $W_0 \in \mathbb{R}^{m\times n}$, $E_0 = 0$, and $M_{-1} = 0$
**for** $t = 0, 1, \ldots, T$ **do**
  $G_t \in \mathbb{R}^{m\times n}$ ▷ Sample a stochastic subgradient
  $M_t = \beta M_{t-1} + (1-\beta)G_t$ ▷ Update the momentum
  $P_t = E_t + \lambda_t M_t$ ▷ Form the error-corrected message
  $W_{t+1} = W_t - \frac{1}{r}\|P_t\|_{\text{nuc}}\,\text{polar}(P_t)$ ▷ Update the iterate with a spectral step
  $E_{t+1} = E_t + W_{t+1} - (W_t - \lambda_t M_t)$ ▷ Update the error feedback memory
**end for**

### C.3 LMO-based compression operator over layers

Neural networks are not parametrized by a single matrix, but instead by a Cartesian product of matrices

$$W := (W^1, \ldots, W^L, \theta) \tag{20}$$

where $W^\ell \in \mathbb{R}^{m_\ell \times n_\ell}$ is the weight matrix for all $\ell = 1, \ldots, L$ layers, and $\theta \in \mathbb{R}^k$ contains all the non-matrix parameters of the entire network.

To define a subgradient method we need to now choose a norm over the product space (20). To have the resulting subgradient method perform Muon-type updates over the weight matrices, we will choose a spectral norm over these parameters. It then remains to aggregate all parameters into one norm and use Proposition 4.1 to define a valid compression operator.

We now choose a norm over this product space so that (regMuon) with respect to this norm recovers a variant of MuonMax [Cra+25; BN24]. MuonMax was the first proposed variant of Muon [BN24]. We choose this variant because in a systematic study of Muon variants, it was found to be best performing regularized variant of Muon [Cra+25]. We define such norm in Lemma C.3 along with corresponding compression operator and LMO. Note that up to a scalar factor, the compression applies the polar factor separately to each layer.

**Lemma C.3** (Product norm)**.** Consider a Cartesian product of matrices as in (20), and for each $\ell = 1, \ldots, L$ consider $d_\ell = \min\{m_\ell, n_\ell\}$. If we define, for some $s > 0$,

$$\|W\| := \left(\left(\max_{\ell\in[L]} \sqrt{\frac{d_\ell}{s}}\|W^\ell\|_{\text{op}}\right)^2 + k\|\theta\|_\infty^2\right)^{1/2}, \tag{21}$$

then $\|\cdot\|$ is a norm on

$$\mathbb{R}^{m_1\times n_1} \times \cdots \times \mathbb{R}^{m_L \times n_L} \times \mathbb{R}^k.$$

Its dual norm is

$$\|W\|_* = \left(sy(W)^2 + \frac{\|\theta\|_1^2}{k}\right)^{1/2},$$

where

$$y(W) = \sum_{\ell=1}^{L} \frac{1}{\sqrt{d_\ell}}\|W^\ell\|_{\text{nuc}}.$$

Moreover, for $W \neq 0$, the least product-Frobenius norm element of the set $\mathsf{LMO}_{\|\cdot\|}(W)$ equals

$$\mathsf{LMO}^{\min}(W) = \frac{1}{\|W\|_*}\left(s\frac{y(W)}{\sqrt{d_1}}\,\text{polar}(W^1), \ldots, s\frac{y(W)}{\sqrt{d_L}}\,\text{polar}(W^L), \frac{\|\theta\|_1}{k}\,\text{sign}(\theta)\right).$$

In addition, the corresponding compressor is

$$\mathcal{C}(W) = \min\{s, 1/L\} \left( \frac{y(W)}{\sqrt{d_1}} \operatorname{polar}(W^1), \ldots, \frac{y(W)}{\sqrt{d_L}} \operatorname{polar}(W^L), \frac{\|\theta\|_1}{sk} \operatorname{sign}(\theta) \right). \quad (22)$$

*Proof.* Let $v = (v_1, \ldots, v_{L+1}) \in \mathbb{R}^{L+1}$ with $v \neq 0$. Define

$$\|v\|_{\text{hyb}} = \left( \|v_{1:L}\|_\infty^2 + v_{L+1}^2 \right)^{1/2}.$$

This is the "hybrid" outer norm introduced in [Cra+25] that satisfies

$$\begin{aligned} \|v\|_{\text{hyb},*} &= \left( \|v_{1:L}\|_1^2 + v_{L+1}^2 \right)^{1/2} \\ \mathsf{LMO}_{\text{hyb}}(v) &= \frac{1}{\|v\|_{\text{hyb},*}} \left( \|v_{1:L}\|_1 \operatorname{SIGN}(v_1), \ldots, \|v_{1:L}\|_1 \operatorname{SIGN}(v_L), v_{L+1} \right). \end{aligned}$$

Now, set for all $\ell = 1, \ldots, L$

$$g_\ell(X) := \sqrt{\frac{d_\ell}{s}} \|X\|_{\text{op}}, \qquad g_{L+1}(z) := \sqrt{k} \|z\|_\infty, \qquad f(r) := \|r\|_{\text{hyb}}.$$

Then

$$\|W\| = f\big(g_1(W^1), \ldots, g_L(W^L), g_{L+1}(\theta)\big),$$

so the claim follows from [Cra+25, Lemma 3.3] after translating the argmin convention to our argmax convention.

Indeed, since $f^* = \|\cdot\|_{\text{hyb},*}$,

$$g_{\ell,*}(X) = \frac{\|X\|_{\text{nuc}}}{\sqrt{d_\ell/s}}, \qquad g_{L+1,*}(z) = \frac{\|z\|_1}{\sqrt{k}},$$

and therefore

$$\|W\|_* = \left( s y(W)^2 + \frac{\|\theta\|_1^2}{k} \right)^{1/2},$$

where

$$y(W) = \sum_{\ell=1}^{L} \frac{1}{\sqrt{d_\ell}} \|W^\ell\|_{\text{nuc}}.$$

Then the full LMO set is

$$\begin{aligned} \mathsf{LMO}(W) &= \left\{ (\phi_1 U^1, \ldots, \phi_L U^L, \phi_{L+1} u^\theta) \mid \phi \in \Phi,\ u = (U^1, \ldots, U^L, u^\theta) \in \mathcal{U} \right\} \\ \Phi &= \mathsf{LMO}_{\text{hyb}} \left( \frac{\|W^1\|_{\text{nuc}}}{\sqrt{d_1/s}}, \ldots, \frac{\|W^L\|_{\text{nuc}}}{\sqrt{d_L/s}}, \frac{\|\theta\|_1}{\sqrt{k}} \right) \\ \mathcal{U} &= \left( \mathsf{LMO}_{g_1}(W^1), \ldots, \mathsf{LMO}_{g_L}(W^L), \mathsf{LMO}_{g_{L+1}}(\theta) \right). \end{aligned}$$

Moreover,

$$\mathsf{LMO}_{g_\ell}(X) = \sqrt{\frac{s}{d_\ell}} \partial \|X\|_{\text{nuc}}, \qquad \mathsf{LMO}_{g_{L+1}}(z) = \frac{1}{\sqrt{k}} \operatorname{SIGN}(z),$$

and using $\sqrt{d_\ell/s} > 0$,

$$\Phi = \frac{1}{\|W\|_*} \left( \sqrt{s} y(W) \operatorname{SIGN}\left( \|W^1\|_{\text{nuc}} \right), \ldots, \sqrt{s} y(W) \operatorname{SIGN}\left( \|W^L\|_{\text{nuc}} \right), \frac{\|\theta\|_1}{\sqrt{k}} \right).$$

We now show that least product-Frobenius norm element of LMO is

$$\mathsf{LMO}^{\min}(W) = \frac{1}{\|W\|_*} \left( s \frac{y(W)}{\sqrt{d_1}} \operatorname{polar}(W^1), \ldots, s \frac{y(W)}{\sqrt{d_L}} \operatorname{polar}(W^L), \frac{\|\theta\|_1}{k} \operatorname{sign}(\theta) \right). \quad (23)$$

The nonnegative block scalings for blocks with $W_\ell \neq 0$ and for vector block are

$$a_\ell = \frac{sy(W)}{\sqrt{d_\ell}\|W\|_*}, \qquad b = \frac{\|\theta\|_1}{k\|W\|_*}.$$

If $W^\ell = 0$, the block $\ell$ does not affect $\|W\|_*$ and $y(W)$, and the least Frobenius norm choice is 0. Indeed, after the outer hybrid LMO fixes the nonnegative block scalings the inner maximization separates over blocks. For each matrix block, the least Frobenius norm maximizer of

$$\max_{\|Z\|_{\mathrm{op}} \le a_\ell} \langle W^\ell, Z \rangle$$

is $a_\ell \operatorname{polar}(W^\ell)$ by Lemma C.2. For the vector block, the least Euclidean norm maximizer of

$$\max_{\|z\|_\infty \le b} \langle \theta, z \rangle$$

is $b \operatorname{sign}(\theta)$. Therefore, (23) gives the least Frobenius norm element of the LMO set.

Finally,

$$\begin{aligned}
\|W\|^2 &= \left(\max_{\ell\in[L]} \sqrt{\frac{d_\ell}{s}}\|W^\ell\|_{\mathrm{op}}\right)^2 + k\|\theta\|_\infty^2 \\
&\ge \left(\max_{\ell\in[L]} \frac{1}{\sqrt{s}}\|W^\ell\|_F\right)^2 + \|\theta\|_F^2 \\
&= \frac{1}{s}\left(\max_{\ell\in[L]} \|W^\ell\|_F\right)^2 + \|\theta\|_F^2 \\
&\ge \frac{1}{sL}\left(\sum_{\ell\in[L]} \|W^\ell\|_F^2\right) + \|\theta\|_F^2 \\
&\ge \min\{1, 1/sL\}\left(\sum_{\ell\in[L]} \|W^\ell\|_F^2 + \|\theta\|_F^2\right) \\
&= \min\{1, 1/sL\}\|W\|_F^2.
\end{aligned}$$

Hence $\alpha = \min\{1, 1/\sqrt{sL}\}$ in Proposition 4.1, so

$$\mathcal{C}(W) = \min\{s, 1/L\}\left(\frac{y(W)}{\sqrt{d_1}}\operatorname{polar}(W^1), \ldots, \frac{y(W)}{\sqrt{d_L}}\operatorname{polar}(W^L), \frac{\|\theta\|_1}{sk}\operatorname{sign}(\theta)\right).$$

□

### C.4 MuonMax

In this section we define MuonMax in Algorithm 3, which is a variant of (regMuon) with respect to the product norm in Lemma C.3.

We also define the product-space version of EF-MuonMax in Algorithm 4. It extends the single-matrix construction in Algorithm 2 to multiple matrix layers and non-matrix parameters, using a shared scale $y(P_t)$ across the matrix layers. This algorithm combines error feedback with Algorithm 3, by specializing Algorithm 1 to compressor in Lemma C.3. Thus by Theorem 4.2 iterates of Algorithm 4 converge to a minimizer.

With a slight abuse of notation, we use the name MuonMax also for variants in which the signed momentum paired optimizer is replaced by another paired optimizer.

### C.5 Convergence bounds for error feedback with momentum

The convergence result from the main body Theorem 4.2 is a direct corollary of Theorem C.4 below.

**Algorithm 3** MuonMax (product space)

**Input:** stepsize schedule $\{\lambda_t\}_{t=0}^{\infty}$, momentum parameter $\beta \in [0,1)$, scaling parameter $s > 0$

Initialize $W_0 \in \mathbb{R}^{m_1 \times n_1} \times \cdots \times \mathbb{R}^{m_L \times n_L} \times \mathbb{R}^k$
Set $M_{-1} = 0$
**for** $t = 0, 1, \ldots, T$ **do**
  $G_t = (G_t^1, \ldots, G_t^L, g_t^\theta)$ ▷ Sample a stochastic subgradient
  $M_t = \beta M_{t-1} + (1-\beta) G_t$ ▷ Update the momentum
  $y(M_t) = \sum_{\ell=1}^{L} \frac{1}{\sqrt{d_\ell}} \|M_t^\ell\|_{\mathrm{nuc}}$
  $\mathsf{LMO}(M_t) = \frac{1}{\|M_t\|_*} \left( s \frac{y(M_t)}{\sqrt{d_1}} \operatorname{polar}(M_t^1), \ldots, s \frac{y(M_t)}{\sqrt{d_L}} \operatorname{polar}(M_t^L), \frac{\|m_t^\theta\|_1}{k} \operatorname{sign}(m_t^\theta) \right)$ ▷ LMO
  $W_{t+1} = W_t - \lambda_t \|M_t\|_* \mathsf{LMO}(M_t)$ ▷ Update the iterate
**end for**
**return** $W_T$

**Algorithm 4** EF-MuonMax (product space)

**Input:** stepsize $\lambda > 0$, momentum parameter $\beta \in [0,1)$, scaling parameter $s > 0$

Initialize $W_0 \in \mathbb{R}^{m_1 \times n_1} \times \cdots \times \mathbb{R}^{m_L \times n_L} \times \mathbb{R}^k$
Set $E_0 = 0$, and set $M_{-1} = 0$
**for** $t = 0, 1, \ldots, T$ **do**
  $G_t = (G_t^1, \ldots, G_t^L, g_t^\theta)$ ▷ Sample a stochastic subgradient
  $M_t = \beta M_{t-1} + (1-\beta) G_t$ ▷ Update the momentum
  $P_t = E_t + \lambda M_t$ ▷ Form the error-corrected message
  $y(P_t) = \sum_{\ell=1}^{L} \frac{1}{\sqrt{d_\ell}} \|P_t^\ell\|_{\mathrm{nuc}}$
  $\mathcal{C}(P_t) = \min\{s, \frac{1}{L}\} \left( \frac{y(P_t)}{\sqrt{d_1}} \operatorname{polar}(P_t^1), \ldots, \frac{y(P_t)}{\sqrt{d_L}} \operatorname{polar}(P_t^L), \frac{\|p_t^\theta\|_1}{sk} \operatorname{sign}(p_t^\theta) \right)$ ▷ Compressor
  $W_{t+1} = W_t - \mathcal{C}(P_t)$ ▷ Update the iterate
  $E_{t+1} = P_t - \mathcal{C}(P_t)$ ▷ Update the error feedback memory
**end for**
**return** $W_T$

**Theorem C.4.** Consider Algorithm 1. Suppose that $f : \mathbb{R}^{m \times n} \to \mathbb{R}$ is convex and admits a minimizer $W^\star$. At iteration $t$ let $G_t$ be a stochastic subgradient such that

$$\mathbb{E}_t[G_t] \in \partial f(W_t), \qquad \mathbb{E}_t \|G_t\|_F^2 \le \sigma^2.$$

Let $\mathcal{C}$ be a $\delta$-approximate compressor, meaning that for all $W \in \mathbb{R}^{m \times n}$,

$$\|\mathcal{C}(W) - W\|_F^2 \le (1-\delta)\|W\|_F^2, \qquad \delta \in (0,1].$$

Then, for the averaged iterate $\bar{W}_T = \frac{1}{T+1} \sum_{t=0}^{T} W_t$ and non-increasing stepsizes $\lambda_t$, we have

$$\mathbb{E}\left[f(\bar{W}_T)\right] - f(W^\star) \le \frac{\|W_0 - W^\star\|_F^2}{2\lambda_T(T+1)} + \sigma^2 \left( \frac{2\sqrt{1-\delta}}{\delta} + \frac{\beta}{1-\beta} + \frac{1}{2} \right) \frac{\sum_{t=0}^{T} \lambda_t^2}{\lambda_T (T+1)}.$$

*Proof of Theorem C.4.* Define

$$\widetilde{W}_t := W_t - E_t, \qquad \Delta_t := \mathcal{C}(P_t).$$

Using the updates of Algorithm 1, we have

$$\begin{aligned} \widetilde{W}_{t+1} &= W_{t+1} - E_{t+1} \\ &= \left(W_t - \Delta_t\right) - \left(P_t - \Delta_t\right) \\ &= W_t - P_t \\ &= W_t - (E_t + \lambda_t M_t) \\ &= \widetilde{W}_t - \lambda_t M_t. \end{aligned}$$

Now set

$$a := \frac{\beta}{1-\beta}, \qquad b_{T+1} := a\lambda_T, \qquad b_t := \beta(\lambda_t + b_{t+1}) \qquad \eta_t := (1-\beta)(\lambda_t + b_{t+1}),$$

and

$$Y_t := \widetilde{W}_t - b_t M_{t-1}.$$

Then

$$\begin{aligned} Y_{t+1} &= \widetilde{W}_{t+1} - b_{t+1} M_t \\ &= \widetilde{W}_t - (\lambda_t + b_{t+1}) M_t \\ &= \widetilde{W}_t - (\lambda_t + b_{t+1})\big(\beta M_{t-1} + (1-\beta) G_t\big) \\ &= \widetilde{W}_t - b_t M_{t-1} - \eta_t G_t \\ &= Y_t - \eta_t G_t. \end{aligned}$$

Next, unrolling the recursion for $b_t$, and using $b_{T+1} = a\lambda_T$, gives

$$b_t = \sum_{j=t}^{T} \beta^{j-t+1} \lambda_j + \beta^{T-t+1} a \lambda_T.$$

Since

$$\sum_{j=t}^{T} \beta^{j-t+1} + a\beta^{T-t+1} = \sum_{k=1}^{T-t+1} \beta^k + a\beta^{T-t+1} = a(1-\beta^{T-t+1}) + \beta^{T-t+1} a = a,$$

and $\lambda_j \le \lambda_t$ for all $j \ge t$, we have $b_t \le a\lambda_t$. Finally,

$$\eta_t = (1-\beta)(\lambda_t + b_{t+1}) = (1-\beta) \sum_{j=t}^{T} \beta^{j-t} \lambda_j + \beta^{T-t+1} \lambda_T.$$

The coefficients are nonnegative and sum to

$$(1-\beta) \sum_{j=t}^{T} \beta^{j-t} + \beta^{T-t+1} = 1.$$

Thus $\eta_t$ is a convex combination of $\lambda_t, \lambda_{t+1}, \dots, \lambda_T$. Since $\lambda_t$ is nonincreasing, $\lambda_T \le \eta_t \le \lambda_t$.

Let

$$S_t := \mathbb{E}_t[G_t] \in \partial f(W_t).$$

From the recursion $Y_{t+1} = Y_t - \eta_t G_t$, we obtain

$$\begin{aligned} \mathbb{E}_t \|Y_{t+1} - W^\star\|_F^2 &= \mathbb{E}_t \|Y_t - W^\star - \eta_t G_t\|_F^2 \\ &= \|Y_t - W^\star\|_F^2 - 2\eta_t \langle \mathbb{E}_t[G_t], Y_t - W^\star \rangle + \eta_t^2 \mathbb{E}_t \|G_t\|_F^2 \\ &\le \|Y_t - W^\star\|_F^2 - 2\eta_t \langle S_t, Y_t - W^\star \rangle + \eta_t^2 \sigma^2. \end{aligned}$$

Since

$$Y_t = \widetilde{W}_t - b_t M_{t-1} = W_t - E_t - b_t M_{t-1},$$

we have

$$\langle S_t, Y_t - W^\star \rangle = \langle S_t, W_t - W^\star \rangle - \langle S_t, E_t \rangle - b_t \langle S_t, M_{t-1} \rangle.$$

Substituting this identity into the previous inequality and taking full expectation gives

$$\begin{aligned} \mathbb{E}\|Y_{t+1} - W^\star\|_F^2 \le \mathbb{E}\|Y_t - W^\star\|_F^2 &- 2\eta_t \mathbb{E}\langle S_t, W_t - W^\star \rangle + 2\eta_t \mathbb{E}\langle S_t, E_t \rangle \\ &+ 2b_t \eta_t \mathbb{E}\langle S_t, M_{t-1} \rangle + \eta_t^2 \sigma^2. \end{aligned}$$

Rearranging and summing over $t = 0, \dots, T$ gives

$$\begin{aligned} 2\sum_{t=0}^{T} \eta_t \mathbb{E}\langle S_t, W_t - W^\star \rangle \le \mathbb{E}\|Y_0 - W^\star\|_F^2 &+ 2\sum_{t=0}^{T} \eta_t \mathbb{E}\langle S_t, E_t \rangle \\ &+ 2\sum_{t=0}^{T} b_t \eta_t \mathbb{E}\langle S_t, M_{t-1} \rangle + \sigma^2 \sum_{t=0}^{T} \eta_t^2. \end{aligned} \tag{24}$$

We can bound each of the terms on the RHS of (24) as follows.

First, we bound $\mathbb{E}\|M_t\|_F^2$. By unfolding the recursion

$$M_t = \beta M_{t-1} + (1-\beta)G_t,$$

and using $M_{-1} = 0$, we obtain

$$M_t = (1-\beta)\sum_{i=0}^{t} \beta^{t-i} G_i.$$

The coefficients $(1-\beta)\beta^{t-i}$ are nonnegative and sum to

$$(1-\beta)\sum_{i=0}^{t} \beta^{t-i} = (1-\beta)\sum_{j=0}^{t} \beta^{j} = 1 - \beta^{t+1} \le 1.$$

Hence $M_t$ is a convex combination of $G_0, \dots, G_t$ and the origin. Since $V \mapsto \|V\|_F^2$ is convex,

$$\|M_t\|_F^2 \le (1-\beta)\sum_{i=0}^{t} \beta^{t-i}\|G_i\|_F^2.$$

Taking expectations and using $\mathbb{E}\|G_i\|_F^2 \le \sigma^2$, we get

$$\mathbb{E}\|M_t\|_F^2 \le (1-\beta)\sum_{i=0}^{t} \beta^{t-i}\sigma^2 \le \sigma^2.$$

Next, recall that

$$E_{t+1} = P_t - \mathcal{C}(P_t), \qquad P_t = E_t + \lambda_t M_t.$$

The compressor assumption implies

$$\|E_{t+1}\|_F^2 = \|P_t - \mathcal{C}(P_t)\|_F^2 \le (1-\delta)\|P_t\|_F^2 = (1-\delta)\|E_t + \lambda_t M_t\|_F^2.$$

If $\delta = 1$, then

$$E_{t+1} = P_t - \mathcal{C}(P_t) = 0, \qquad t = 0, 1, \dots.$$

In this case, all terms involving $E_t$ vanish, and the remainder of the proof goes through unchanged. Thus it remains only to consider the case $0 < \delta < 1$. Assume now that $0 < \delta < 1$. Using

$$\|U+V\|_F^2 \le (1+\alpha)\|U\|_F^2 + (1+1/\alpha)\|V\|_F^2$$

with

$$\alpha = \frac{\delta}{2(1-\delta)},$$

we get

$$\|E_t + \lambda_t M_t\|_F^2 \le \frac{2-\delta}{2(1-\delta)}\|E_t\|_F^2 + \frac{2-\delta}{\delta}\lambda_t^2\|M_t\|_F^2.$$

Multiplying by $(1-\delta)$ yields

$$\|E_{t+1}\|_F^2 \le \Big(1 - \frac{\delta}{2}\Big)\|E_t\|_F^2 + \frac{2(1-\delta)}{\delta}\lambda_t^2\|M_t\|_F^2.$$

Taking expectations and using $\mathbb{E}\|M_t\|_F^2 \le \sigma^2$,

$$\mathbb{E}\|E_{t+1}\|_F^2 \le \Big(1 - \frac{\delta}{2}\Big)\mathbb{E}\|E_t\|_F^2 + \frac{2(1-\delta)}{\delta}\lambda_t^2\sigma^2.$$

Since $E_0 = 0$, summing this inequality and multiplying by $2/\delta$ yields

$$\sum_{t=0}^{T} \mathbb{E}\|E_t\|_F^2 \le \frac{4(1-\delta)}{\delta^2}\sigma^2 \sum_{t=0}^{T} \lambda_t^2.$$

By Cauchy–Schwarz and $\eta_t \le \lambda_t$,

$$\begin{aligned}
2\sum_{t=0}^{T} \eta_t \mathbb{E}\langle S_t, E_t\rangle &\le 2\sum_{t=0}^{T} \eta_t \sqrt{\mathbb{E}\|S_t\|_F^2}\sqrt{\mathbb{E}\|E_t\|_F^2} \\
&\le 2\sigma \sum_{t=0}^{T} \eta_t \sqrt{\mathbb{E}\|E_t\|_F^2} \\
&\le 2\sigma \left(\sum_{t=0}^{T} \eta_t^2\right)^{1/2} \left(\sum_{t=0}^{T} \mathbb{E}\|E_t\|_F^2\right)^{1/2} \\
&\le 2\sigma \left(\sum_{t=0}^{T} \lambda_t^2\right)^{1/2} \left(\frac{4(1-\delta)}{\delta^2}\sigma^2 \sum_{t=0}^{T} \lambda_t^2\right)^{1/2} \\
&= 4\sigma^2 \frac{\sqrt{1-\delta}}{\delta} \sum_{t=0}^{T} \lambda_t^2.
\end{aligned}$$

Similarly,

$$\begin{aligned}
2b_t\eta_t \mathbb{E}\langle S_t, M_{t-1}\rangle &\le 2b_t\eta_t \mathbb{E}\left[\|S_t\|_F \|M_{t-1}\|_F\right] \\
&\le 2b_t\eta_t \sqrt{\mathbb{E}\|S_t\|_F^2}\sqrt{\mathbb{E}\|M_{t-1}\|_F^2} \\
&\le 2b_t\eta_t \sigma^2 \\
&\le 2a\sigma^2 \lambda_t^2,
\end{aligned}$$

where the last inequality uses $b_t \le a\lambda_t$ and $\eta_t \le \lambda_t$.

Applying the two bounds above to (24),

$$\begin{aligned}
2\sum_{t=0}^{T} \eta_t \mathbb{E}\langle S_t, W_t - W^\star\rangle &\le \mathbb{E}\|Y_0 - W^\star\|_F^2 + 4\sigma^2 \frac{\sqrt{1-\delta}}{\delta} \sum_{t=0}^{T} \lambda_t^2 + 2a\sigma^2 \sum_{t=0}^{T} \lambda_t^2 + \sigma^2 \sum_{t=0}^{T} \eta_t^2 \\
&\le \mathbb{E}\|Y_0 - W^\star\|_F^2 + \sigma^2 \left(\frac{4\sqrt{1-\delta}}{\delta} + 2a + 1\right) \sum_{t=0}^{T} \lambda_t^2.
\end{aligned}$$

Using $Y_0 = W_0$ and $f(W_t) - f(W^\star) \le \langle S_t, W_t - W^\star\rangle$ from convexity of $f$, this means

$$\sum_{t=0}^{T} \eta_t \mathbb{E}\left[f(W_t) - f(W^\star)\right] \le \frac{\|W_0 - W^\star\|_F^2}{2} + \sigma^2 \left(\frac{2\sqrt{1-\delta}}{\delta} + a + \frac{1}{2}\right) \sum_{t=0}^{T} \lambda_t^2.$$

Since $\eta_t \ge \lambda_T$ and $f(W_t) - f(W^\star) \ge 0$, dividing both sides by $\lambda_T(T+1)$ yields

$$\frac{1}{T+1} \sum_{t=0}^{T} \mathbb{E}\left[f(W_t) - f(W^\star)\right] \le \frac{\|W_0 - W^\star\|_F^2}{2\lambda_T(T+1)} + \sigma^2 \left(\frac{2\sqrt{1-\delta}}{\delta} + a + \frac{1}{2}\right) \frac{\sum_{t=0}^{T} \lambda_t^2}{\lambda_T(T+1)},$$

so by convexity of $f$,

$$\mathbb{E}\left[f(\bar{W}_T)\right] - f(W^\star) \le \frac{\|W_0 - W^\star\|_F^2}{2\lambda_T(T+1)} + \sigma^2 \left(\frac{2\sqrt{1-\delta}}{\delta} + a + \frac{1}{2}\right) \frac{\sum_{t=0}^{T} \lambda_t^2}{\lambda_T(T+1)},$$

which proves the claimed convergence rate. □

We can now prove Theorem 4.2 by plugging in a stepsize choice to Theorem C.4.

**Theorem 4.2** (Anytime convex Lipschitz convergence of EF-M)**.** Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be convex and let $W^\star$ be a minimizer. Let $\mathcal{C}$ be a compression operator, meaning that for all $W \in \mathbb{R}^{m\times n}$,

$$\|\mathcal{C}(W) - W\|_F^2 \leq (1-\delta)\|W\|_F^2, \qquad \delta \in (0,1].$$

Consider the iterates of Algorithm 1, and assume that $G_t$ is a stochastic subgradient satisfying

$$\mathbb{E}_t[G_t] \in \partial f(W_t), \qquad \mathbb{E}_t\|G_t\|_F^2 \leq \sigma^2.$$

If we consider a vanishing schedule for the stepsizes $\lambda_t = 1/\sqrt{t+1}$, then

$$\mathbb{E}\left[f(\bar{W}_T)\right] - f(W^\star) \leq \frac{\|W_0 - W^\star\|_F^2}{2\sqrt{T+1}} + \sigma^2\left(\frac{2\sqrt{1-\delta}}{\delta} + \frac{\beta}{1-\beta} + \frac{1}{2}\right)\frac{1+\log(T+1)}{\sqrt{T+1}},$$

where $\bar{W}_T = \frac{1}{T+1}\sum_{t=0}^T W_t$.

*Proof of Theorem 4.2.* Recall that Theorem C.4 implies for any non-increasing stepsizes $\lambda_t$ that

$$\mathbb{E}\left[f(\bar{W}_T)\right] - f(W^\star) \leq \frac{\|W_0 - W^\star\|_F^2}{2\lambda_T(T+1)} + \sigma^2\left(\frac{2\sqrt{1-\delta}}{\delta} + \frac{\beta}{1-\beta} + \frac{1}{2}\right)\frac{\sum_{t=0}^T \lambda_t^2}{\lambda_T(T+1)},$$

For the particular stepsize choice $\lambda_t = 1/\sqrt{t+1}$, we have

$$\sum_{t=0}^T \lambda_t^2 = \sum_{t=1}^{T+1}\frac{1}{t} \leq 1 + \log(T+1),$$

so

$$\mathbb{E}\left[f(\bar{W}_T)\right] - f(W^\star) \leq \frac{\|W_0 - W^\star\|_F^2}{2\sqrt{T+1}} + \sigma^2\left(\frac{2\sqrt{1-\delta}}{\delta} + \frac{\beta}{1-\beta} + \frac{1}{2}\right)\frac{1+\log(T+1)}{\sqrt{T+1}}.$$

□

**Background.** Error feedback has been considered with Muon [Gru+25] but only in the smooth and distributed setting, and where the compression is done in addition to the matrix sign operation. That is, the matrix sign itself is not interpreted as a compression.

## D Convergence, complexity and limitations that depend on $\beta$

Our work shows that Muon does not converge on a certain class of functions. But there may still exist a *complexity* result. Here we will clarify the difference, and properly define these notions.

By convergence, what we mean is an any time convergence result [GG23]. In slightly informal terms, we say that a sequence of iterates $W_t$ (or some appropriate average of iterates) converges anytime if the suboptimality converges to zero, that is $\lim_{t\to\infty} f(W_t) - \inf f = 0$. Our results in Counterexamples 3.4 and 3.3 show that this is not possible, because for fixed $\beta$ the suboptimality remains bounded away from zero for *all time*.

But sometimes it is enough to show that the suboptimality can be small, but not necessarily converge to zero. This is where complexity results matter. For instance, a typical complexity would be as follows: given $T$ number of iterations, by setting the hyperparameters depending on $T$, show that $f(W_T) - \inf f \leq \frac{1}{T}$. Our counterexamples do not rule out such a result because given $T$ we could choose $\beta = 1 - \frac{1}{2T}$ and our lower bound (5) from Counterexample 3.3 states that

$$f(W_T) - \inf f \geq \frac{1}{2T},$$

which does not rule out the possibility that $f(W_T) - \inf f \leq \frac{1}{T}$. As an illustration of such complexity result, in the stochastic smooth setting, $O(T^{-1/4})$ stationarity complexity results have been

established for Muon by choosing $\beta = 1 - O(1/\sqrt{T})$ [She+25, Theorem 4.1]. But counter to this point, we know that when $\beta = 1$, Muon (Muon) will halt since the momentum buffer will become stationary $M_t = M_{-1}$, thus also ruling out convergence.

This brings us to another one of our stated limitations. All counterexamples rely on a fixed momentum $\beta \in [0, 1)$ coefficient. As such, there could exist a convergence result where, for instance $\beta_t = 1 - \frac{1}{t}$. Our attempts at establishing such a proof with such a scheduled momentum coefficient have failed, so the question remains open whether convergence can be achieved with such a schedule.

## E Experiments on counterexample function

In this section we run Muon algorithm (Muon), and Algorithm 1 with the function from Definition 3.1, see Figure 3.

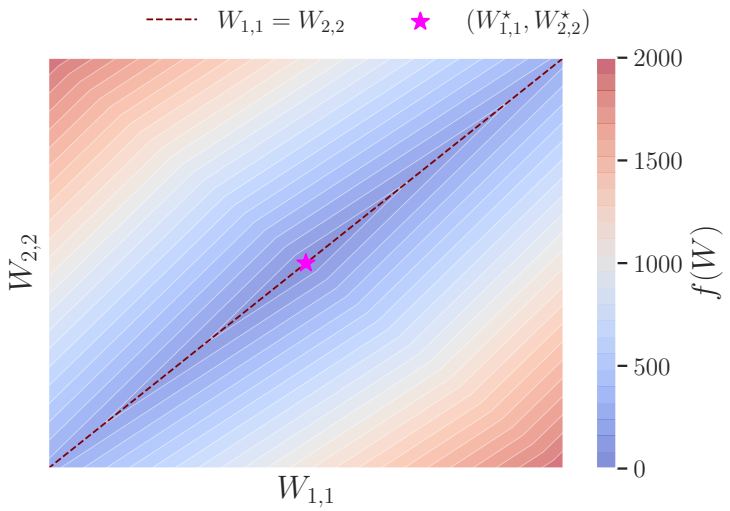


Figure 3: Level sets of $f$ from Definition 3.1.

**Problem instance.** Consider minimizing function from Definition 3.1 with

$$c = \frac{1-\beta}{2(1+\beta)}, \qquad \beta = 0.9.$$

We choose the initial point satisfying

$$(W_0)_{1,1} = 1 + \log 2, \qquad (W_0)_{2,2} = 1 - \log 2,$$

so that the conditions in Counterexample 3.3 hold. We set learning rate in (Muon) as $\lambda_t = \frac{1}{t+1}$, and $\lambda_t = \frac{1}{\sqrt{t+1}}$ in Algorithm 1 for all $t = 0, \ldots, T-1$. We run the methods for $T = 5000$ iterations.

**Muon algorithm.** Counterexample 3.3 shows that, for every $\beta \in [0, 1)$, there exists an initialization $W_0$ for which Muon algorithm fails to converge whenever

$$c \in \left(0, \frac{1-\beta}{1+\beta}\right).$$

Figure 4 illustrates this behavior for the choices above. The iterates satisfy the invariant

$$(W_t)_{1,1} + (W_t)_{2,2} = 2,$$

so they cycle along a line in the $((W_t)_{1,1}, (W_t)_{2,2})$ plane. The resulting trajectory is shown in Figure 1.

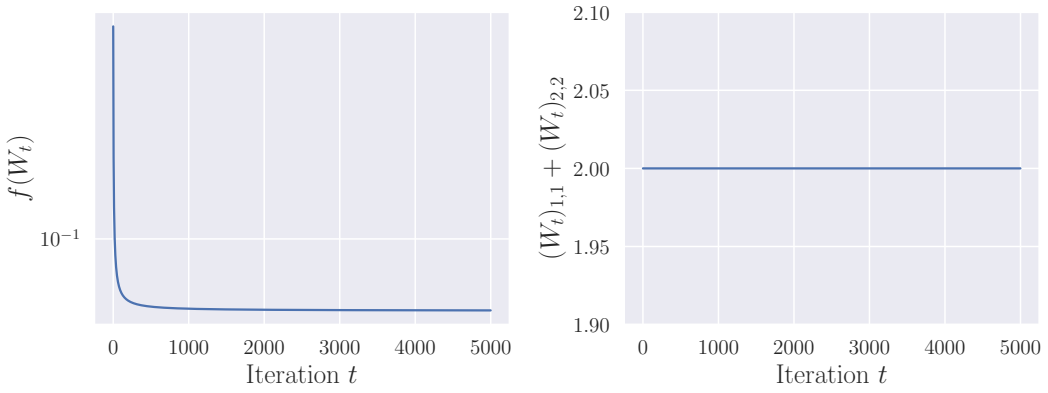


Figure 4: Muon algorithm (Muon) on the counterexample: (left) objective value $f(W_t)$; (right) invariant $(W_t)_{1,1} + (W_t)_{2,2}$.

**EF-M algorithm.** Consider Algorithm 1 with learning rate $\lambda_t = \frac{1}{\sqrt{t+1}}$. In contrast to Muon method, Theorem 4.2 guarantees that Algorithm 1 converges for all convex Lipschitz functions. Figure 5 illustrates that $f(W_t) \to f^\star$, and the right of Figure 1 shows that the iterates converge, $((W_t)_{1,1}, (W_t)_{2,2}) \to (0, 0)$.

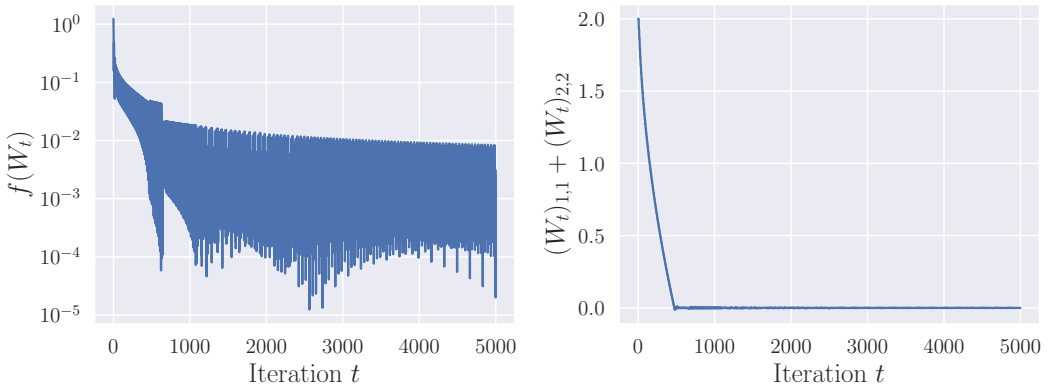


Figure 5: Algorithm 1 on the counterexample: (left) objective value $f(W_t)$; (right) invariant $(W_t)_{1,1} + (W_t)_{2,2}$.

**Error feedback with spectral descent.** We consider the case without momentum ($\beta = 0$). Similarly to the above, Theorem 4.2 guarantees that Algorithm 1 converges for all convex Lipschitz functions. Indeed, Figure 6 shows that $f(W_t) \to f^\star = 0$, and resulting trajectory in Figure 7 shows that $((W_t)_{1,1}, (W_t)_{2,2}) \to (0, 0) = ((W^\star)_{1,1}, (W^\star)_{2,2})$.

## F Deep learning experiments

We compare MuonMax (Algorithm 3), EF-MuonMax (Algorithm 4), Muon, and EF-Muon on image classification and language modeling tasks. This section complements Section 5.

**Parameter partition.** We split all parameters into two groups: matrix parameters and paired parameters. Matrix parameters are weight matrices, excluding embeddings. These are updated by MuonMax, EF-MuonMax, Muon, or EF-Muon. Paired parameters are embeddings, biases, and the remaining one-dimensional parameters. The paired optimizer depends on the method and task.

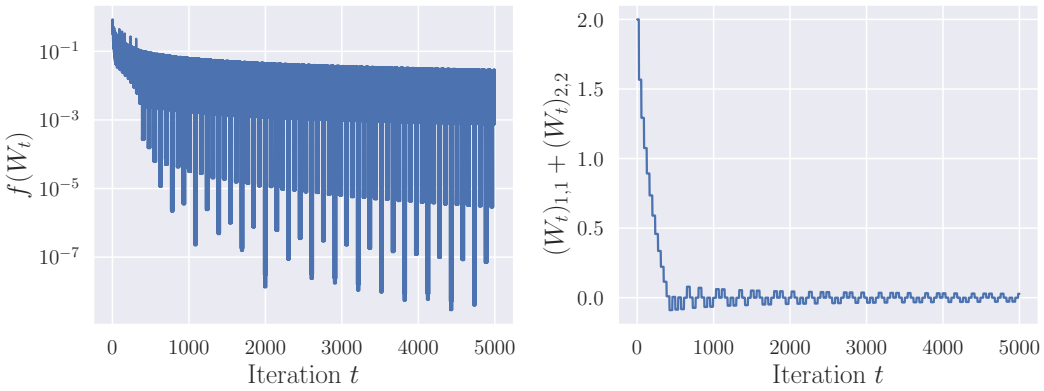


Figure 6: Algorithm 1 without momentum ($\beta = 0$) with $\lambda_t = \frac{1}{\sqrt{t+1}}$ on the counterexample: (left) objective value $f(W_t)$; (right) $(W_t)_{1,1} + (W_t)_{2,2}$.

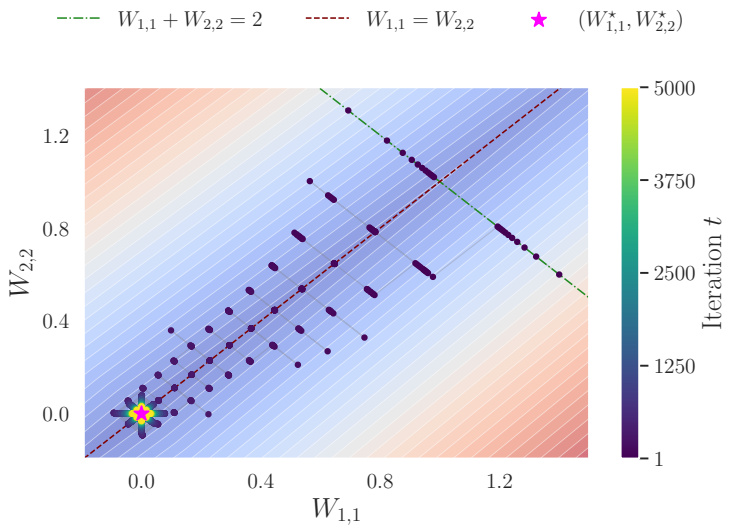


Figure 7: Algorithm 1 without momentum ($\beta = 0$) with $\lambda_t = \frac{1}{\sqrt{t+1}}$ trajectory in the $((W_t)_{1,1}, (W_t)_{2,2})$ plane on the counterexample. Contours are level sets of $f$; dot color indicates the iteration $t$.

**Muon and EF-Muon tuning.** On each dataset, Muon and EF-Muon use the same full grid of base learning rates

$$\lambda_{\max} \in \{5\times10^{-5}, 10^{-4}, 2\times10^{-4}, 5\times10^{-4}, 10^{-3}, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2\},$$

LR multipliers $\{0.5, 1, 10, 30\}$, and Nesterov momentum on or off. This gives $12 \times 4 \times 2 = 96$ configurations per method and dataset, each evaluated once. We select the trial with the lowest final recorded validation CE. Both methods use $\beta = 0.95$, five Newton–Schulz steps, and Adam as the paired optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The EF-Muon scale parameter is fixed at $s = 1$; it is not tuned.

### F.1 Image classification

**Signed momentum paired optimizer.** Matrix parameters are updated as above. For MuonMax and EF-MuonMax, paired parameters are updated in the coupled paired mode, using signed momentum

for non-matrix parameters. We use momentum parameter $\beta = 0.95$ and fix the scale parameter at $s = 1$.

**Training setup.** The model is a WideResNet-28-10 with 4 blocks per group and channel multiplier 10. It has 36M parameters. We train on CIFAR-10 dataset with 45K training images and 5K validation images, batch size 128, for 40 epochs (14,062 steps). All trials reach this step count. The learning rate schedule is cosine decay without warmup. We use no weight decay.

**Sweep.** For MuonMax and EF-MuonMax, we sweep learning rate

$$\lambda_{\max} \in \{10^{-4},\ 5\times 10^{-4},\ 10^{-3},\ 5\times 10^{-3},\ 10^{-2},\ 5\times 10^{-2}\}$$

and Muon matrix learning rate (LR) multiplier

$$\{0.1,\ 1,\ 10,\ 30,\ 100\}.$$

We also sweep Nesterov momentum over $\{\mathrm{True}, \mathrm{False}\}$. This gives 60 trials per method. We select the best run by final validation cross-entropy loss. Results are shown in Table 2.

| Method | $\lambda_{\max}$ | LR multiplier | Nesterov | Validation CE |
|---|---|---|---|---|
| MuonMax | $5\times 10^{-4}$ | 100 | False | 0.2651 |
| EF-MuonMax | $5\times 10^{-3}$ | 30 | True | 0.3884 |
| Muon | 0.01 | 0.5 | True | 0.2746 |
| EF-Muon | 0.05 | 30 | True | **0.2574** |

Table 2: Best configurations for image classification task.

### F.2 Language modeling

**NAdamW paired optimizer.** For MuonMax and EF-MuonMax, paired parameters are updated by a separate NAdamW optimizer [Doz16; Med+25], with

$$\beta_1 = 0.9, \qquad \beta_2 = 0.999, \qquad \epsilon = 10^{-8}.$$

This is not covered by the theory, but it is common in practice.

**Training setup.** The model is a 12-layer RoPE transformer with embedding dimension 1024, MLP dimension 4096, 16 attention heads, RMSNorm, and QK-norm. It has 206M parameters. We train on FineWeb-Edu 10B with sequence length 1024, batch size 64, with a target of 100K steps. The learning rate schedule is cosine decay with 10% warmup. For MuonMax and EF-MuonMax, we use momentum $\beta = 0.95$, weight decay $10^{-3}$, and scale $s = 1$. For Muon and EF-Muon, matrix weight decay is 0.01 and paired Adam weight decay is 0.1.

**Sweep.** For MuonMax and EF-MuonMax, we sweep learning rate

$$\lambda_{\max} \in \{10^{-4},\ 5\times 10^{-4},\ 10^{-3},\ 5\times 10^{-3},\ 10^{-2},\ 5\times 10^{-2}\}$$

and Muon matrix LR multiplier

$$\{0.1,\ 1,\ 10,\ 30,\ 100\}.$$

This gives 30 trials per method, with Nesterov momentum off. We select the best run by final validation cross-entropy.

**Run lengths and selected configurations.** The saved logs reach 100K steps for 29/30 MuonMax trials, 28/30 EF-MuonMax trials, 52/96 Muon trials, and 75/96 EF-Muon trials. The other logs end earlier without recording a reason. All four selected trials reach 100K steps. For Muon and EF-Muon, the selected base learning rate is 0.002, the LR multipliers are 1 and 30, respectively, and Nesterov momentum is on.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim in the abstract and introduction that Muon does not converge on the class of convex and Lipschitz functions (proven in Counterexamples 3.3 and 3.4). We also claim that error feedback enables convergence (proven in Theorem 4.2), but does not consistently improve practical performance (evidence in Figure 2).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 7, titled "Limitations".

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our main theoretical results (Counterexamples 3.3, 3.4, Theorem 4.2) are all stated formally with complete assumptions and complete proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Full details of the experiments (including architecture, datasets, optimizers, and tuning procedures) are given in Section 5 and Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We provide full details to reproduce all of our experiments, and our experiments are based on a publicly available repository, but we do not provide source code for the optimizers we evaluated. We use standard datasets and network architectures, for which many public implementations exist, and the main codebase we used to run experiments (init2winit) is already public [Gil+26]. For the optimizers in our experiments, we provide a complete description needed to reproduce results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Full details of the experiments (including architecture, datasets, optimizers, and tuning procedures) are given in Section 5 and Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the computational cost of additional runs, we use a single random seed for each training run. We opted to use compute for thorough tuning of baselines and algorithms, rather than to measure statistical significance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5 and Appendix F provide full information on the details of the experimental setup, including the hardware we used to run experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research does not deviate from the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of our research, as it is essentially purely theoretical.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper poses no risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use CIFAR-10 [KH+09] and FineWeb-Edu [Pen+24] (ODC-By v1.0), and the `init2winit` codebase [Gil+26] (Apache-2.0). All assets are publicly available and used in accordance with their respective licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The research in this paper does not involve LLMs for original or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.