The Ky Fan Norms and Beyond: Dual Norms and Combinations for Matrix Optimization

Alexey Kravatskiy KRAVTSKII.AIU@PHYSTECH.EDU
Moscow Institute of Physics and Technology (MIPT)

Ivan Kozyrev KOZYREV.IN@PHYSTECH.EDU
Moscow Institute of Physics and Technology (MIPT)
Marchuk Institute of Numerical Mathematics

Nikolai Kozlov KOZLOV.NA@PHYSTECH.EDU
Moscow Institute of Physics and Technology (MIPT)

Alexander Vinogradov VINOGRADOV.AM@PHYSTECH.EDU
Moscow Institute of Physics and Technology (MIPT)

Daniil Merkulov DANIIL.MERKULOV@PHYSTECH.EDU
Moscow Institute of Physics and Technology (MIPT)
Skoltech, HSE, AI4Science

Ivan Oseledets I.OSELEDETS@SKOLTECH.RU
AIRI, Skoltech
Marchuk Institute of Numerical Mathematics

Abstract

In this article, we explore the use of various matrix norms for optimizing functions of weight matrices, a crucial problem in training large language models. Moving beyond the spectral norm that underlies the Muon update, we leverage the duals to the Ky Fan k-norms to introduce a family of Muon-like algorithms we name *Fanions*, which happen to be similar to Dion. Then working with the duals of convex combinations of the Ky Fan k-norms and the Frobenius norm or the l_∞ norm, we construct the families of *F-Fanions* and *S-Fanions* respectively. Their most prominent members are *F-Muon* and *S-Muon*. We complement our theoretical analysis with an extensive empirical study of the algorithms across a wide range of tasks and settings, from which it follows that F-Muon and S-Muon are always on par with Muon, while on fine-tuning of NanoGPT and synthetic linear least squares they are even better than vanilla Muon optimizer.

1. Introduction

Minimizing loss functions in unprecedentedly high-dimensional spaces has recently become an integral and crucial part in training large language models. Hence, new scalable, time- and memory-efficient algorithms have been demanded. Besides well-known Adam (**?**) and AdamW (**?**), recently proposed Muon (**?**) has shown promising results on training very large models (**?**). Its key difference from Adam and AdamW is that it has been constructed specifically for optimizing functions of weight matrices, which are common in deep learning.

That is what can be said from a practical point of view. From a theoretical perspective, a key innovation of Muon was its principled derivation of the update rule, which emerged

as the solution to an optimization problem constrained by the RMS-to-RMS norm (scaled version of the spectral norm) (**?**).

Motivated by the success of Muon, many generalizations and variations of it were proposed. Among the notable ones are Scion (**?**), Dion (**?**) and Gluon (**?**). Those works try to explain Muon's efficiency and establish convergence bounds. One central question, however, remains unanswered:

In deriving Muon's update step, why should one constrain by the spectral or any other operator norm? How would alternative norms affect performance and computational cost?

In this article, we tackle this question by actually showing that there are many viable non-operator norms. We leverage the family of norms dual to Ky Fan k-norms to derive a new family of **Fanions**, algorithms with low-rank updates. This approach theoretically explains the backbone of Dion's update (**?**) and generalizes the memory-motivated application of the nuclear norm to Sharpness-Aware Minimization (**?**). As it was done with Muon, we come up with an effective procedure for computing Fanions' updates. Lanczos algorithm, which is described and compared with its competitors in Section 5, is the most operation-effective algorithm, which, however, for now lacks an effective GPU- and PyTorch-friendly implementation.

Working with duals to conic combinations of dual norms, we construct the families of **F-Fanions** and **S-Fanions**, which are hybrids of Muon and NormalizedSGD and SignSGD, respectively.

Then we compare the performances of the algorithm families on various model and real-world problems:

- Synthetic least squares experiment Section **??**
- CIFAR-10 airbench (**?**)
- Pre-training NanoGPT on FineWeb dataset (**?**)
- Fine-tuning NanoGPT on Tiny Stories (**?**)

Our experiments reveal important insights into the role of matrix norms in optimization. First, we show on the example of Neon, the one-rank Fanion, that not every LMO-based algorithm is effective, despite the same asymptotics in the general bounds of **?** and **?**. This suggests that existing theoretical guarantees should be reworked to explain empirical performance.

Most notably, our experiments on real-world tasks demonstrate that the choice of underlying matrix norm is remarkably flexible. On CIFAR-10 airbench, properly-tuned F-Muon and S-Muon achieve $94.02 \pm 0.13\%$ and $94.03 \pm 0.13\%$ accuracy, matching Muon's $94.01 \pm 0.13\%$ performance. On NanoGPT pre-training, F-Muon achieves 3.281 cross-entropy loss, while fully-tuned Muon achieves 3.279. Finally, S-Muon matches Muon on fine-tuning of NanoGPT on Tiny Stories, while F-Muon is far more resistant to the learning rate choice than Muon. These results show that Muon-like algorithms can maintain competitive performance even when the underlying norm constraint is significantly modified, answering affirmatively the central question posed above. Moreover, the tools from Section 4 give the researchers an unheard-of flexibility in designing algorithms that do not have to be modifications of Muon.

2. Preliminaries: Linear Minimization Oracle Framework

Training a neural network is essentially an optimization of a function of several weight matrices and a few vectors. Let us start by considering the problem of minimizing a differentiable function of a *single* matrix:

$$F(\cdot)\colon \mathbb{R}^{m\times n} \to \mathbb{R}\,, \qquad F(\boldsymbol{X}) \to \min_{\boldsymbol{X}\in\mathbb{R}^{m\times n}} \tag{1}$$

We equip the matrix space $\mathbb{R}^{m\times n}$ with a standard dot product $\langle\cdot,\cdot\rangle \to \mathbb{R}$ and a norm $\|\cdot\|\colon \mathbb{R}^{m\times n} \to \mathbb{R}_+$, which does not have to coincide with the Frobenius norm $\|\cdot\|_{\mathrm{F}}$. The dual norm $\|\cdot\|^\dagger\colon \mathbb{R}^{m\times n} \to \mathbb{R}_+$ that is associated with $\|\cdot\|$ is defined as

$$\|\boldsymbol{G}\|^\dagger = \sup_{\boldsymbol{D}\in\mathbb{R}^{m\times n}\,:\|\boldsymbol{D}\|\leq 1} \langle \boldsymbol{G}, \boldsymbol{D}\rangle\,. \tag{2}$$

Such problems can be solved with an iterative algorithm based on the Linear Minimization Oracle (LMO):

$$\mathrm{LMO}(\boldsymbol{M}^k) \in \arg\min_{\boldsymbol{D}\in\mathcal{S}} \langle \boldsymbol{M}^k, \boldsymbol{D}\rangle\,, \tag{3}$$

where $\boldsymbol{M}^k$ is a gradient (or a momentum) of F in $\boldsymbol{X}^k$ and $\mathcal{S}\subset\mathbb{R}^{m\times n}$ is some set. The update of the algorithm is defined as follows:

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k + \gamma_k \mathrm{LMO}\left(\boldsymbol{M}^k\right)\,. \tag{4}$$

We are particularly interested in the case when $\mathcal{S}$ is a unit ball in the norm $\|\cdot\|$:

$$\mathcal{S} = \mathcal{B}_{\|\cdot\|} = \{\boldsymbol{D}\in\mathbb{R}^{m\times n} \mid \|\boldsymbol{D}\|\leq 1\}\,.$$

In this case,

$$\arg\min_{\boldsymbol{D}\in\mathcal{S}} \langle \boldsymbol{M}^k, \boldsymbol{D}\rangle = -\{\boldsymbol{D}\in\mathcal{B}_1 \mid \langle \boldsymbol{M}^k, \boldsymbol{D}\rangle = \|\boldsymbol{M}^k\|^\dagger\}\,,$$

and update for $\boldsymbol{X}^k$ in Equation (4) simplifies to

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \gamma_k\{\boldsymbol{D}\in\mathcal{B}_1 \mid \langle \boldsymbol{M}^k, \boldsymbol{D}\rangle = \|\boldsymbol{M}^k\|^\dagger\}\,. \tag{5}$$

Using this formula it is easy to compute updates for algorithms induced by various norms $\|\cdot\|$:

Frobenius norm and Normalized SGD When the norm $\|\cdot\|$ is the Frobenius norm $\|\cdot\|_{\mathrm{F}}$, Equation (5) turns into

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \eta_k \frac{\boldsymbol{M}^k}{\|\boldsymbol{M}^k\|_{\mathrm{F}}}\,, \tag{6}$$

which recovers Normalized SGD (NSGD).

Spectral norm and Muon When the norm is the spectral norm $\|\cdot\|_2$, we get

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \eta_k \boldsymbol{U}\boldsymbol{V}^\top\,, \tag{7}$$

which is Muon without the $\sqrt{m/n}$ factor. Here, $\boldsymbol{M}^k = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is the Singular Value Decomposition (SVD) of $\boldsymbol{M}^k$ ($\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r]$, $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$, and $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r]$). Muon can be recovered by taking the RMS-to-RMS operator norm $\sqrt{\frac{n}{m}}\|\cdot\|_2$.

Chebyshev norm and SignSGD When the norm is the Chebyshev norm $\|\cdot\|_{\mathrm{C}}$, we get

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \eta_k \operatorname{sign}(\boldsymbol{M}^k)\,, \tag{8}$$

which recovers SignSGD (**?**). Here, $\operatorname{sign}(\boldsymbol{M}^k)$ denotes the element-wise sign function. SignSGD is particularly notable for its communication efficiency in distributed training, as it compresses gradients to 1-bit per parameter.

3. Duals to Ky Fan Norms Instead of the Spectral Norm

3.1. $\|\boldsymbol{M}^k\|_*$ and Neon

After considering $\|\boldsymbol{M}^k\|_{\mathrm{F}}$ and $\|\boldsymbol{M}^k\|_2$, it is natural to look at the nuclear norm $\|\boldsymbol{M}^k\|_*$.

Lemma 1 *When $\|\cdot\| = \|\cdot\|_*$, Equation (5) becomes*

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \eta_k \boldsymbol{u}_1 \boldsymbol{v}_1^\top\,. \tag{9}$$

Proof Since $\|\cdot\|^\dagger = \|\cdot\|_2$, the goal is to reach $\|\boldsymbol{M}^k\|_2 = \sigma_1$. Let us note that $\Delta = \boldsymbol{u}_1\boldsymbol{v}_1^\top$ delivers this value. Indeed, $\|\Delta\|_* = 1$ and by the trace property and orthogonality of the singular vectors,

$$\langle \boldsymbol{M}^k, \Delta\rangle = \langle \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \boldsymbol{u}_1\boldsymbol{v}_1^\top\rangle = \operatorname{tr}\operatorname{diag}(\sigma_1, 0, \ldots, 0) = \|\boldsymbol{M}^k\|_2\,,$$

which completes the proof. ■

We name the derived algorithm *Neon*. In Section 5, we will discuss how to compute the Neon's update.

3.2. $\|\boldsymbol{M}^k\|^\dagger_{\textbf{KF-}k}$ and Muon, Neon, and Centralized Dion without error feedback

Neon's and Muon's updates seem to be complete opposites: one has rank one, while the other is full-rank. It would be interesting to derive algorithms with updates of intermediate ranks.

3.2.1. Schatten norms

? considered Schatten-p norms:

$$\|\boldsymbol{M}^k\|_{S_p} = \left(\sum_{i=1}^{\min(m,n)} \sigma_i^p\right)^{1/p}\,,$$

which produce the update

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \eta_k \boldsymbol{U} \frac{\operatorname{diag}\left(\sigma_1^{q-1}, \ldots, \sigma_{\min(m,n)}^{q-1}\right)}{\left(\sum_{i=1}^{\min(m,n)} \sigma_i^q\right)^{\frac{q-1}{q}}} \boldsymbol{V}^\top$$

for p and q such that: $\frac{1}{p}+\frac{1}{q}=1$. This formula recovers Neon when $p\to 1$ provided that $\sigma_1>\sigma_2$, which is true on real data; NSGD when $p=2$, and Muon when $p\to\infty$.

However, Schatten norms do not fill the gap rank: indeed, when $p>1$, the rank of the update is full, while when $p=1$ ($p\to 1$) it is one. Moreover, it is not clear how to calculate the update for $p\neq 1,2,\infty$: it seems one has to know all σ_i to compute the update, which makes the problem as hard as the full SVD.

3.2.2. Ky Fan norms

There is another family of matrix norms, which might help us: Ky Fan norms. For $k\in\{1,\dots,\min(m,n)\}$, the Ky Fan k-norm $\|\cdot\|_{\text{KF-}k}$ is $\sum_{i=1}^{k}\sigma_i$, i.e. the sum of the k largest singular values of the matrix. Special cases of the Ky Fan k-norm are the Ky Fan 1-norm, which is the spectral norm, and the Ky Fan $\min\{m,n\}$-norm, which is the nuclear norm.

Let us derive the update for an arbitrary k. Since $\|\cdot\|^{\dagger}_{\text{KF-}k}=\max\{\frac{1}{k}\|\cdot\|_*,\|\cdot\|_2\}$ (see **?**, p. 96), the goal is to reach $\max\{\frac{1}{k}\sum_{i=1}^{\min(m,n)}\sigma_i,\sigma_1\}$. $\Delta=\boldsymbol{u}_1\boldsymbol{v}_1^\top$ from Neon delivers σ_1, while $\Delta=\frac{1}{k}\sum_{i=1}^{\min(m,n)}\boldsymbol{u}_i\boldsymbol{v}_i^\top$ from Muon (with an extra factor of $1/k$) delivers $\frac{1}{k}\sum_{i=1}^{\min(m,n)}\sigma_i$. Thus, the update is either

$$\boldsymbol{X}^{t+1}=\boldsymbol{X}^t-\eta_t\boldsymbol{u}_1\boldsymbol{v}_1^\top \text{ or } \boldsymbol{X}^{t+1}=\boldsymbol{X}^t-\frac{\eta_t}{k}\boldsymbol{U}\boldsymbol{V}^\top,$$

depending on which one better minimizes the linear function $L(\boldsymbol{X})\equiv F(\boldsymbol{X}^t)+\boldsymbol{M}^t(\boldsymbol{X}-\boldsymbol{X}^t)$.

The rank gap is not closed by the Ky Fan norms because the obtained update is either one-rank or full-rank.

3.2.3. Duals to Ky Fan norms

While Schatten norms are closed under dualization and $\|\boldsymbol{M}^k\|^{\dagger}_{S_p}=\|\boldsymbol{M}^k\|_{S_q}$, for Ky Fan norms it is not generally the case. $k=1$ and $k=\min(m,n)$ are exceptional: for $k=1$ the dual norm is $\|\cdot\|_*$ with $k=\min(m,n)$ and for $k=\min(m,n)$ the dual norm is $\|\cdot\|_2$ with $k=1$. For each other k, $\|\cdot\|_{\text{KF-}k}\neq\|\cdot\|_{\text{KF-}k'}$ for any k': $k'=\min(m,n)$ and $k'=1$ correspond to the already discussed cases, while for other k' one can change $\|\boldsymbol{M}^k\|_{\text{KF-}k'}$ by moving a small value between $\sigma_{k'}$ and $\sigma_{k'+1}$. During this change, $\|\boldsymbol{M}^k\|_2$ and $\|\boldsymbol{M}^k\|_*$ will remain constant, hence $\|\boldsymbol{M}^k\|_{\text{KF-}k}$ will not change as well.

Lemma 2 *When $\|\cdot\|=\|\boldsymbol{M}^t\|^{\dagger}_{KF\text{-}k}$, Equation (5) turns into:*

$$\boldsymbol{X}^{t+1}=\boldsymbol{X}^t-\eta\sum_{i=1}^{k}\boldsymbol{u}_i\boldsymbol{v}_i^\top \tag{10}$$

Proof Since $\|\cdot\|^{\dagger}=\|\cdot\|^{\dagger\dagger}_{\text{KF-}k}=\|\cdot\|_{\text{KF-}k}$, the goal is to reach $\|\boldsymbol{M}^t\|_{\text{KF-}k}$.

Let us note that $\Delta=\sum_{i=1}^{k}\boldsymbol{u}_i\boldsymbol{v}_i^\top$ delivers the value. Indeed,

$$\langle\boldsymbol{M}^t,\Delta\rangle=\langle\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top,\sum_{i=1}^{k}\boldsymbol{u}_i\boldsymbol{v}_i^\top\rangle=\sum_{i,j=1}^{r,k}\langle\boldsymbol{u}_i\sigma_i\boldsymbol{v}_i^\top,\boldsymbol{u}_j\boldsymbol{v}_j^\top\rangle=\sum_{i=1}^{k}\sigma_i=\|\boldsymbol{M}^t\|_{\text{KF-}k},$$

which completes the proof. ■

These updates constitute the family of *Fanions*, LMO-based algorithms under the $\|\boldsymbol{M}^t\|^\dagger_{\text{KF-}k}$ norms. The algorithm for a particular k we will call *Fanion-k*. Thus, Muon is a Fanion-min$\{m, n\}$, while Neon is a Fanion-1. Moreover, the unsharded version of the rank-r Dion (Algorithm 1 from **?**) without the error feedback and without scaling of the update is actually a Fanion-r (see **?**, where Dion is written down in a notation more similar to ours).

4. Conic Combination of LMO-algorithms is an LMO-algorithm

Simply applying norm dual to the Ky Fan k-norm however, is not enough to provide family of algorithms diverse enough for our cause. Next we consider linear combinations of algorithms from LMO framework, which happen to be LMO-algorithms too, as we show in succeeding paragraphs.

4.1. General Case

First, we need a well-known fact mentioned, for example, in **?**, Table 1. For the sake of completeness, we provide a proof here.

Lemma 3 *Let $\|\cdot\|_{(1)}, \ldots, \|\cdot\|_{(n)}$ be norms on a finite-dimensional Euclidean space, and let $\alpha_1, \ldots, \alpha_n$ be non-negative reals. Define*

$$\|\cdot\| := \sum_{i=1}^{n} \alpha_i \|\cdot\|_{(i)}.$$

Then the dual unit ball of $\|\cdot\|$ satisfies

$$\mathcal{B}_{\|\cdot\|^\dagger} = \sum_{i=1}^{n} \alpha_i \mathcal{B}_{\|\cdot\|^\dagger_{(i)}}$$

where $\sum$ denotes the Minkowski sum and $\mathcal{B}_{\|\cdot\|^\dagger_{(i)}}$ is the unit ball of the dual norm $\|\cdot\|^\dagger_{(i)}$.

Proof Let us first prove the lemma for the case $n = 2$. Denote $f(x) = \alpha_1\|x\|_{(1)}$ and $g(x) = \alpha_2\|x\|_{(2)}$, such that $\|x\| = f(x) + g(x)$. Recall two standard facts:

1. For any norm $\|\cdot\|$ and $\lambda > 0$,

$$(\lambda\|\cdot\|)^*(y) = \sup_x \big(\langle y, x\rangle - \lambda\|x\|\big) = \delta_{\lambda\mathcal{B}_{\|\cdot\|^\dagger}}(y)\,,$$

   i.e., the indicator function of the scaled dual ball.

2. The Fenchel conjugate of a sum satisfies

$$(f + g)^*(y) = \inf_{u+v=y} \big(f^*(u) + g^*(v)\big)\,.$$

Applying these to f and g, we have

$$f^*(u) = \delta_{\alpha_1 \mathcal{B}_{\|\cdot\|_{(1)}^\dagger}}(u)\,, \quad g^*(v) = \delta_{\alpha_2 \mathcal{B}_{\|\cdot\|_{(2)}^\dagger}}(v)\,.$$

Thus,

$$\|\cdot\|^*(y) = (f+g)^*(y) = \inf_{u+v=y}\left(\delta_{\alpha_1 \mathcal{B}_{\|\cdot\|_{(1)}^\dagger}}(u) + \delta_{\alpha_2 \mathcal{B}_{\|\cdot\|_{(2)}^\dagger}}(v)\right) = \delta_{\alpha_1 \mathcal{B}_{\|\cdot\|_{(1)}^\dagger} + \alpha_2 \mathcal{B}_{\|\cdot\|_{(2)}^\dagger}}(y).$$

By definition, the conjugate of a norm is exactly the indicator of its dual unit ball:

$$\|\cdot\|^*(y) = \delta_{\mathcal{B}_{\|\cdot\|^\dagger}}(y)\,.$$

Therefore, $\mathcal{B}_{\|\cdot\|^\dagger} = \alpha_1 \mathcal{B}_{\|\cdot\|_{(1)}^\dagger} + \alpha_2 \mathcal{B}_{\|\cdot\|_{(2)}^\dagger}$.

Now we prove the general case by induction. The base case ($n = 2$) is already proven. Suppose that the assumption of the lemma holds for $n = k$. Then, for $n = k+1$,

$$\|x\| = \sum_{i=1}^{k} \alpha_i \|x\|_{(i)} + \alpha_{k+1}\|x\|_{(k+1)} = \|x\|_{(1:k)} + \alpha_{k+1}\|x\|_{(k+1)}\,.$$

Applying the result for $n = 2$ combined with the induction assumption, we obtain

$$\mathcal{B}_{\|\cdot\|^\dagger} = \mathcal{B}_{\|\cdot\|_{(1:k)}^\dagger} + \alpha_{k+1}\mathcal{B}_{\|\cdot\|_{(k+1)}^\dagger} = \sum_{i=1}^{k+1} \alpha_i \mathcal{B}_{\|\cdot\|_{(i)}^\dagger}\,,$$

which proves the lemma. ■

Lemma 4 *Let $\|\cdot\|_{(1)}, \ldots, \|\cdot\|_{(n)}$ be norms on a finite-dimensional Euclidean space, and let $\alpha_1, \ldots, \alpha_n$ be non-negative reals. Consider Linear Minimization Oracles $\mathrm{LMO}_1, \ldots, \mathrm{LMO}_n$, corresponding to the unit balls of these norms. Then, $\sum_{i=1}^n \alpha_i \mathrm{LMO}_i$ is the LMO corresponding to the norm $\|\cdot\|$ dual to the $\sum_{i=1}^n \alpha_i \|\cdot\|_{(i)}^\dagger$.*

Proof Using Lemma 3 and the fact $\|\cdot\|^{\dagger\dagger} = \|\cdot\|$, we can obtain general form of the unit ball in the $\|\cdot\|$ norm: $\mathcal{B}_{\|\cdot\|} = \sum_{i=1}^n \alpha_i \mathcal{B}_{\|\cdot\|_{(i)}}$. Thus, the linear function minimization objective on a $\|\cdot\|$ norm ball can be transformed as follows:

$$\arg\min_{\boldsymbol{D} \in \mathcal{B}_{\|\cdot\|}} \langle \boldsymbol{M}, \boldsymbol{D}\rangle = \arg\min_{\boldsymbol{D}_1 \in \alpha_1 \mathcal{B}_{\|\cdot\|_{(1)}}, \ldots, \boldsymbol{D}_n \in \alpha_n \mathcal{B}_{\|\cdot\|_{(n)}}} \langle \boldsymbol{M}, \sum_{i=1}^n \boldsymbol{D}_i\rangle = \sum_{i=1}^n \arg\min_{D_i \in \mathcal{B}_{\|\cdot\|_{(i)}}} \langle \boldsymbol{M}, \boldsymbol{D}_i\rangle\,,$$

where the last summation denotes the Minkowski sum. This immediately implies $\sum_{i=1}^n \alpha_i \mathrm{LMO}_i \in \arg\min_{\boldsymbol{D} \in \mathcal{B}_{\|\cdot\|}} \langle \boldsymbol{M}, \boldsymbol{D}\rangle$, which proves the claim of the lemma. ■

Applying it to optimization algorithms, we obtain the following corollary.

Corollary 5 *Let there be a finite family of LMO based algorithms indexed by $i = 1, \dots, n$, where the update of the i-th algorithm is defined by*

$$\boldsymbol{X}^{k+1} - \boldsymbol{X}^k = \gamma_k \mathrm{LMO}_i(\boldsymbol{M}^k)\,,$$

and LMO_i *corresponds to the unit ball of norm $\|\cdot\|_i$. For arbitrary non-negative $\alpha_1, \dots, \alpha_n$, the algorithm with the update given by*

$$\boldsymbol{X}^{k+1} - \boldsymbol{X}^k = \gamma_k \sum_{i=1}^{n} \alpha_i \mathrm{LMO}_{\|\cdot\|_{(i)}}(\boldsymbol{M}^k)$$

is an LMO-algorithm itself, with LMO corresponding to the unit ball of the norm $\|\cdot\|$ dual to the norm given by $\sum_{i=1}^n \alpha_i \|\cdot\|_{(i)}^\dagger$.

4.2. Frobeniusize Them: F-Muon and F-Neon

Let us construct the concrete examples of algorithms given by linear combinations of LMO-algorithms. It follows from Corollary 5 that those algorithms can also be viewed as LMO-algorithms.

Combining Fanions for various k with another excellent LMO-algorithm, NSGD, we obtain the family of algorithms with updates defined by

$$\boldsymbol{X}^{t+1} = \boldsymbol{X}^t - \gamma_k \left(\alpha \sum_{i=1}^{k} \boldsymbol{u}_i \boldsymbol{v}_i^\top + (1-\alpha) \frac{\boldsymbol{M}^k}{\|\boldsymbol{M}^k\|_{\mathrm{F}}} \right). \tag{11}$$

Recall that Fanion-k update is governed by the norm dual to the Ky Fan-k norm, and NSGD corresponds to the Frobenius norm, which is dual to itself. Thus, by Corollary 5, our new algorithm is an LMO-algorithm corresponding to the unit ball of the norm $\|\cdot\|_{\mathrm{F-KF-k}}$:

$$\|\cdot\|_{\mathrm{F-KF-k}}^\dagger = \alpha \|\cdot\|_{\text{KF-}k} + (1-\alpha)\|\cdot\|_{\mathrm{F}}\,. \tag{12}$$

We name the derived family of algorithms *F-Fanions*.

Two edge members of this family, with $k = 1$ and $k = \min\{m, n\}$ correspondingly, *F-Neon* and *F-Muon*, are of particular interest to us. A bit of information and visualizations related to the $\|\cdot\|_{\mathrm{F}*} = \|\cdot\|_{\mathrm{F-KF-1}}$ norm is presented in the appendix (see Equation (**??**), Figure **??**).

4.3. Add SignSGD: S-Muon and S-Neon

Another algorithm we consider is the SignSGD. Mixing its update with that of Fanion-k, we obtain the update of *S-Fanion-k*:

$$\boldsymbol{X}^{t+1} = \boldsymbol{X}^t - \gamma_k \left(\alpha \sum_{i=1}^{k} \boldsymbol{u}_i \boldsymbol{v}_i^\top + (1-\alpha)\eta\, \mathrm{sign}(\boldsymbol{M}^k) \right), \tag{13}$$

where η is the special SignSGD learning rate coefficient.

The norm $\|\cdot\|_{\mathrm{C-KF-k}}$ which produces this update in the LMO framework is again given by Corollary 5:

$$\|\cdot\|_{\mathrm{C-KF-k}}^\dagger = \alpha \|\cdot\|_{\text{KF-}k} + \frac{1-\alpha}{\eta}\|\cdot\|_{\mathrm{C}}^\dagger\,. \tag{14}$$

5. Computing the Updates

We employ the thick-restarted Lanczos method for the symmetric eigenvalue problem (thick-restart Lanczos, TRLan) to compute the low-rank updates of Fanions. We apply TRLan to either $\boldsymbol{M}^{k\top}\boldsymbol{M}^k$ or $\boldsymbol{M}^k\boldsymbol{M}^{k\top}$ (whichever matrix is smaller). We use the CuPy implementation of `cupy.sparse.linalg.svds` (**?**) which internally relies on TRLan (**?**).

TRLan is specifically designed for efficiently approximating the largest singular values and corresponding singular vectors of very large matrices. The thick-restart strategy retains the most informative Ritz vectors across restarts, which dramatically accelerates convergence while keeping memory consumption moderate. TRLan is particularly attractive in our GPU setting because its dominant cost is a modest number of highly parallelizable matrix-vector multiplications (matvecs) and it avoids full reorthogonalization against the entire Krylov basis by using short recurrence relations combined with thick restarting.

The per-cycle complexity is $\mathcal{O}(mn^2 + n^2k + nk^2)$, where $m \geq n$ are the dimensions of the target matrix and k is the size of the retained subspace (typically $k \ll n$).

In Table 1, we compare TRLan against randomized SVD (RSVD) and simple power iterations when computing the rank-k update used in Fanion-k and related algorithms. Experiments are performed on dense random matrices with i.i.d. $\mathcal{N}(0,1)$ entries and use CPU implementations for fair comparison. We report:

- err_1: relative error in the Frobenius norm of $\sum_i^k \boldsymbol{u}_i\boldsymbol{v}_i^T$,
- err_2: relative error in the Frobenius norm of $\sum_i^k \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^T$.

On 500×500 matrices, TRLan and RSVD require comparable wall-clock time, but TRLan delivers orders-of-magnitude lower error and far fewer matvecs. On larger 5000×5000 matrices the advantage becomes even more pronounced: TRLan is 3-4 times faster than RSVD while using $\sim$30 times fewer matvecs at comparable or better accuracy.

An interesting empirical observation is that RSVD tends to approximate the *singular values* themselves reasonably well but the reconstructed low-rank matrix noticeably deviates from the truncated SVD, whereas TRLan provides an excellent approximation to the truncated SVD matrix itself (low err_2) at the cost of occasionally less accurate individual singular values. This makes TRLan the preferred choice for algorithms like Neon/Fanion-k that only need the low-rank term $\sum \sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^\top$, but less ideal for methods (e.g., Dion) that explicitly require accurate σ_i for error feedback or step-size control.

A current practical limitation is the lack of a native PyTorch implementation of thick-restart Lanczos; existing PyTorch-based randomized SVD routines cannot match TRLan's accuracy/efficiency combination for the matrix reconstruction task.

For reference, Table 2 shows results for the Newton-Schulz polar decomposition iteration on the same matrices, err_1 is the relative error of $\boldsymbol{U}\boldsymbol{V}^T$ (29-30 iterations to converge, significantly higher matvec count than TRLan).

References

Kwangjun Ahn, Byron Xu, Natalie Abreu, Ying Fan, Gagik Magakyan, Pratyusha Sharma, Zheng Zhan, and John Langford. Dion: Distributed orthonormalized updates, 2025. URL https://arxiv.org/abs/2504.05295.

Matrix sizes	k	Method	Time (s)	Matvecs	Iterations	err_1	err_2
500×500	5	Power Iterations	0.062	2005	200	$9.2{\cdot}10^{-3}$	$9.1{\cdot}10^{-3}$
500×500	5	RSVD	0.017	1170	38	$9.8{\cdot}10^{-3}$	$9.6{\cdot}10^{-3}$
500×500	5	TRLan	0.018	131	65	$9.6{\cdot}10^{-5}$	$9.4{\cdot}10^{-5}$
500×500	50	Power Iterations	0.44	43750	437	$9.9{\cdot}10^{-3}$	$9.0{\cdot}10^{-3}$
500×500	50	RSVD	0.61	6120	50	$9.9{\cdot}10^{-3}$	$9.1{\cdot}10^{-3}$
500×500	50	TRLan	0.16	462	231	$3.3{\cdot}10^{-7}$	$3.0{\cdot}10^{-7}$
5000×5000	5	Power Iterations	9.6	9065	906	$8.6{\cdot}10^{-3}$	$8.6{\cdot}10^{-3}$
5000×5000	5	RSVD	2.1	5640	187	$9.7{\cdot}10^{-3}$	$9.7{\cdot}10^{-3}$
5000×5000	5	TRLan	0.70	205	102	$7.7{\cdot}10^{-3}$	$7.7{\cdot}10^{-3}$

Table 1: Comparison of methods for computing rank-k updates on dense random matrices (CPU, double precision). Lower is better in all columns.

Matrix size	Time (s)	Matvecs	Iterations	err_1
500×500	0.041	27 000	27	4.8e-3
5000×5000	26.4	290 000	29	6.5e-3

Table 2: Newton–Schulz iteration on the same matrices (for reference).

Jeremy Bernstein. Deriving muon, 2025. URL https://jeremybernste.in/writing/deriving-muon.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International conference on machine learning*, pages 560–569. PMLR, 2018.

Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

Franz Louis Cesista. Steepest Descent under Schatten-p Norms, February 2025. URL https://leloykun.github.io/ponder/steepest-descent-schatten-p/.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023. URL https://arxiv.org/abs/2305.07759.

Keller Jordan, Jeremy Bernstein, Brendan Rappazzo, @fernbear.bsky.social, Boza Vlado, You Jiacheng, Franz Cesista, Braden Koszarsky, and @Grad62304977. modded-nanogpt: Speedrunning the nanogpt baseline, 2024a. URL https://github.com/KellerJordan/modded-nanogpt.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024b. URL https://kellerjordan.github.io/posts/muon/.

Jordan Keller. cifar10-airbench, 2023. URL https://github.com/KellerJordan/cifar10-airbench.

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

Dmitry Kovalev. Understanding gradient orthogonalization for deep learning via non-euclidean trust-region optimization. *arXiv preprint arXiv:2503.12645*, 2025.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Thomas Pethick. Understanding dion, 2025. URL https://pethick.dk/posts/2025-08-18-understanding-dion/.

Thomas Pethick, Parameswaran Raman, Lenon Minorics, Mingyi Hong, Shoham Sabach, and Volkan Cevher. νSAM: Memory-efficient sharpness-aware minimization via nuclear norm constraints. *Transactions on Machine Learning Research*, 2025a. ISSN 2835-8856. URL https://openreview.net/forum?id=V6ia5hWIMD.

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained lmos. *arXiv preprint arXiv:2502.07529*, 2025b.

Inc. Preferred Infrastructure and CuPy Developers. CuPy: `cupyx.scipy.sparse.linalg.svds` — api reference. https://docs.cupy.dev/en/stable/reference/generated/cupyx.scipy.sparse.linalg.svds.html, 2025. Accessed: 2025-08-24.

Artem Riabinin, Egor Shulgin, Kaja Gruntkowska, and Peter Richtárik. Gluon: Making muon & scion great again!(bridging theory and practice of lmo-based optimizers for llms). *arXiv preprint arXiv:2505.13416*, 2025.

Kesheng Wuzand Horst Simonz. Thick-restart lanczos method for symmetric eigenvalue problemsy. 1998.

Yao-Liang Yu. Arithmetic duality for norms, 2012. URL https://cs.uwaterloo.ca/~y328yu/notes/normduality.pdf.