When to use what Schatten-p norm in deep learning?

Thomas Pethick

2026

Why do we care about Schatten-p norms?

Stochastic spectral descent

- Recently an explosion in spectral descent based optimizer
- Muon was used to train Kimi K2 (1T) and GLM-5 (744B)
- Originates from stochastic spectral descent [Car+15], for $x \in \mathbb{R}^{m \times n}$:

$$U \operatorname{diag}(s) V^\top = \operatorname{SVD}(\nabla f(x^k, \xi_k))$$
$$x^{k+1} = x^k - \gamma_k \|s\|_1 U V^\top$$

- SDD is steepest descent under Schatten-∞ geometry:

$$x^{k+1} \in \arg\min_x \langle \nabla f(x^k, \xi_k), x \rangle + \frac{1}{2\gamma_k} \left\| x - x^k \right\|_{S_\infty}^2$$

Empirical state of affairs

Seemingly conflicting observations:

- for speedruns Muon shines [Jor24; Jor+24a]
- large-scale studies are less clear [Wen+25; SPJ25]
- SGD remains strong at extremely low batch sizes [SGO25; Mar+25]
- finite Schatten-p norms can beat Muon [Pan+26; Abr26; Shu+26]

How can these possibly be true simultaneously?

We ask

What Schatten-p norm should the optimizer use, and when?

What is on the agenda?

Setup

- d dimensions
- N tokens (a.k.a. stochastic oracle calls)

Best Schatten-p choice is regime dependence!

- high-dimensional regime ($d \gtrsim N$): Muon is provably beneficial
- low-dimensional regime ($d \lesssim N$): lower p can be better
  even the problem geometry is Schatten-∞!
- Chinchilla scaling puts us in the low-dimensional regime

. . . and optimal batch size will be p-dependent.

What is on the agenda?

Mechanisms at play. The choice of p-norm connects to three things:

- it should match the geometry of the neural network
- it changes the ability to accelerate
- the heavy-tailedness of the noise

Muon as Optimistic Dual Averaging

SGD with momentum

PyTorch SGD with momentum and weight decay

SGD

$$
\begin{aligned}
m^k &= (1-\alpha)m^{k-1} + \alpha \nabla f(x^k, \xi_k) \\
\bar{m}^k &= (1-\alpha)m^k + \alpha \nabla f(x^k, \xi_k) && \text{(Nesterov momentum)} \\
x^{k+1} &= x^k - \eta_k \mu x^k - \eta_k \bar{m}^k && \text{(weight decay)}
\end{aligned}
$$

Muon

PyTorch SGD with momentum and weight decay + msign

Muon [Jor+24b]

$$
\begin{aligned}
m^k &= (1-\alpha)m^{k-1} + \alpha \nabla f(x^k, \xi_k) \\
\bar{m}^k &= (1-\alpha)m^k + \alpha \nabla f(x^k, \xi_k) && \text{(Nesterov momentum)} \\
x^{k+1} &= x^k - \eta_k \mu x^k - \eta_k \operatorname{msign}(\bar{m}^k) && \text{(weight decay)}
\end{aligned}
$$

with msign : $U \operatorname{diag}(\sigma) V^\top \mapsto UV^\top$ (approximated efficiently on GPUs)

Optimistic Dual Averaging

Nesterov momentum (in Lion, Muon, NAdam) can be viewed as:

Optimistic Dual Averaging (ODA)

$$
\begin{aligned}
m^{k+1} &= (1-\alpha_k)m^k + \alpha_k \nabla f(z^k, \xi_k) \\
\bar{m}^{k+1} &= (1-\bar{\alpha}_k)m^{k+1} + \bar{\alpha}_k \nabla f(z^k, \xi_k) \quad \text{(optimism)} \\
z^{k+1} &\in \partial h^*(-\gamma_k \bar{m}^{k+1}) := \arg\min_x \gamma_k \langle x, \bar{m}^{k+1} \rangle + h(x) \\
x^{k+1} &= (1-\lambda_k)x^k + \lambda_k z^{k+1}
\end{aligned}
$$

SODA

Nesterov momentum (in Lion, Muon, NAdam) can be viewed as:
Optimistic Dual Averaging (ODA)

$$
\begin{aligned}
m^{k+1} &= (1-\alpha_k)m^k + \alpha_k \nabla f(y^k, \xi_k) \\
\bar{m}^{k+1} &= (1-\bar{\alpha}_k)m^{k+1} + \bar{\alpha}_k \nabla f(y^k, \xi_k) \quad \text{(optimism)} \\
z^{k+1} &\in \partial h^*(-\gamma_k \bar{m}^{k+1}) := \arg\min_x \gamma_k \langle x, \bar{m}^{k+1} \rangle + h(x) \\
x^{k+1} &= (1-\lambda_k)x^k + \lambda_k z^{k+1} \\
y^{k+1} &= (1-\bar{\lambda}_k)x^{k+1} + \bar{\lambda}_k z^{k+1} \qquad \text{(primal extrapolation)}
\end{aligned}
$$

SODA: Schedule-free [Def+24] + ODA.

For $\bar{\lambda}_k = 0$ we have $y^k = x^k$, which reduces to:

- PyTorch SGD for $h(x) = \frac{1}{2}\|x\|_2^2$ and Muon for $h(x) = \frac{1}{2}\|x\|_{S_\infty}^2$
- Optimistic version of Double Averaging [NS15]
- Non-adaptive version of [Jou+20]

First challenge

- Good news: Muon with weight decay fits ODA + schedule-free template.
- Bad news:
  - Analysis typically require strongly convex reference functions.
  - This restricts us to $h(x) = \frac{1}{2}\|x\|_p^2$ for $p \in (1, 2]$.

Our savior: Uniform convexity

$$h(u) = \frac{1}{p}\|u\|_p^p \qquad \text{is } p\text{-uniformly convex.}$$

ℓ_p case:

$$\nabla h^*(\bar{m}) \;=\; \operatorname{sign}(\bar{m}) \odot |\bar{m}|^{q-1}, \qquad q = \frac{p}{p-1}$$

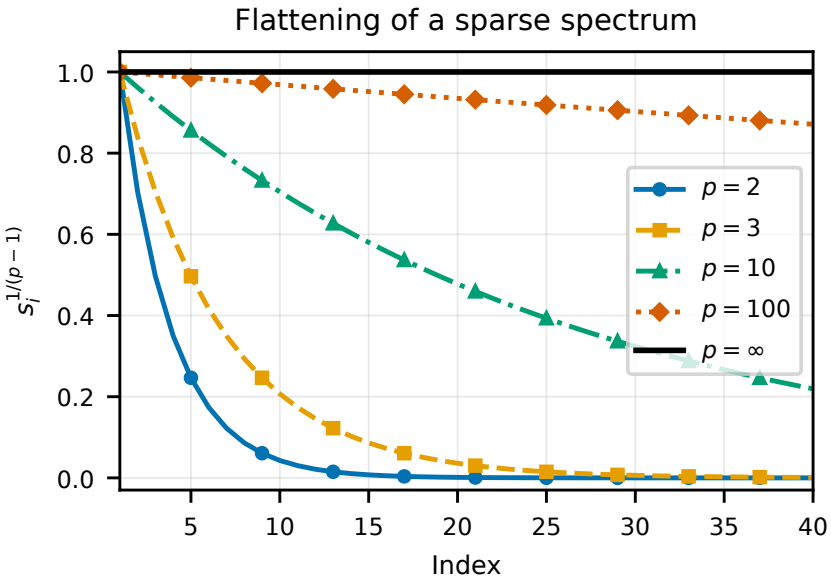
Schatten-p case:

$$\nabla h^*(\bar{M}) \;=\; U \operatorname{diag}(s^{q-1}) V^\top$$

with $\bar{M} = U \operatorname{diag}(s) V^\top$.

Interpolates between

- unnormalized "SGD" at $p = 2$
- normalized sign map at $p = \infty$



Unification

Method	Reference	Opt.	Norm	Dual map ($\partial h^* / \nabla h^*$)
Lion	[Che+23]	✓	ℓ_∞	$\operatorname{sign}(u)$
	–		ℓ_p	$\operatorname{sign}(u) \odot \lvert u \rvert^{q-1}$
	–		mixed $\ell_{\infty,p}$	$\lVert Y_i \rVert_1^{q-1} \operatorname{sign}(Y_i)$
Muon	[Jor+24b]	✓	Schatten-∞	$UV^\top$
HTMuon†	[Pan+26]	✓	Schatten-p	$U \operatorname{diag}(s^{q-1}) V^\top$
Scion (RowNorm)	[Pet+25]	✗	$2 \to \infty$	$Y_i / \lVert Y_i \rVert_2$
	–		mixed $\ell_{2,p}$	$\lVert Y_i \rVert_2^{q-2} Y_i$

† The same finite-Schatten norm is used in Soft-Muon and Freon [Abr26; Shu+26].

Analysis

Noise assumption

h p-uniformly convex $\Rightarrow$ h^* is q-uniformly smooth.

Assumption Gradient-variation condition

For $q = \frac{p}{p-1}$, assume that for $k \geq 0$,

$$\mathbb{E}\left[\left\|g^k - g^{k-1}\right\|_*^q\right]^{1/q} \leq \tau_q \mathbb{E}\left[\left\|\nabla f(y^k) - \nabla f(y^{k-1})\right\|_*^q\right]^{1/q} + \sigma_q.$$

Heavy-tailed: It follows from bounded q-moment stochastic gradients,

$$\mathbb{E}[\left\|\nabla f(y^k, \xi_k) - \nabla f(y^k)\right\|_*^q \mid \mathcal{F}_{k-1}] \leq \bar{\sigma}_q^q,$$

with $\tau_q = 1$ and $\sigma_q = 2\bar{\sigma}_q$.

What can we prove?

We can provide a noise-robust accelerated parameterization for given p:

$$\alpha_k = \frac{2}{k+2}, \qquad \bar{\alpha}_k = \lambda_k = \frac{2}{k+3}, \qquad \bar{\lambda}_k \leq \frac{\lambda_k}{6}.$$

Suppose

- f is convex
- f is L_p-smooth w.r.t. $\|\cdot\|_p$
- Unbiased + gradient-variation condition
- $h(u) = \frac{1}{p}\left\|u - z^0\right\|_p^p$ with $R_p := \|x^\star - z^0\|_p$

Then

$$\mathbb{E}[f(x^{n-1}) - f(x^\star)] = O\left(\frac{L_p R_p^2}{n^{1+2/p}} + \frac{\sigma_q R_p}{n^{1/p}}\right).$$

- the allowed acceleration decreases with p
- the admissible noise can be increasingly heavy-tailed as p grows.

Intuition

$$\mathbb{E}[f(x^{n-1}) - f(x^\star)] = O\left(\frac{L_p R_p^2}{n^{1+2/p}} + \frac{\sigma_q R_p}{n^{1/p}}\right).$$

- Pick too small p: pay a geometry mismatch price if the problem geometry has larger p.
- Pick too large p: pay through slower stochastic convergence and weaker acceleration.

A batch size scaling rule

- $O(\frac{\sigma_q R_p}{n^{1/p}})$ is very slow for large p.
- What if noise is less heavy-tailed, $r \leq q$?

Mini-batch

$$\bar{g}(x) := \frac{1}{B} \sum_{i=1}^{B} \nabla f(x, \xi_i)$$

Norm conversion + von Bahr–Esseen:

$$\sigma_q \leq d^{1/q-1/r} \sigma_r B^{-1/p_r},$$

where $p_r := r/(r-1)$.

Batch size scaling rule

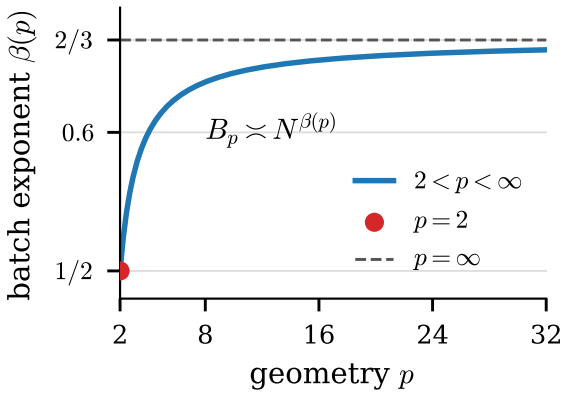
$$B_p \asymp N^{\frac{1-1/p}{1+1/p_r-1/p}}.$$

A batch size scaling rule: how to understand it?

Bounded variance case:

Geometry	Iterations	Batch size	Rate
$p = 2$	$N^{1/2}$	$N^{1/2}$	$N^{-1/2}$
$2 < p < \infty$	$N^{p/(3p-2)}$	$N^{2(p-1)/(3p-2)}$	$N^{-p/(3p-2)}$
$p \to \infty$	$N^{1/3}$	$N^{2/3}$	$N^{-1/3}$

For $p \to \infty$ this recovers the BST scaling rule [Isl+26].



- It looks like Euclidean is always better??
- Only if we ignore the dimensionality factor...

Regimes

Regimes: Deterministic case

Modular norm [BN24] Neural networks natural geometry is Schatten-∞.

Mismatch will pay a dimensionality price: $R_p := \left\|x^\star - z^0\right\|_p \leq d^{1/p} R_\infty$

Deterministic case

$$\epsilon_p^{\mathrm{det}}(N) \lesssim \frac{L_\infty R_\infty^2 d^{2/p}}{N^{1+2/p}} = \frac{L_\infty R_\infty^2}{N} \left(\frac{d}{N}\right)^{2/p}.$$

High-dimensional:

$$N \lesssim d$$

- Small p pays the dim. price.
- Prefer large p.

Low-dimensional:

$$N \gtrsim d$$

- Acceleration wins.
- Prefer small p.

Regimes: Stochastic case

Stochastic case: We can do the same!

Setting	Regime	Preferred geometry
Deterministic ($\sigma_q = 0$)	$N \gtrsim d$	$p = 2$
Deterministic ($\sigma_q = 0$)	$N \lesssim d$	$p \to \infty$
Heavy-tailed noise ($\sigma_r > 0$)	$N \gtrsim d \max\{1, (S/A)^{p_r}\}$	$p = p_r$
Heavy-tailed noise ($\sigma_r > 0$)	$N \lesssim d \max\{1, (S/A)^{p_r}\}$	$p \to \infty$

$A := L_\infty R_\infty^2$ and $S := \sigma_r R_\infty$

Scaling laws

What regime is LLM training in?

Chinchilla [Hof+22]:

$$\texttt{model\_size} \propto N.$$

For a matrix parameter in a depth-D network, however, the model size is

$$\texttt{model\_size} \propto \texttt{fan\_in} * \texttt{fan\_out} * D.$$

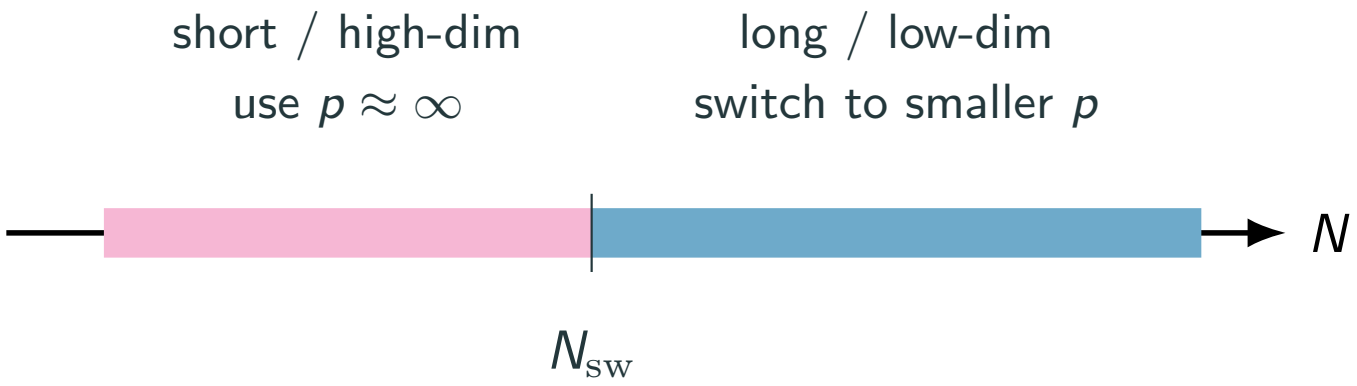
If width and depth are scaled together, then the rank of each matrix grows only like

$$\texttt{rank} \propto (\texttt{model\_size})^{1/3} \propto N^{1/3}.$$

As we scale up we shift further and further into the low-dim regime

Practical implications

Switching strategy



Switching point

$$N_{\mathrm{sw}} \asymp d \max\left\{1, \left(\frac{S}{A}\right)^{p_r}\right\}, \qquad A := L_\infty R_\infty^2, \quad S := \sigma_r R_\infty.$$

- Start close to Schatten-∞ when dimension price dominates.
- Switch to smallest admissible p_r once token budget dominates.

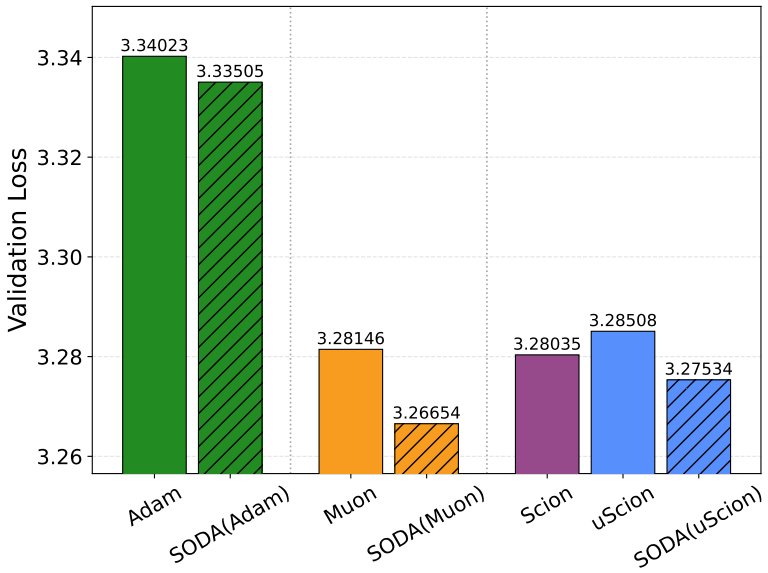
Remarkably, Nilin Abrahamsen arrived at such a schedule empirically for Soft-Muon [Abr26].

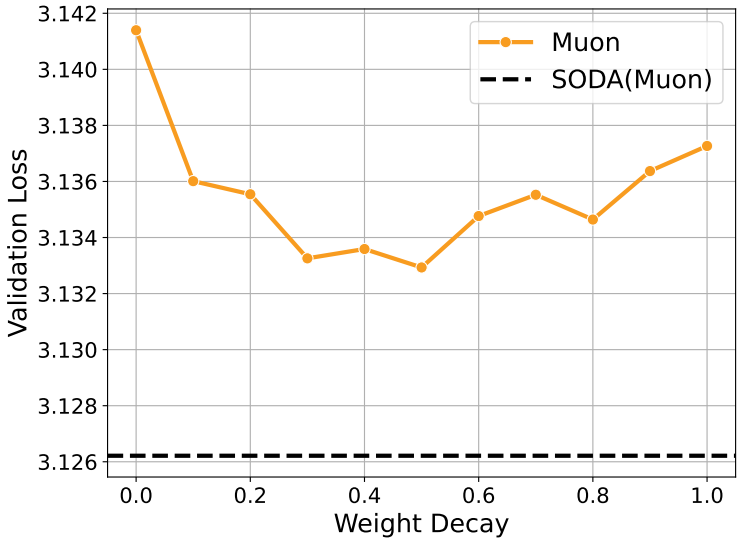
Anchoring and weight decay schedule

SODA for $h(x) = \frac{1}{p}\left\|x - z^0\right\|_p^p$

$$x^{k+1} = (1 - \lambda_k)x^k + \lambda_k z^0 - \lambda_k \gamma_k^{q-1} \nabla\left(\frac{1}{q}\|\cdot\|_q^q\right)(\bar{m}^{k+1}).$$

Theory applied to practice [Pet+26]:

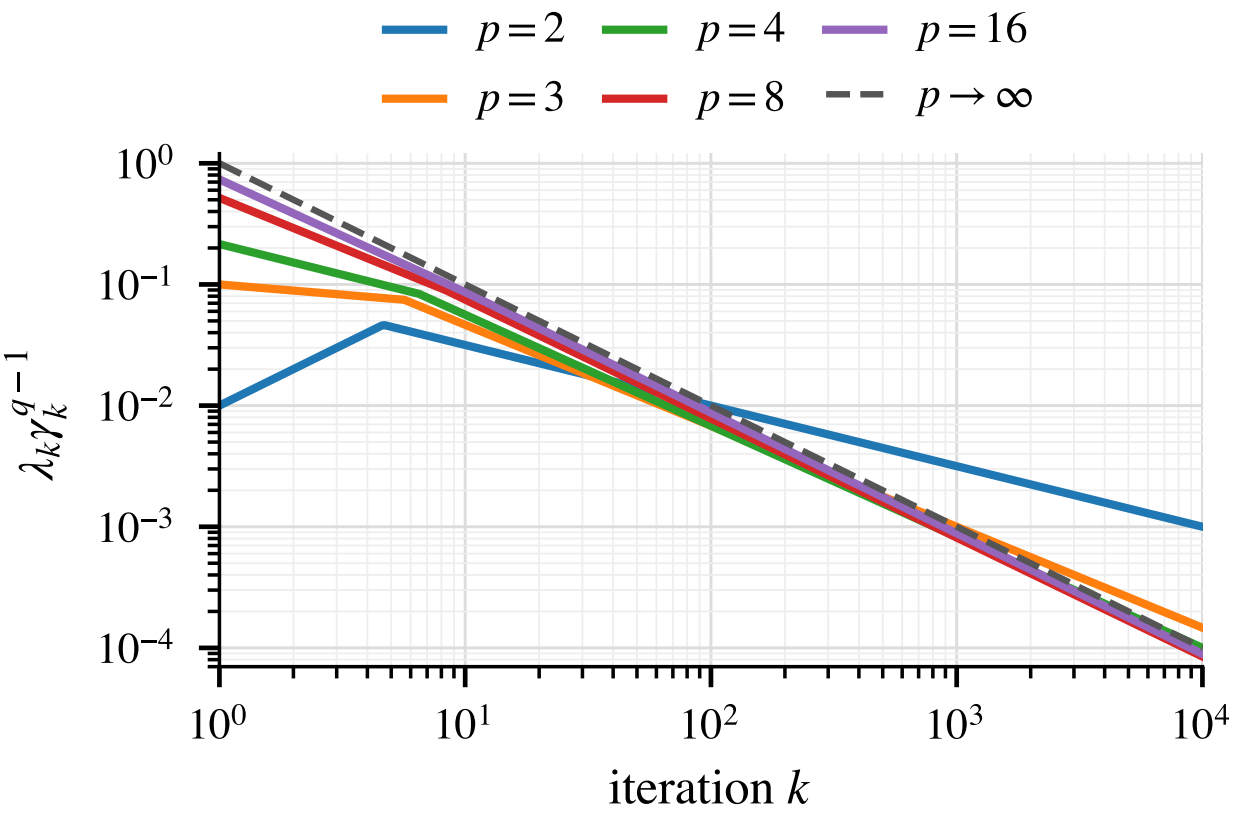




ADANA arrives at a similar schedule [Fer+26].

Warmup depends on geometry

Also suggests warmup for Euclidean but not Schatten-∞ (Muon):



Summary

Optimal choice of p is regime dependent!

Implications for HTMuon/Freon/Soft-Muon:

- when to use: low-dimensional setting with heavy-tailed noise
- how to set batch size, weight decay, stepsize ...

Thank you!

Why do we care about Schatten-p norms?

Muon as Optimistic Dual Averaging

Analysis

Regimes

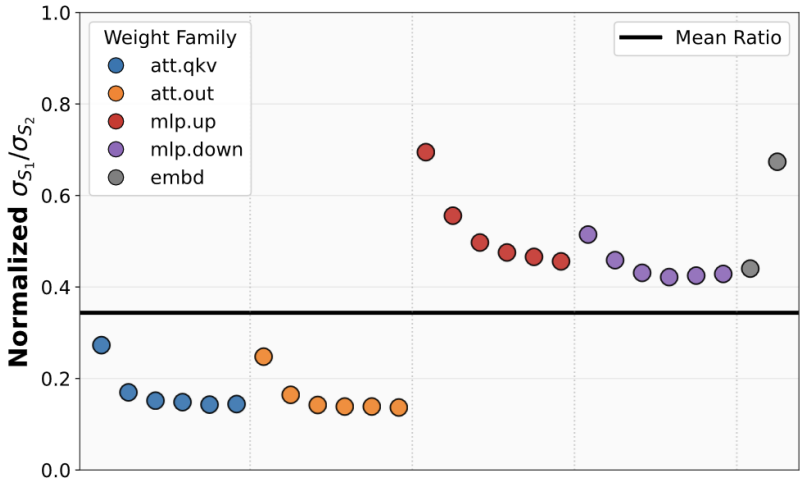
Practical implications

Appendix

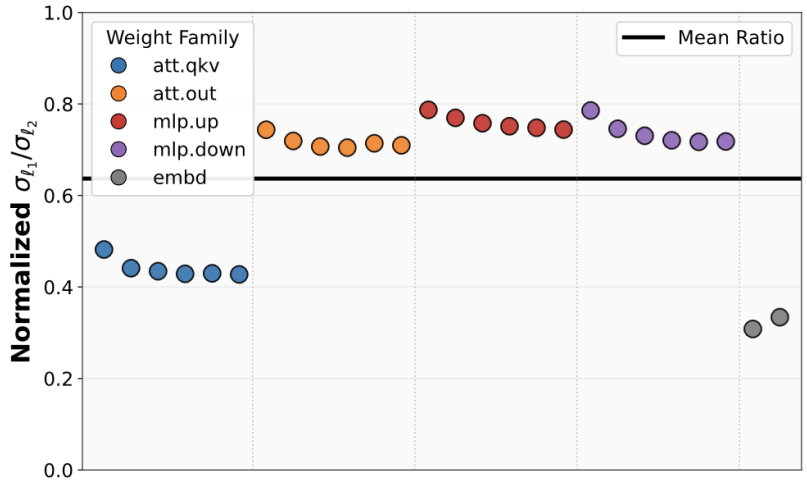
Appendix: A confession

We've been cheating:

- By $p \to \infty$ we refer to our $p = \log(d)$ result (introduces log factor)
- The true endpoint $p = \infty$ we treat in [HPS26] (incl. lower bounds)
- Also includes indication for why $\ell_\infty > S_\infty$ for input/ouput layers:



(a) Ratios in S_1.



(b) Ratios in ℓ_1.

Appendix: uniform convexity and smoothness

p-uniform convexity

A convex function h is p-uniformly convex with constant $\mu > 0$ if

$$h(y) \geq h(x) + \langle g, y - x \rangle + \frac{\mu}{p} \|y - x\|^p \qquad \forall x, y, \quad g \in \partial h(x).$$

q-uniform smoothness

A differentiable function ϕ is q-uniformly smooth with constant $L > 0$ if

$$\phi(y) \leq \phi(x) + \langle \nabla \phi(x), y - x \rangle + \frac{L}{q} \|y - x\|_*^q \qquad \forall x, y.$$

If $q = p/(p-1)$, then p-uniform convexity of h implies q-uniform smoothness of h^*.

Appendix: when is spectral worth it?

Spectral solvers such as Newton-Schulz compute:

$$U \operatorname{diag}(\sigma) V^\top \mapsto U V^\top.$$

For `batchsize=1`, spectral descent $\equiv$ SGD:

- MLP gradients are rank one.
- All Schatten norms are equivalent.
- The spectral solver just adds overhead.

Appendix: two efficiency regimes

- **Depth dominated**: parallelize the solver across many layers and GPUs.
- **Batch dominated**: large micro-batch size and sequence length make forward/backward passes expensive enough to amortize matrix sign.

Spectral methods pay off when the matrix-sign cost is hidden by depth or batch compute.

References i

[Abr26] Nilin Abrahamsen. **Contra-Muon and Soft-Muon.** Blog post. Accessed: 2026-05-27. 2026. URL: https://nilin.github.io/contra-muon-and-soft-muon/.

[BN24] Jeremy Bernstein and Laker Newhouse. **"Modular Duality in Deep Learning".** In: *arXiv preprint arXiv:2410.21265* (2024).

[Car+15] David Carlson et al. **"Stochastic spectral descent for discrete graphical models".** In: *IEEE Journal of Selected Topics in Signal Processing* 10.2 (2015), pp. 296–311.

[Che+23] Xiangning Chen et al. **"Symbolic Discovery of Optimization Algorithms".** In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=ne6zeqLFCZ.

References ii

[Def+24] Aaron Defazio et al. **The Road Less Scheduled.** 2024. arXiv: 2405.15682 [cs.LG]. URL: https://arxiv.org/abs/2405.15682.

[Fer+26] Damien Ferbach et al. **"Logarithmic-time Schedules for Scaling Language Models with Momentum".** In: *arXiv preprint arXiv:2602.05298* (2026).

[Hof+22] Jordan Hoffmann et al. **"Training compute-optimal large language models".** In: *Advances in Neural Information Processing Systems*. 2022.

[HPS26] Florian Hübler, Thomas Pethick, and Suvrit Sra. **"Free Heavy-Tailed Lunch for Muon: A Theoretical Justification of Empirical Success".** In: *arXiv preprint arXiv:2606.14560* (2026).

References iii

[Isl+26] Rustem Islamov et al. **"On the Role of Batch Size in Stochastic Conditional Gradient Methods".** In: *arXiv preprint arXiv:2603.21191* (2026).

[Jor+24a] Keller Jordan et al. **modded-nanogpt: Speedrunning the NanoGPT baseline.** 2024. URL: https://github.com/KellerJordan/modded-nanogpt.

[Jor+24b] Keller Jordan et al. **Muon: An optimizer for hidden layers in neural networks.** 2024. URL: https://kellerjordan.github.io/posts/muon/.

[Jor24] Keller Jordan. **"94% on CIFAR-10 in 3.29 Seconds on a Single GPU".** In: *arXiv preprint arXiv:2404.00498* (2024).

References iv

[Jou+20] Pooria Joulani et al. **"A simpler approach to accelerated optimization: iterative averaging meets optimism".** In: *International conference on machine learning*. PMLR. 2020, pp. 4984–4993.

[Mar+25] Martin Marek et al. **"Small batch size training for language models: When vanilla sgd works, and why gradient accumulation is wasteful".** In: *arXiv preprint arXiv:2507.07101* (2025).

[NS15] Yu Nesterov and Vladimir Shikhman. **"Quasi-monotone subgradient methods for nonsmooth convex minimization".** In: *Journal of Optimization Theory and Applications* 165.3 (2015), pp. 917–940.

References v

[Pan+26] Tianyu Pang et al. **"HTmuon: Improving muon via heavy-tailed spectral correction".** In: *arXiv preprint arXiv:2603.10067* (2026).

[Pet+25] Thomas Pethick et al. **"Training Deep Learning Models with Norm-Constrained LMOs".** In: *International Conference on Machine Learning*. 2025.

[Pet+26] Thomas Pethick et al. **"Optimistic Dual Averaging Unifies Modern Optimizers".** In: *arXiv preprint arXiv:2605.11172* (2026).

[SGO25] Teodora Srećković, Jonas Geiping, and Antonio Orvieto. **"Is your batch size the problem? Revisiting the Adam-SGD gap in language modeling".** In: *arXiv preprint arXiv:2506.12543* (2025).

References vi

[Shu+26] Zakhar Shumaylov et al. **"Muon is Not That Special: Random or Inverted Spectra Work Just as Well".** In: *arXiv preprint arXiv:2605.11181* (2026).

[SPJ25] Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. **"Benchmarking optimizers for large language model pretraining".** In: *arXiv preprint arXiv:2509.01440* (2025).

[Wen+25] Kaiyue Wen et al. **"Fantastic pretraining optimizers and where to find them".** In: *arXiv preprint arXiv:2509.02046* (2025).