Two Perspectives on Muon for Deep Learning Optimization: Preconditioning and Isotropic Curvature

Weijie Su

The University of Pennsylvania

The role of optimization toward AGI

- Optimization methods are the cornerstone for the success of modern large-scale AI (Bottou et al., 2018)
- Pre-training of SOTA base models costs $\gg$ hundreds of millions USD
- Scaling bottlenecks may stem from limitations in
  - Data? ("fossil fuel" of AI, yet there is only one Internet)
  - GPU compute? (Moore's law is no longer valid?)
  - Architecture? (Transformer and attention mechanism)

The role of optimization toward AGI

- Optimization methods are the cornerstone for the success of modern large-scale AI (Bottou et al., 2018)
- Pre-training of SOTA base models costs $\gg$ hundreds of millions USD
- Scaling bottlenecks may stem from limitations in
  - Data? ("fossil fuel" of AI, yet there is only one Internet)
  - GPU compute? (Moore's law is no longer valid?)
  - Architecture? (Transformer and attention mechanism)

What about optimizers (training algorithms)?

Which optimizer do you use for deep learning?

Adam(W) is the default one

```
model = NeuralNet() # torch.nn.Module
params = model.parameters()

optimizer = torch.optim.AdamW(params, lr=0.001, betas=(0.9, 0.999), eps=1e-08,

weight_decay=0.01)
```

Which optimizer do you use for deep learning?

Adam(W) is the default one

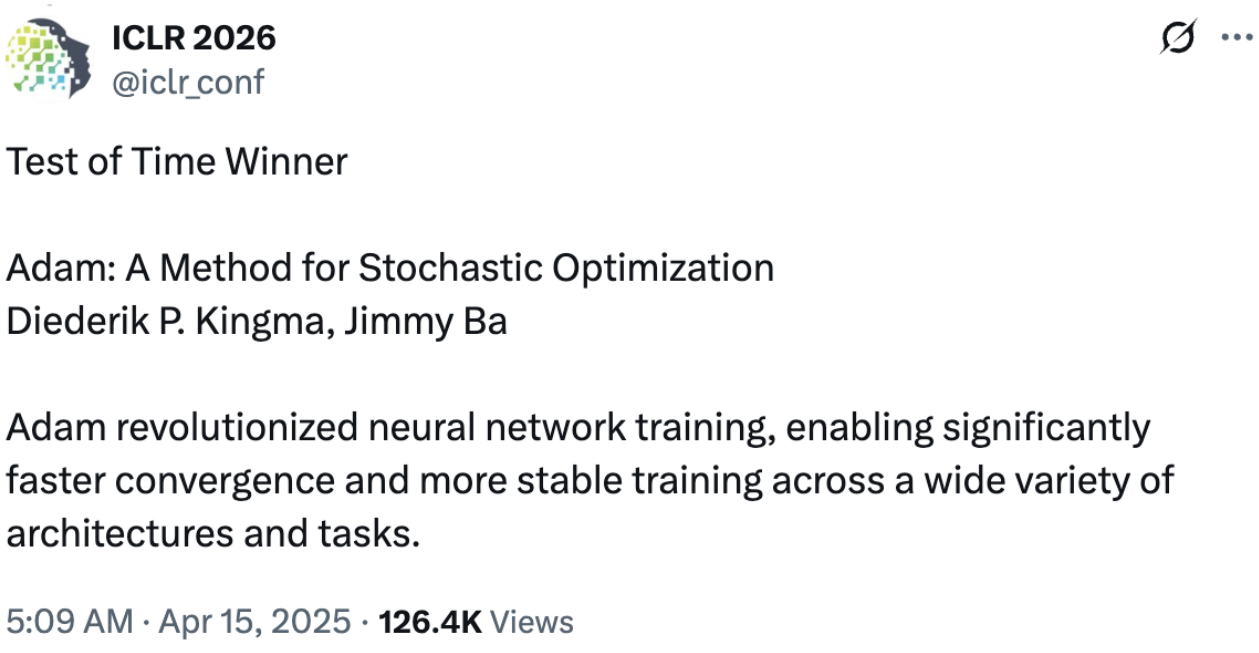
```
model = NeuralNet() # torch.nn.Module
params = model.parameters()

optimizer = torch.optim.AdamW(params, lr=0.001, betas=(0.9, 0.999), eps=1e-08,

weight_decay=0.01)
```

Update rules of Adam(W) (Kingma and Ba, 2015; Loshchilov and Hutter, 2019):

$$
\begin{aligned}
m_k &= \beta_1 m_{k-1} + (1-\beta_1) g_k, \quad g_k = \nabla f(w_k) \\
v_k &= \beta_2 v_{k-1} + (1-\beta_2) g_k^{\odot 2} \\
\widehat{m}_k &= m_k / (1-\beta_1^k) \\
\widehat{v}_k &= v_k / (1-\beta_2^k) \\
w_{k+1} &= (1-\lambda\gamma_k) w_k - \gamma_k \frac{\widehat{m}_k}{\sqrt{\widehat{v}_k} + \varepsilon}
\end{aligned}
$$

The impact of Adam

TITLE	CITED BY	YEAR
Adam: A method for stochastic optimization DP Kingma, J Ba arXiv preprint arXiv:1412.6980	222340	2014



Adam as a vector preconditioned gradient method

- Adam's update rule:

$$m_k = \beta_1 m_{k-1} + (1-\beta_1) g_k \qquad \widehat{m}_k = m_k/(1-\beta_1^k)$$
$$v_k = \beta_2 v_{k-1} + (1-\beta_2) g_k^{\odot 2} \qquad \widehat{v}_k = v_k/(1-\beta_2^k)$$
$$w_{k+1} = w_k - \gamma_k \widehat{m}_k / \left(\sqrt{\widehat{v}_k} + \varepsilon\right)$$

- Define a diagonal preconditioner $P_k = \mathrm{Diag}\left(\sqrt{\widehat{v}_k} + \varepsilon\right)$ and denote $\|w\|_P := \sqrt{\langle w, Pw\rangle}$

$$w_{k+1} = \mathrm{argmin}_{w\in\mathbb{R}^d} \left\{ \langle \widehat{m}_k, w - w_k\rangle + \frac{1}{2\gamma_k} \|w - w_k\|_{P_k}^2 \right\} = w_k - \gamma_k P_k^{-1} \widehat{m}_k$$

- Both $\widehat{m}_k$ and P_k are functions of g_k; in Gauss–Newton, $\nabla^2 f(w_k) \approx g_k g_k^\top$
- Majorization-minimization methods? Unclear if $\nabla^2 f(w_k) \preccurlyeq P_k$ always holds (unlikely here; P_k is just a Hessian approximation)

Adam as (smoothed) *normalized* steepest descent w.r.t. ℓ_∞-norm

- signSGD (i.e., Adam with $\beta_1 = \beta_2 = 0$; Bernstein et al., 2018):

$$w_{k+1} = w_k - \gamma_k \cdot \operatorname{sgn}(g_k),$$

  i.e., normalized steepest descent w.r.t. (squared) ℓ_∞-norm: $\|w\|_\infty = \max_{1 \le i \le d} |w_i|$

- Unnormalized steepest descent w.r.t. (squared) ℓ_∞-norm (with dual norm scaling):

$$\begin{aligned} w_{k+1} &= \operatorname{argmin}_{w \in \mathbb{R}^d} \left\{ \langle g_k, w - w_k \rangle + \frac{1}{2\gamma_k} \|w - w_k\|_\infty^2 \right\} \\ &= w_k - \gamma_k \cdot \|g_k\|_1 \operatorname{sgn}(g_k) \end{aligned}$$

Parameters are matrices instead of vectors

Multilayer perceptron for classification:

$$\mathrm{Prob}(x; W) = \mathrm{softmax}(W^L \sigma\left(W^{L-1}\sigma(\cdots \sigma(W^1 x)\cdots)\right))$$

Parameters are matrices instead of vectors

Multilayer perceptron for classification:

$$\mathrm{Prob}(x; W) = \mathrm{softmax}(W^L \sigma\left(W^{L-1}\sigma(\cdots \sigma(W^1 x)\cdots)\right))$$

- $\sigma(\cdot)$ nonlinear activation; e.g., $\sigma(x) = \mathrm{ReLu}(x) = \max\{x, 0\}$; bias omitted
- The size of W^i is

  #neurons in previous layer $\times$ #neurons in next layer

- Parameters are predominantly matrices in all architectures (such as QKV in Transformer)

Empirical evidence: The AlgoPerf (training algorithms competition)

- The AlgoPerf: Training algorithms competition (Dahl et al., 2023; Kasimbeg et al., 2025) aims at evaluating practical speed-ups in neural network training achieved solely by improving the underlying training algorithm
- Winner: Distributed Shampoo (Shi et al., 2023), an optimizer that does *not* treat gradients as vectors

Empirical evidence: The AlgoPerf (training algorithms competition)

- The AlgoPerf: Training algorithms competition (Dahl et al., 2023; Kasimbeg et al., 2025) aims at evaluating practical speed-ups in neural network training achieved solely by improving the underlying training algorithm
- Winner: Distributed Shampoo (Shi et al., 2023), an optimizer that does *not* treat gradients as vectors

Results challenge belief in Adam's optimality for deep learning

A new optimizer in 2024

Keller Jordan blog

Muon: An optimizer for hidden layers in neural networks

December 8, 2024 · 16 min

Muon is an optimizer for the hidden layers in neural networks. It is used in the current training speed records for both NanoGPT and CIFAR-10 speedrunning.

A new optimizer in 2024

Keller Jordan blog

Muon: An optimizer for hidden layers in neural networks

December 8, 2024 · 16 min

Muon is an optimizer for the hidden layers in neural networks. It is used in the current training speed records for both NanoGPT and CIFAR-10 speedrunning.

Algorithm 2 Muon

Require: Learning rate η, momentum μ

1: Initialize $B_0 \leftarrow 0$
2: **for** $t = 1, \ldots$ **do**
3: Compute gradient $G_t \leftarrow \nabla_\theta \mathcal{L}_t(\theta_{t-1})$
4: $B_t \leftarrow \mu B_{t-1} + G_t$
5: $O_t \leftarrow \text{NewtonSchulz5}(B_t)$
6: Update parameters $\theta_t \leftarrow \theta_{t-1} - \eta O_t$
7: **end for**
8: **return** θ_t

A new optimizer in 2024

Keller Jordan blog

Muon: An optimizer for hidden layers in neural networks

December 8, 2024 · 16 min

Muon is an optimizer for the hidden layers in neural networks. It is used in the current training speed records for both NanoGPT and CIFAR-10 speedrunning.

- As a blog post, not published
- Keller Jordan was invited to join OpenAI

Algorithm 2 Muon

Require: Learning rate η, momentum μ

Initialize $B_0 \leftarrow 0$
for $t = 1, \ldots$ **do**
  Compute gradient $G_t \leftarrow \nabla_\theta \mathcal{L}_t(\theta_{t-1})$
  $B_t \leftarrow \mu B_{t-1} + G_t$
  $O_t \leftarrow \text{NewtonSchulz5}(B_t)$
  Update parameters $\theta_t \leftarrow \theta_{t-1} - \eta O_t$
end for
return θ_t

The Muon optimizer (Jordan et al., 2024)

Muon

- Let $G_k = \nabla f(W_k) = U\Sigma V^\top \in \mathbb{R}^{m\times n}$ be the SVD
- The new update:

$$W_{k+1} = W_k - \gamma_k UV^\top$$

The Muon optimizer (Jordan et al., 2024)

Muon

- Let $G_k = \nabla f(W_k) = U\Sigma V^\top \in \mathbb{R}^{m\times n}$ be the SVD
- The new update:

$$W_{k+1} = W_k - \gamma_k UV^\top$$

- In practice, momentum is used: $M_k = \beta M_{k-1} + (1-\beta)G_k$
- Earlier ideas: Carlson et al. (2016); Gupta et al. (2018); Vyas et al. (2025)

The Muon optimizer (Jordan et al., 2024)

Muon

- Let $G_k = \nabla f(W_k) = U\Sigma V^\top \in \mathbb{R}^{m\times n}$ be the SVD
- The new update:
$$W_{k+1} = W_k - \gamma_k UV^\top$$
- In practice, momentum is used: $M_k = \beta M_{k-1} + (1-\beta)G_k$
- Earlier ideas: Carlson et al. (2016); Gupta et al. (2018); Vyas et al. (2025)

- $UV^\top$ is the projection of $\nabla f(W_k)$ onto the semi-orthogonal space $O^{m\times n} = \{A \in \mathbb{R}^{m\times n} : AA^\top = I_m \text{ or } A^\top A = I_n\}$
- $\mathrm{msgn}(X) := UV^\top$ extends matrix sign function for (symmetric) square matrices: let $X = U\Sigma U^\top$, then $\mathrm{msgn}(X)$ coincides with $U\mathrm{sgn}(\Sigma)U^\top$

Finding $\mathrm{msgn}(X)$ via Newton–Schulz

To be GPU-friendly, Jordan et al. (2024) used polynomials to approximate $\mathrm{msgn}(X)$

- $\mathrm{polar}(X) - p(X) = U(I - p(\Sigma))V^\top$, for odd polynomial

$$p(X) = a_0 X + a_1 X(X^\top X) + \cdots + a_q X(X^\top X)^q$$

- Example: $p(x) = \frac{15}{8}x - \frac{10}{8}x^3 + \frac{3}{8}x^5$
- Amsel et al. (2025) suggested approximation via compositions of polynomials:

$$p^\star \;=\; \mathrm{argmin}_{p \in \mathbb{P}_d^{odd}} \max_{x \in [\ell, u]} |1 - p(x)|$$

Finding $\mathrm{msgn}(X)$ via Newton–Schulz

To be GPU-friendly, Jordan et al. (2024) used polynomials to approximate $\mathrm{msgn}(X)$

- $\mathrm{polar}(X) - p(X) = U(I - p(\Sigma))V^\top$, for odd polynomial

$$p(X) = a_0 X + a_1 X(X^\top X) + \cdots + a_q X(X^\top X)^q$$

- Example: $p(x) = \frac{15}{8}x - \frac{10}{8}x^3 + \frac{3}{8}x^5$
- Amsel et al. (2025) suggested approximation via compositions of polynomials:

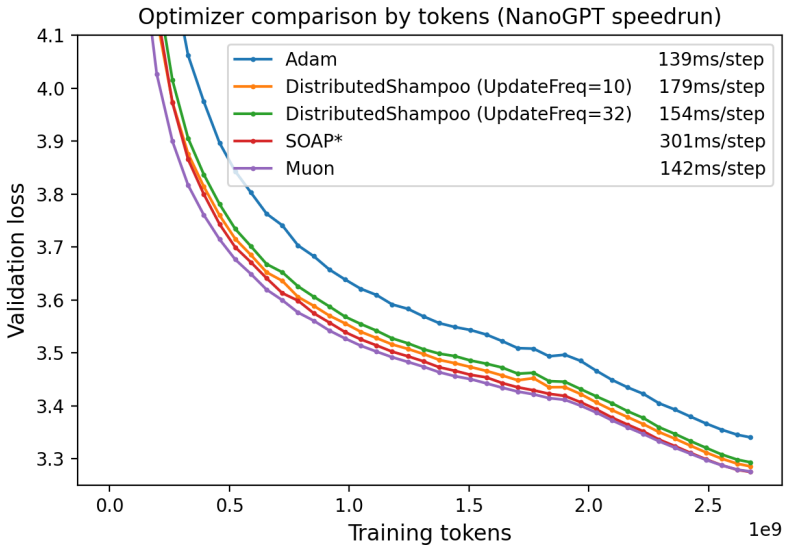
$$p^\star \;=\; \mathrm{argmin}_{p \in \mathbb{P}_d^{odd}} \max_{x \in [\ell, u]} |1 - p(x)|$$

- Weirdly, more accurate approximation doesn't yield better LLM optimization

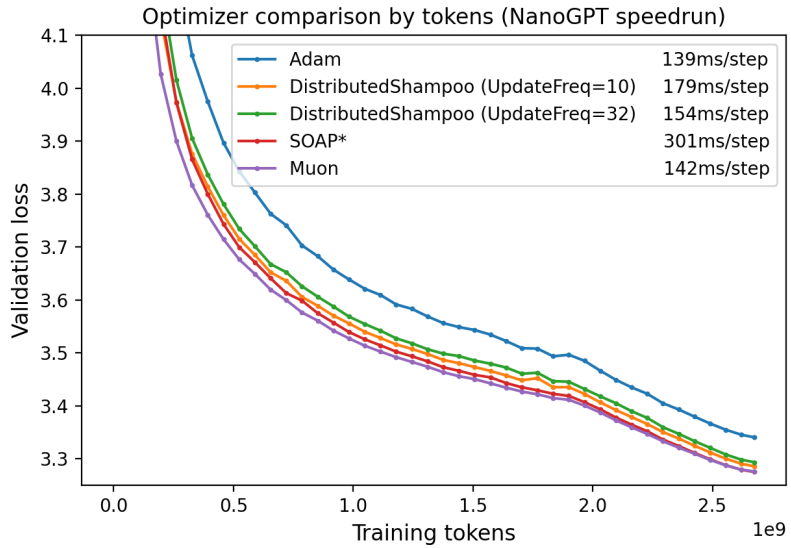
Early promising experimental results (Jordan et al., 2024)

Why it's difficult to beat Adam

Neural architectures (like Transformer) may be "overfitted" to Adam's optimization characteristics (Orabona, 2020)



*SOAP is under active development. Future versions will significantly improve the wallclock overhead.

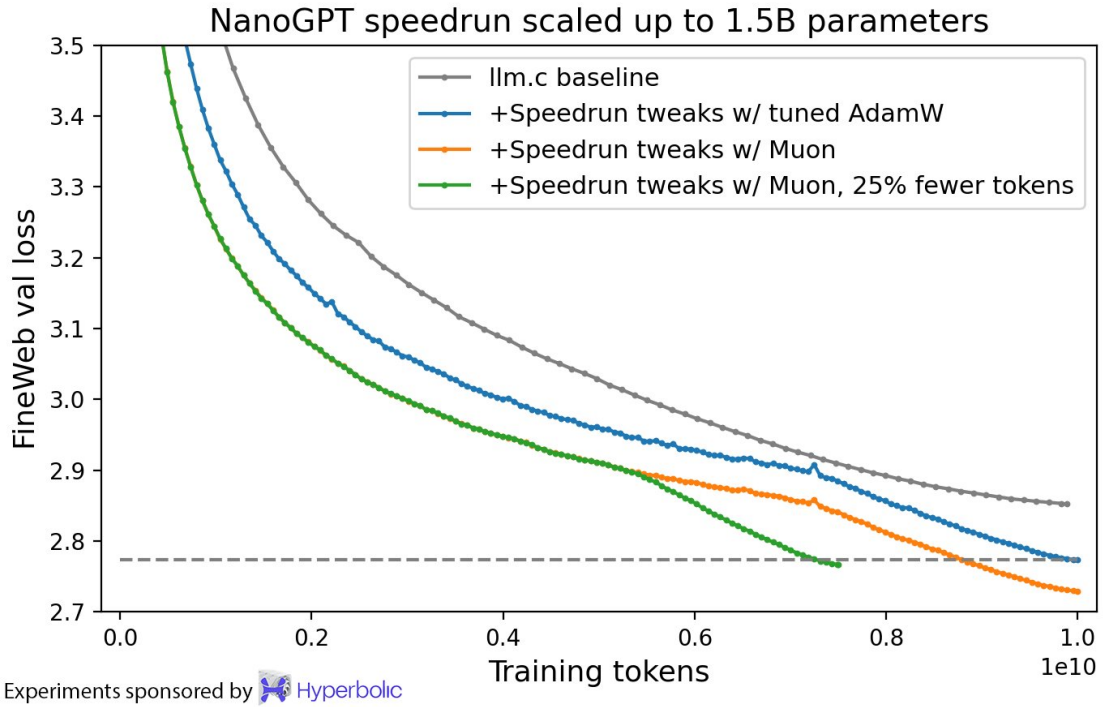


*SOAP is under active development. Future versions will significantly improve the wallclock overhead.

Early promising experimental results (Jordan et al., 2024)

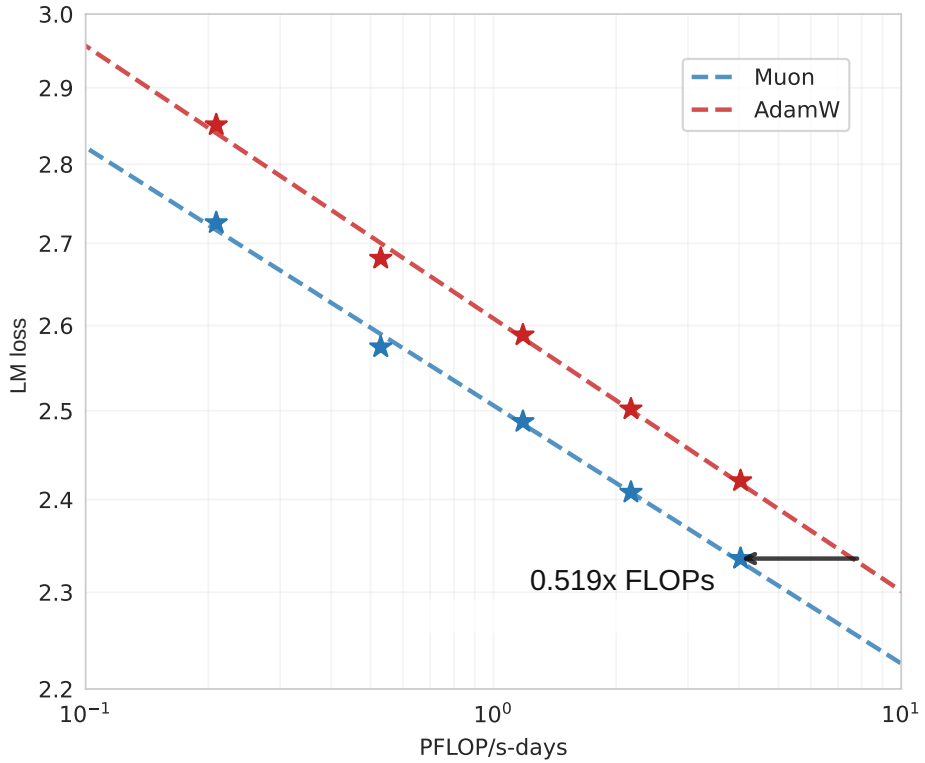
Why it's difficult to beat Adam

Neural architectures (like Transformer) may be "overfitted" to Adam's optimization characteristics (Orabona, 2020)



Superiority of Muon over Adam on 16B LLMs by Moonshot (Liu et al., 2025)

Muon is about $2\times$ more computationally efficient than Adam



Questions that this talk will address

- Why is orthogonalization good for deep learning optimization?
- Is (exact) orthogonalization really optimal?

Two perspectives on matrix-gradient methods

Outline

Collaborators

- Tim Tsz-Kit Lau (Penn/DRW)
- Qi Long (Penn)

However, curvature information is (completely) missing

Curvature information from singular values are completely gone

However, curvature information is (completely) missing

Curvature information from singular values are completely gone

- Moonshot's tuning parameters for Muon (Liu et al., 2025):

$$W_{k+1} = W_k - 0.2\gamma_k \sqrt{\max\{m, n\}} \cdot \mathrm{msgn}(G_k)$$

- $\{\gamma_k\}$ are pre-fixed
- Oscillate even if $G_k \to 0$

Definition (Null-gradient consistency)

An optimization algorithm exhibits null-gradient consistency if the magnitude of its update step tends to zero as the effective gradient term approaches zero

Curvature (partially) recovered via polar decomposition

- Polar decomposition: $U_{\mathsf{p}} H = \mathrm{polar}(X)$
  - $U_{\mathsf{p}} \in \mathbb{O}^{m \times n}$ has orthonormal columns
  - $H \in \mathbb{S}_+^n$ is a symmetric positive semidefinite matrix
- If $U \Sigma V^\top = \mathrm{SVD}(X)$, then $U_{\mathsf{p}} = U V^\top = \mathrm{msgn}(X)$ and $H = V \Sigma V^\top$
- H contains the curvature information

Polar gradient methods

- Partial curvature information can then be recovered since the nuclear norm is the sum of singular values
- Matrix sign descent with polar decomposition of gradient:

$$U\Sigma V^\top = \mathrm{SVD}(G) \Rightarrow \|\|G\|\|_{\mathrm{nuc}} = \mathrm{tr}(\Sigma)$$

$$\mathrm{tr}(H) = \mathrm{tr}(V\Sigma V^\top) = \mathrm{tr}(V^\top V\Sigma) = \mathrm{tr}(\Sigma)$$

Polar gradient methods

- Partial curvature information can then be recovered since the nuclear norm is the sum of singular values
- Matrix sign descent with polar decomposition of gradient:

$$U\Sigma V^\top = \mathrm{SVD}(G) \Rightarrow \|\!|G|\!\|_{\mathrm{nuc}} = \mathrm{tr}(\Sigma)$$

$$\mathrm{tr}(H) = \mathrm{tr}(V\Sigma V^\top) = \mathrm{tr}(V^\top V\Sigma) = \mathrm{tr}(\Sigma)$$

PolarGrad

$$U_k H_k = \mathrm{polar}(G_k),\ W_{k+1} = W_k - \gamma_k\,\mathrm{tr}(H_k)\,U_k$$

Polar gradient methods

- Partial curvature information can then be recovered since the nuclear norm is the sum of singular values
- Matrix sign descent with polar decomposition of gradient:

$$U\Sigma V^\top = \mathrm{SVD}(G) \Rightarrow \|\!|G|\!\|_{\mathrm{nuc}} = \mathrm{tr}(\Sigma)$$

$$\mathrm{tr}(H) = \mathrm{tr}(V\Sigma V^\top) = \mathrm{tr}(V^\top V\Sigma) = \mathrm{tr}(\Sigma)$$

PolarGrad

$$U_k H_k = \mathrm{polar}(G_k),\ W_{k+1} = W_k - \gamma_k\,\mathrm{tr}(H_k)\,U_k$$

- Step size/learning rate matters!

It's Muon with Armijo's backtracking line search

Seek a learning rate α_k such that:

$$f(X_k - \alpha_k U_k) \leq f(X_k) - c\alpha_k \langle G_k, U_k \rangle_{\mathrm{F}} = f(X_k) - c\alpha_k \|\|\|G_k\|\|\|_{\mathrm{nuc}}, \quad 0 < c < 1$$

It's Muon with Armijo's backtracking line search

Seek a learning rate α_k such that:

$$f(X_k - \alpha_k U_k) \leq f(X_k) - c\alpha_k \langle G_k, U_k \rangle_{\mathrm{F}} = f(X_k) - c\alpha_k |||G_k|||_{\mathrm{nuc}}, \quad 0 < c < 1$$

If f is L-Lipschitz smooth, then

$$f(X_k - \alpha_k U_k) \leq f(X_k) - \alpha_k |||G_k|||_{\mathrm{nuc}} + \frac{L}{2}\alpha_k^2 r_k$$

Hence, the learning rate satisfies

$$\alpha_k \leq \frac{2(1-c)}{L r_k} |||G_k|||_{\mathrm{nuc}}$$

- Backtracking keeps $\alpha_k / |||G_k|||_{\mathrm{nuc}}$ in a stable range
- Implicitly recovers the nuclear norm scaling term

It's Muon with Armijo's backtracking line search

Seek a learning rate α_k such that:

$$f(X_k - \alpha_k U_k) \leq f(X_k) - c\alpha_k \langle G_k, U_k \rangle_{\mathrm{F}} = f(X_k) - c\alpha_k |||G_k|||_{\mathrm{nuc}}, \quad 0 < c < 1$$

If f is L-Lipschitz smooth, then

$$f(X_k - \alpha_k U_k) \leq f(X_k) - \alpha_k |||G_k|||_{\mathrm{nuc}} + \frac{L}{2}\alpha_k^2 r_k$$

Hence, the learning rate satisfies

$$\alpha_k \leq \frac{2(1-c)}{L r_k} |||G_k|||_{\mathrm{nuc}}$$

- Backtracking keeps $\alpha_k / |||G_k|||_{\mathrm{nuc}}$ in a stable range
- Implicitly recovers the nuclear norm scaling term

Why explicit scaling in PolarGrad?

- Backtracking is computationally expensive and rarely used in DL
- PolarGrad makes the scaling explicit: $\gamma_k \mathrm{tr}(H_k) \equiv \gamma_k |||G_k|||_{\mathrm{nuc}}$

PolarGrad as spectral-norm-regularized steepest descent

$$U_k H_k = \mathrm{polar}(G_k),\ W_{k+1} = W_k - \gamma_k \,\mathrm{tr}(H_k)\, U_k$$

$$W_k - \gamma_k \,\mathrm{tr}(H_k)\, U_k = \mathrm{argmin}_{W \in \mathbb{R}^{m\times n}} \left\{ \langle G_k, W - W_k \rangle_{\mathrm{F}} + \frac{1}{2\gamma_k} |||W - W_k|||_{\mathrm{S}}^2 \right\}$$

- Satisfies the null-gradient consistency
- Spectral norm is submultiplicative: $|||XY||| \leq |||X|||\,|||Y|||$, and any unitarily invariant matrix norm satisfies $|||X||| \geq |||X|||_{\mathrm{S}}$

PolarGrad as spectral-norm-regularized steepest descent

$$U_k H_k = \mathrm{polar}(G_k),\ W_{k+1} = W_k - \gamma_k \,\mathrm{tr}(H_k)\, U_k$$

$$W_k - \gamma_k \,\mathrm{tr}(H_k)\, U_k = \mathrm{argmin}_{W \in \mathbb{R}^{m\times n}} \left\{ \langle G_k, W - W_k \rangle_{\mathrm{F}} + \frac{1}{2\gamma_k} |||W - W_k|||_{\mathrm{S}}^2 \right\}$$

- Satisfies the null-gradient consistency
- Spectral norm is submultiplicative: $|||XY||| \le |||X|||\,|||Y|||$, and any unitarily invariant matrix norm satisfies $|||X||| \ge |||X|||_{\mathrm{S}}$

However, Adam's ($\beta_1 = \beta_2 = 0$) update rules for matrix parameters is

$$W_{k+1} \in \mathrm{argmin}_{W \in \mathbb{R}^{m\times n}} \left\{ \langle G_k, W - W_k \rangle_{\mathrm{F}} + \frac{1}{2\gamma_k} |||W - W_k|||_{\max}^2 \right\},$$

where $|||W|||_{\max} = \max\limits_{1\le i\le m, 1\le j\le n} |w_{i,j}|$ is not a submultiplicative matrix norm

Skip “convergence analysis”

Convergence analysis

Theorem *Suppose that* $f : \mathbb{R}^{m\times n} \to \overline{\mathbb{R}}$ *is* L*-Lipschitz smooth and* μ*-strongly convex, then with* $\gamma_k = 1/(Lr_k)$

$$f(W_{k+1}) - f^\star \leq \left(1 - 1/(r_k^2 \kappa_H)\right)(f(W_k) - f^\star),$$

$$f(W_{k+1}) - f^\star \leq \left(1 - 1/(\kappa_{G_k}^2 \kappa_H)\right)(f(W_k) - f^\star),$$

where $r_k := \operatorname{rank}(\nabla f(W_k))$, $\kappa_{G_k} := \sigma_{\max}(\nabla f(W_k))/\sigma_{\min}(\nabla f(W_k))$, $\kappa_H := L/\mu$

- Gradient-based rate can significantly outperform the Hessian-based rate when $\kappa_{G_k} \ll r_k$

Convergence analysis

Theorem *Suppose that* $f : \mathbb{R}^{m\times n} \to \overline{\mathbb{R}}$ *is* L*-Lipschitz smooth and* μ*-strongly convex, then with* $\gamma_k = 1/(Lr_k)$

$$f(W_{k+1}) - f^\star \leq \left(1 - 1/(r_k^2 \kappa_H)\right)(f(W_k) - f^\star),$$

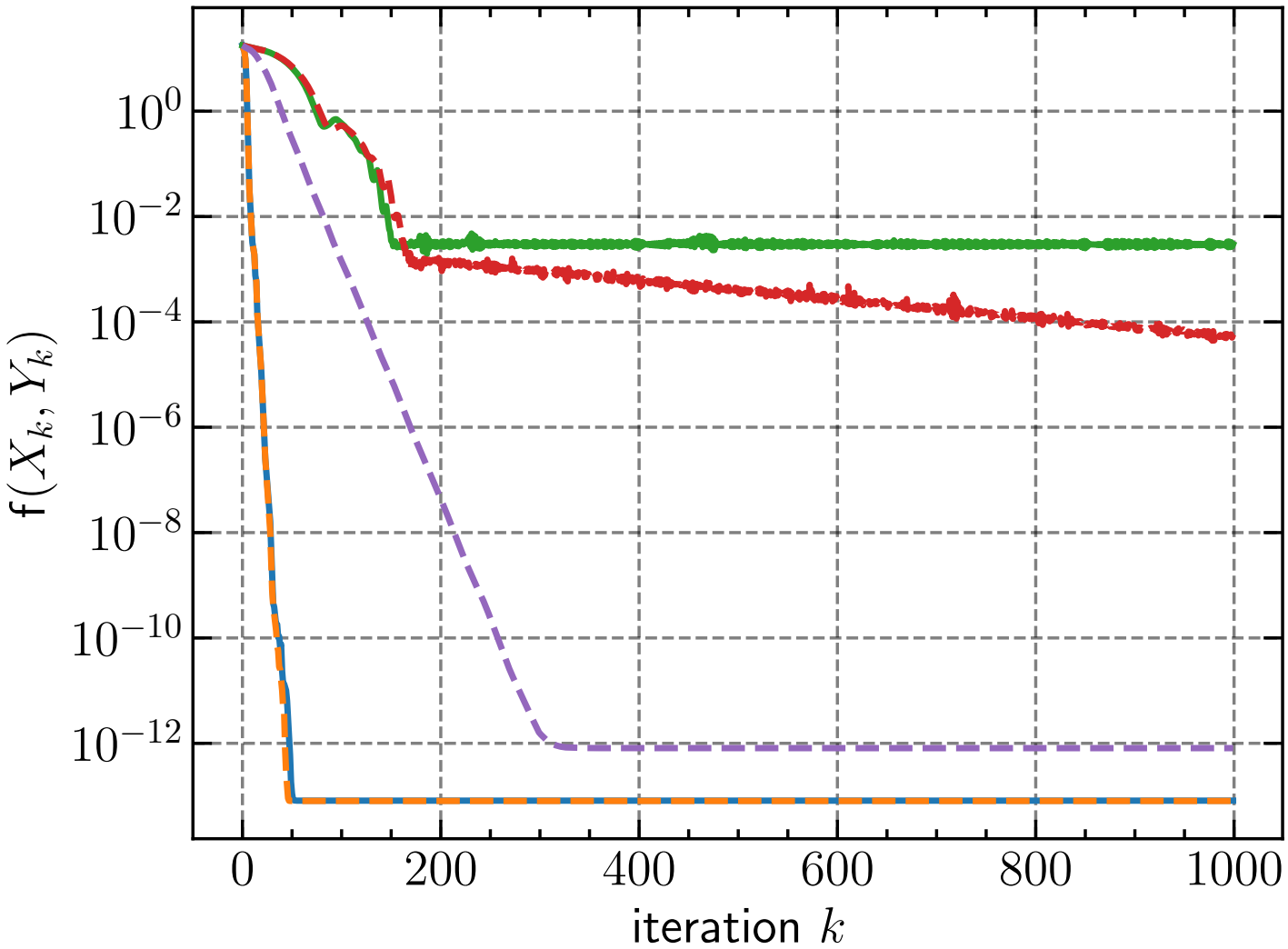
$$f(W_{k+1}) - f^\star \leq \left(1 - 1/(\kappa_{G_k}^2 \kappa_H)\right)(f(W_k) - f^\star),$$

where $r_k := \operatorname{rank}(\nabla f(W_k))$, $\kappa_{G_k} := \sigma_{\max}(\nabla f(W_k))/\sigma_{\min}(\nabla f(W_k))$, $\kappa_H := L/\mu$

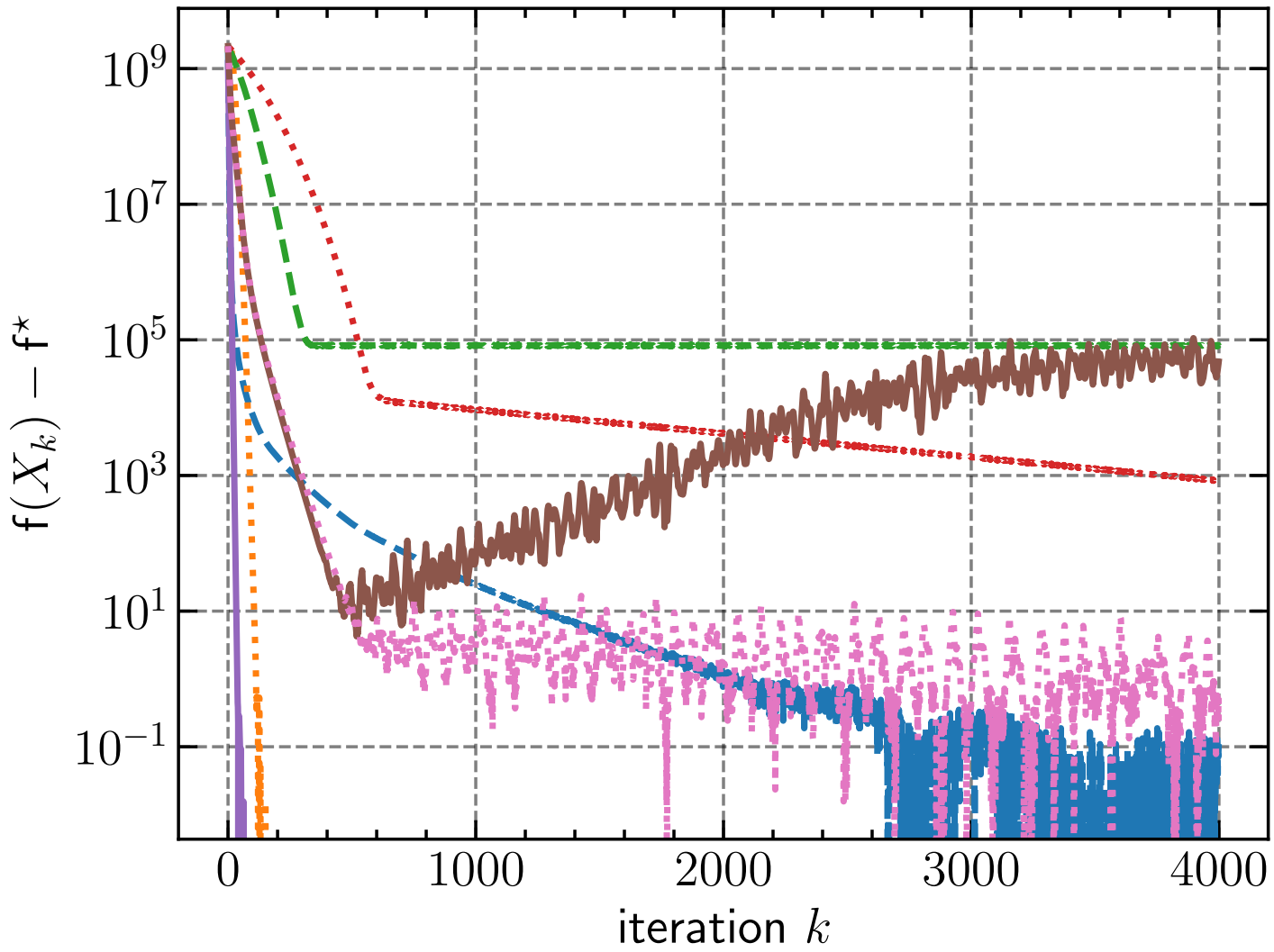
- Gradient-based rate can significantly outperform the Hessian-based rate when $\kappa_{G_k} \ll r_k$

Theorem *Assume unbiased stochastic gradients with bounded variance* $\varsigma^2 \in (0, \infty)$*, then* $f(W_K) - f^\star \leq O(1/K)$

Low-rank matrix completion
$(f(X,Y) = \|\|\|(XY^\top - M_\star)_{\text{obs}}\|\|\|_{\text{F}}^2)$



Matrix quadratic regression ($\mathsf{f}(X) = \sum \log(1 + \exp(-c_i \odot (a_i X B)))$)



Qwen2.5 pre-training

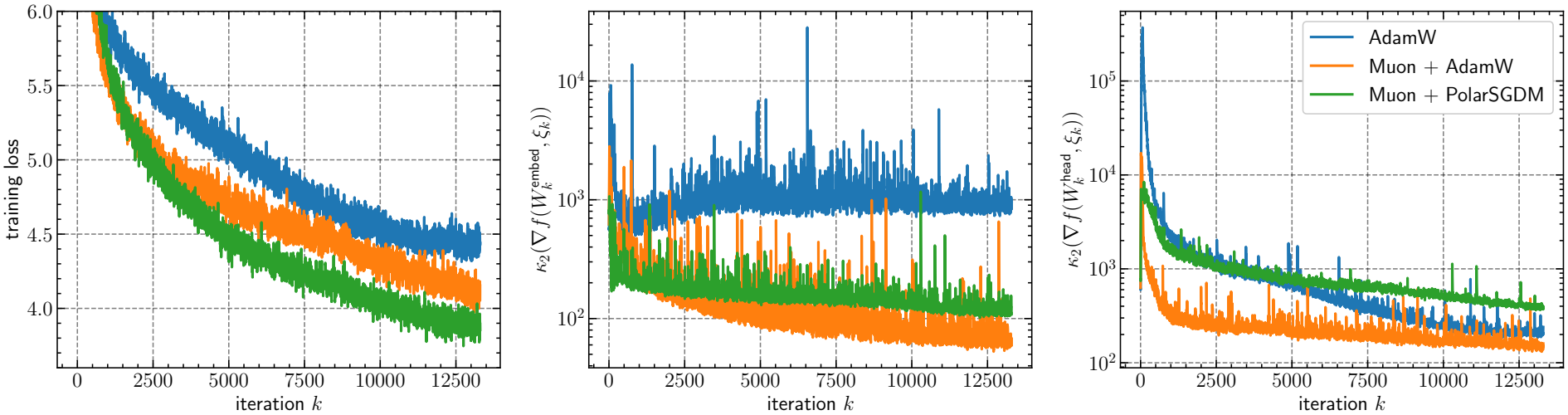


Figure: Adam(W)—Adam(W) for all parameters; Muon + Adam(W) (PolarSGDM)—Muon for hidden layers and Adam(W) (PolarSGDM) for embedding and head layers

Outline

Taylor expanding the loss

Total loss $f(W) = \frac{1}{N}\sum_{i=1}^{N} L(Wu_i)$

$$f(W+\delta W) = f(W) + \mathrm{Tr}(\delta W \nabla f(W)^\top) + \frac{1}{N}\sum_i (\delta W u_i)^\top \left[\int_0^1 (1-t)\nabla^2 L(Wu_i + t\delta W u_i)\mathrm{d}t\right] \delta W u_i$$

Taylor expanding the loss

Total loss $f(W) = \frac{1}{N}\sum_{i=1}^{N} L(Wu_i)$

$$f(W+\delta W) = f(W) + \mathrm{Tr}(\delta W \nabla f(W)^\top) + \frac{1}{N}\sum_i (\delta W u_i)^\top \left[\int_0^1 (1-t)\nabla^2 L(Wu_i + t\delta W u_i)\mathrm{d}t\right] \delta W u_i$$

All complexity arises from

$$\int_0^1 (1-t)\nabla^2 L(Wu_i + t\delta W u_i)\mathrm{d}t$$

- Weighted average of Hessians in the space
- Curvature (higher-order information)

Assumption on the curvature

Isotropy assumption

- Input data u_i's are isotropic, distributed over a sphere
- $\int_0^1 (1-t)\nabla^2 L(Wu_i + t\delta W u_i)\mathrm{d}t$ is isotropic, depending only on $\|\delta W u_i\|$

Rationale

- Deep learning/LLMs are massive systems

Assumption on the curvature

Isotropy assumption

- Input data u_i's are isotropic, distributed over a sphere
- $\int_0^1 (1-t)\nabla^2 L(Wu_i + t\delta W u_i)\mathrm{d}t$ is isotropic, depending only on $\|\delta W u_i\|$

Rationale

- Deep learning/LLMs are massive systems
- No specific directions are favored by design

Assumption on the curvature

Isotropy assumption

- Input data u_i's are isotropic, distributed over a sphere
- $\int_0^1 (1-t)\nabla^2 L(Wu_i + t\delta W u_i)\mathrm{d}t$ is isotropic, depending only on $\|\delta W u_i\|$

Rationale

- Deep learning/LLMs are massive systems
- No specific directions are favored by design
- Data universe may exhibit convergent laws

Assumption on the curvature

Isotropy assumption

- Input data u_i's are isotropic, distributed over a sphere
- $\int_0^1 (1-t)\nabla^2 L(Wu_i + t\delta W u_i)\mathrm{d}t$ is isotropic, depending only on $\|\delta W u_i\|$

Rationale

- Deep learning/LLMs are massive systems
- No specific directions are favored by design
- Data universe may exhibit convergent laws
- Decoupling first-order and higher-order information

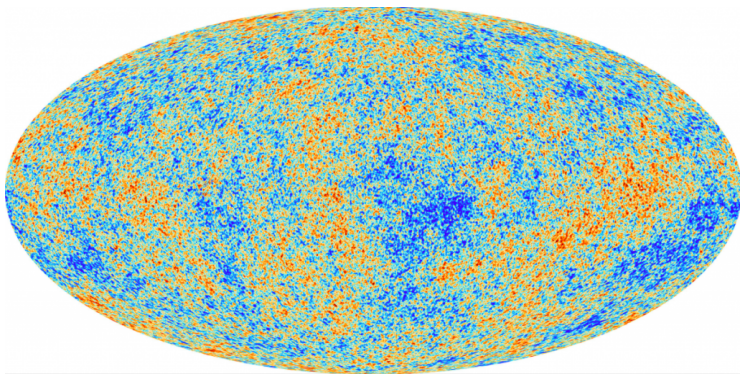
Assumption on the curvature

Isotropy assumption

- Input data u_i's are isotropic, distributed over a sphere
- $\int_0^1 (1-t)\nabla^2 L(Wu_i + t\delta W u_i)\mathrm{d}t$ is isotropic, depending only on $\|\delta W u_i\|$

Rationale

- Deep learning/LLMs are massive systems
- No specific directions are favored by design
- Data universe may exhibit convergent laws
- Decoupling first-order and higher-order information



Isotropic curvature model

Find update matrix $Q := -\delta W$ that minimizes

$$f(W - Q) \approx f(W) - \mathrm{Tr}(Q \nabla f(W)^\top) + \frac{1}{N} \sum_i H(\|Q u_i\|)$$

for some curvature function H. Write $G = \nabla f(W)$

Isotropic curvature model

Find update matrix $Q := -\delta W$ that minimizes

$$f(W - Q) \approx f(W) - \operatorname{Tr}(Q \nabla f(W)^\top) + \frac{1}{N} \sum_i H(\|Q u_i\|)$$

for some curvature function H. Write $G = \nabla f(W)$

Isotropic curvature model

$$\min_Q - \operatorname{Tr}(Q G^\top) + \mathbb{E}_{u \sim \text{sphere}} H(\|Q u\|)$$

Isotropic curvature model

Find update matrix $Q := -\delta W$ that minimizes

$$f(W - Q) \approx f(W) - \operatorname{Tr}(Q \nabla f(W)^\top) + \frac{1}{N} \sum_i H(\|Q u_i\|)$$

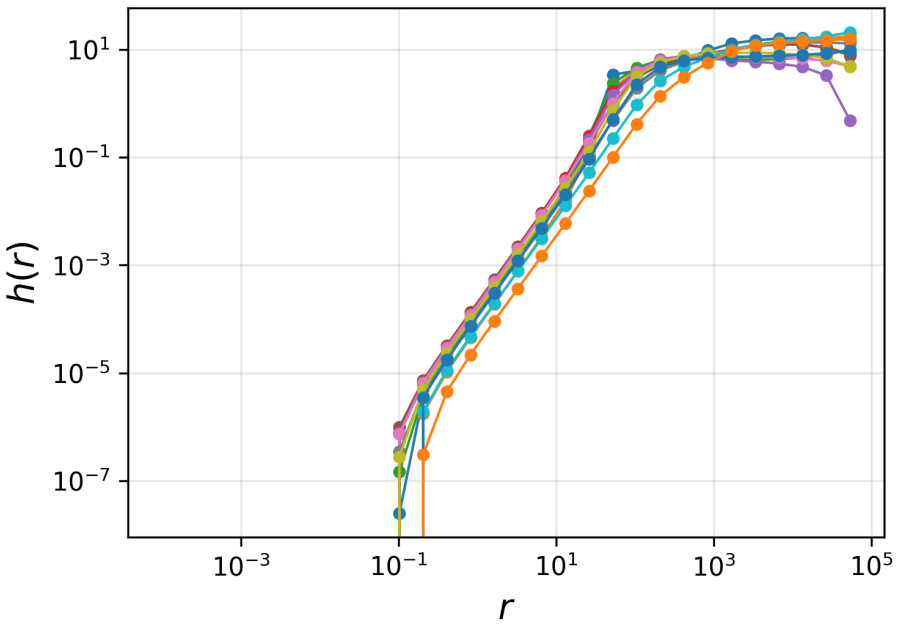
for some curvature function H. Write $G = \nabla f(W)$

Isotropic curvature model

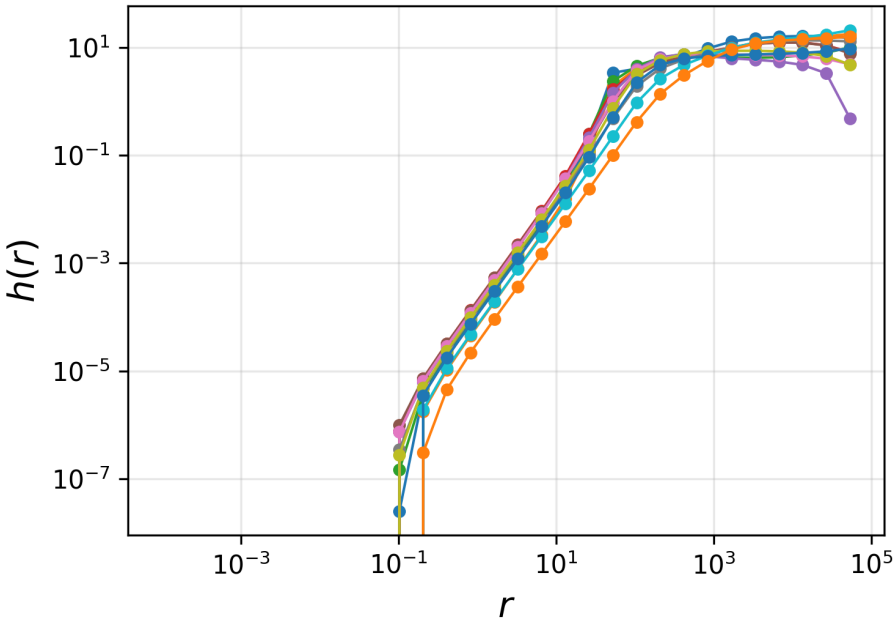
$$\min_Q - \operatorname{Tr}(Q G^\top) + \mathbb{E}_{u \sim \text{sphere}} H(\|Q u\|)$$

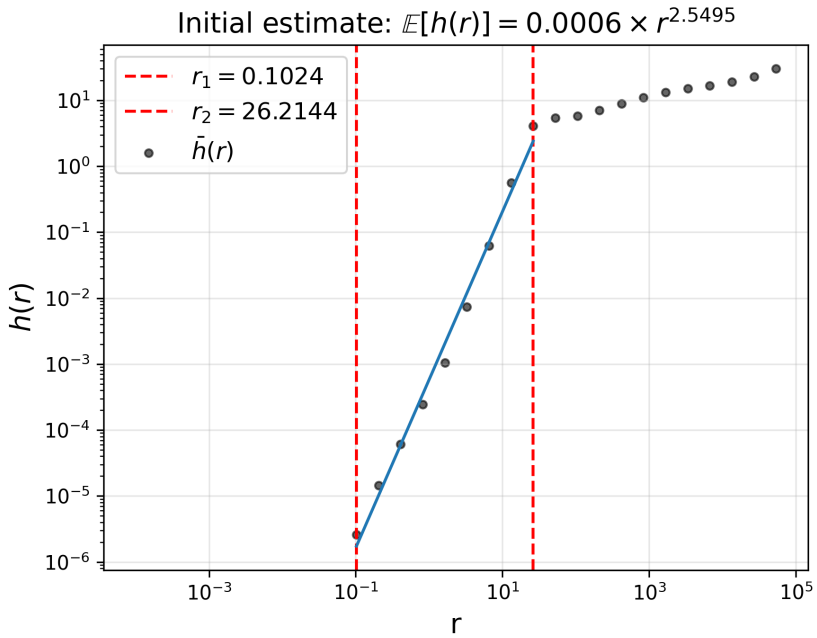
- The model considers one iteration, thereby adopting a greedy approach
- Learning rate has been incorporated
- H grows rapidly (large Lipschitz smoothness constant)

Empirical evidence: How curvature depends on radius

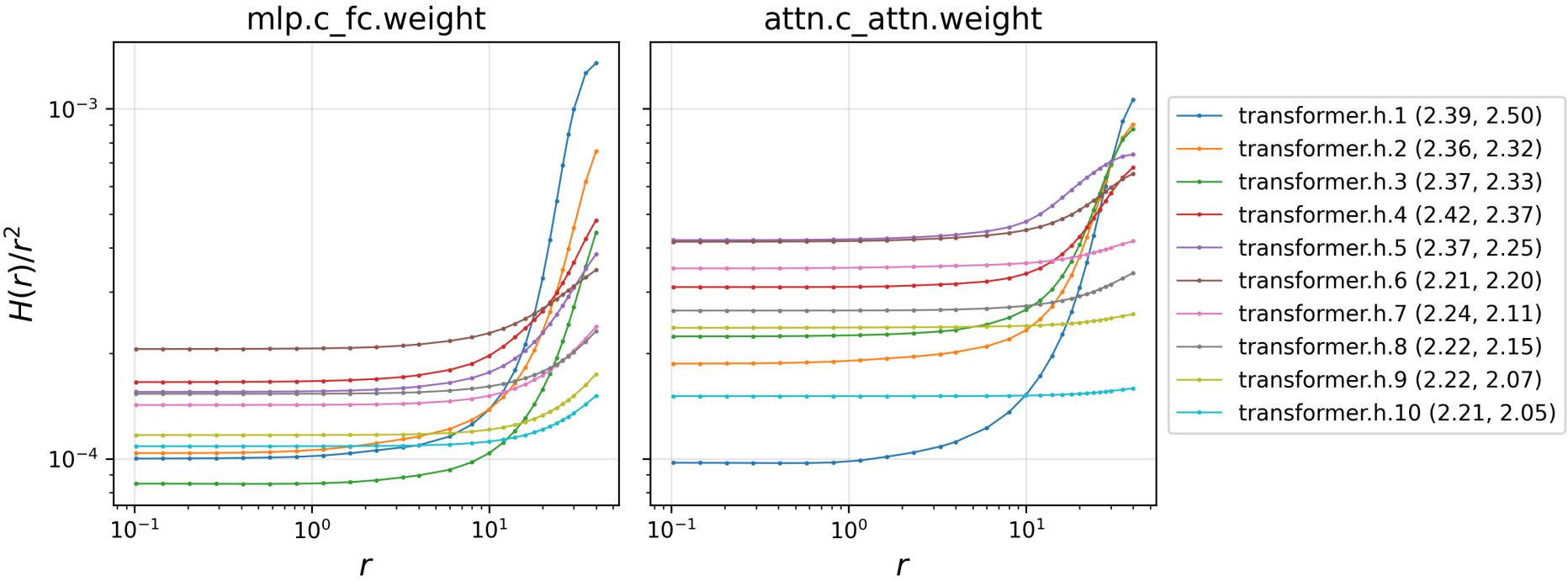


Empirical evidence: How curvature depends on radius





A closer look: $H(r)/r^2$



- It will eventually reach a plateau
- But this occurs beyond the “trust region”

Spectrum homogenization

Assumption

Curvature function H is strictly increasing and $H(\sqrt{x})$ is convex in x

- Example: $H(r) = r^{2+\alpha}$ with $\alpha > 0$ (super-quadratic growth)

Spectrum homogenization

Assumption

Curvature function H is strictly increasing and $H(\sqrt{x})$ is convex in x

- Example: $H(r) = r^{2+\alpha}$ with $\alpha > 0$ (super-quadratic growth)

Theorem

Let $Q^\star$ be the optimal solution. Let $G = U\Sigma V^\top$ and $Q^\star = U^\star \Sigma^\star V^{\star\top}$ be the SVDs. Then

- $U^\star = U, V^\star = V$
- *$\Sigma^\star$ and Σ have the same ordering of the singular values*
- $$\frac{\max\{\sigma_i^\star, \sigma_j^\star\}}{\min\{\sigma_i^\star, \sigma_j^\star\}} \leq \frac{\max\{\sigma_i, \sigma_j\}}{\min\{\sigma_i, \sigma_j\}}$$

Spectrum homogenization

Assumption

Curvature function H is strictly increasing and $H(\sqrt{x})$ is convex in x

- Example: $H(r) = r^{2+\alpha}$ with $\alpha > 0$ (super-quadratic growth)

Theorem

Let $Q^\star$ be the optimal solution. Let $G = U\Sigma V^\top$ and $Q^\star = U^\star \Sigma^\star V^{\star\top}$ be the SVDs. Then

- $U^\star = U, V^\star = V$
- *$\Sigma^\star$ and Σ have the same ordering of the singular values*
- $$\frac{\max\{\sigma_i^\star, \sigma_j^\star\}}{\min\{\sigma_i^\star, \sigma_j^\star\}} \leq \frac{\max\{\sigma_i, \sigma_j\}}{\min\{\sigma_i, \sigma_j\}}$$

- More homogeneous spectrum than the original gradient
- *Not* exact orthogonalization
- Newton–Schulz seems *not* optimal since it is not monotone

An example: Quartic growth

Consider $H(r) = cr^4$. The isotropic curvature model becomes:

$$\min_{\tilde{\sigma}} -\sum_i \sigma_i \tilde{\sigma}_i + \frac{c}{n(n+2)} \left[\left(\sum_i \tilde{\sigma}_i^2 \right)^2 + 2 \sum_i \tilde{\sigma}_i^4 \right]$$

An example: Quartic growth

Consider $H(r) = cr^4$. The isotropic curvature model becomes:

$$\min_{\tilde{\sigma}} -\sum_i \sigma_i \tilde{\sigma}_i + \frac{c}{n(n+2)} \left[\left(\sum_i \tilde{\sigma}_i^2 \right)^2 + 2 \sum_i \tilde{\sigma}_i^4 \right]$$

- Optimality condition: Unique real root for the cubic system

$$(\sigma_j^\star)^3 + D\sigma_j^\star \propto \sigma_j \quad \text{where } D = \tfrac{1}{2} \textstyle\sum (\sigma_i^\star)^2$$

- Order is preserved: $\sigma_i > \sigma_j \implies \sigma_i^\star > \sigma_j^\star$

An example: Quartic growth

Consider $H(r) = cr^4$. The isotropic curvature model becomes:

$$\min_{\tilde{\sigma}} -\sum_i \sigma_i \tilde{\sigma}_i + \frac{c}{n(n+2)} \left[\left(\sum_i \tilde{\sigma}_i^2 \right)^2 + 2 \sum_i \tilde{\sigma}_i^4 \right]$$

- Optimality condition: Unique real root for the cubic system

$$(\sigma_j^\star)^3 + D\sigma_j^\star \propto \sigma_j \quad \text{where } D = \tfrac{1}{2} \textstyle\sum (\sigma_i^\star)^2$$

- Order is preserved: $\sigma_i > \sigma_j \implies \sigma_i^\star > \sigma_j^\star$

Spectrum homogenization

If $\sigma_i > \sigma_j > 0$, we can derive:

$$\frac{\sigma_i^\star}{\sigma_j^\star} = \underbrace{\frac{(\sigma_j^\star)^2 + D}{(\sigma_i^\star)^2 + D}}_{<1} \cdot \frac{\sigma_i}{\sigma_j} < \frac{\sigma_i}{\sigma_j}$$

What if curvature is sub-quadratic?

- Quadratic case ($H(r) = cr^2$):

$$Q^\star \propto G$$

  Solution is proportional to gradient (no spectral change)

What if curvature is sub-quadratic?

- Quadratic case ($H(r) = cr^2$):

$$Q^\star \propto G$$

  Solution is proportional to gradient (no spectral change)

Theorem (Sub-quadratic growth)

*If H is strictly increasing convex, but $H(\sqrt{x})$ is **concave**, then*

$$\frac{\max\{\sigma_i^\star, \sigma_j^\star\}}{\min\{\sigma_i^\star, \sigma_j^\star\}} \geq \frac{\max\{\sigma_i, \sigma_j\}}{\min\{\sigma_i, \sigma_j\}}$$

What if curvature is sub-quadratic?

- Quadratic case ($H(r) = cr^2$):

$$Q^\star \propto G$$

  Solution is proportional to gradient (no spectral change)

Theorem (Sub-quadratic growth)

*If H is strictly increasing convex, but $H(\sqrt{x})$ is **concave**, then*

$$\frac{\max\{\sigma_i^\star, \sigma_j^\star\}}{\min\{\sigma_i^\star, \sigma_j^\star\}} \geq \frac{\max\{\sigma_i, \sigma_j\}}{\min\{\sigma_i, \sigma_j\}}$$

- The spectrum becomes *more heterogeneous*
- "Rich get richer": Dominant singular values are amplified relative to smaller ones

When does (exact) orthogonalization become optimal?

Assumption

H is a non-decreasing convex function such that there exists $\tilde{r} > 0$ such that $H'(\tilde{r}-) = A$ and $H'(\tilde{r}+) = B$ for some $0 < A < B < \infty$

- In words, there is a kink where H "takes off"
- Proof idea: at the kink the subdifferential is "wide" enough to cover all singular values of G

When does (exact) orthogonalization become optimal?

Assumption

H is a non-decreasing convex function such that there exists $\tilde{r} > 0$ such that $H'(\tilde{r}-) = A$ and $H'(\tilde{r}+) = B$ for some $0 < A < B < \infty$

- In words, there is a kink where H "takes off"
- Proof idea: at the kink the subdifferential is "wide" enough to cover all singular values of G

Theorem

For sufficiently small A and large B, we have $G^\star = c \cdot \mathrm{msgn}(G) \equiv c \cdot UV^\top$

When does (exact) orthogonalization become optimal?

Assumption

H is a non-decreasing convex function such that there exists $\tilde{r} > 0$ such that $H'(\tilde{r}-) = A$ and $H'(\tilde{r}+) = B$ for some $0 < A < B < \infty$

- In words, there is a kink where H "takes off"
- Proof idea: at the kink the subdifferential is "wide" enough to cover all singular values of G

Theorem

For sufficiently small A and large B, we have $G^\star = c \cdot \mathrm{msgn}(G) \equiv c \cdot UV^\top$

- The subdifferential at the kink is wide enough to cover the spectrum of G
- Suggests that, in practice, orthogonalization *might* not be optimal

Kink is necessary for gradient orthogonalization

Theorem

For sufficiently small A and large B, we have $G^\star = c \cdot \mathrm{msgn}(G) \equiv c \cdot UV^\top$

Theorem

Let G not be a (scaled) orthonormal matrix. If the isotropic curvature model has a global solution that takes the form $Q^\star = c \cdot \mathrm{msgn}(G)$, then H must have a kink

Kink is necessary for gradient orthogonalization

Theorem

For sufficiently small A and large B, we have $G^\star = c \cdot \operatorname{msgn}(G) \equiv c \cdot UV^\top$

Theorem

Let G not be a (scaled) orthonormal matrix. If the isotropic curvature model has a global solution that takes the form $Q^\star = c \cdot \operatorname{msgn}(G)$, then H must have a kink

- Proof idea: without any kink, the subgradient reduces to a unique gradient, which cannot satisfy the first-order optimality condition

Insights from the isotropic curvature perspective

- While first-order information must be obtained through "observation" (backpropagation), higher-order information might resemble a *static field*

Insights from the isotropic curvature perspective

- While first-order information must be obtained through "observation" (backpropagation), higher-order information might resemble a *static field*
- More experiments are needed, possibly requiring refined adjustments to the model

Insights from the isotropic curvature perspective

- While first-order information must be obtained through "observation" (backpropagation), higher-order information might resemble a *static field*
- More experiments are needed, possibly requiring refined adjustments to the model
- Orthogonalization is "directionally correct", but perhaps not optimal

Insights from the isotropic curvature perspective

- While first-order information must be obtained through "observation" (backpropagation), higher-order information might resemble a *static field*
- More experiments are needed, possibly requiring refined adjustments to the model
- Orthogonalization is "directionally correct", but perhaps not optimal
  - Evidence: the improvement of Muon compared to Adam becomes less significant as epoch increases (Wen et al., 2025)

Insights from the isotropic curvature perspective

- While first-order information must be obtained through "observation" (backpropagation), higher-order information might resemble a *static field*
- More experiments are needed, possibly requiring refined adjustments to the model
- Orthogonalization is "directionally correct", but perhaps not optimal
  - Evidence: the improvement of Muon compared to Adam becomes less significant as epoch increases (Wen et al., 2025)
- More likely, optimality is attained through spectrum homogenization

A (possible) design principle for optimizers

- Estimate the curvature function H
- Practical challenge: H varies as it depends on architecture, data, width, depth, etc., and whether at initialization or later in training dynamics
- Mathematical challenge: can polynomial methods solve the isotropic curvature model?

Concluding remarks

Comparisons between the two perspectives

Both agree that we should leave singular spaces unchanged; the task is to work on the spectrum!

- Preconditioning built concretely on optimization theory
- But does not directly relate orthogonalization to optimization performance
- Considers orthogonalization only

- Isotropic curvature perspective is phenomenological
- But it relates spectrum homogenization to improved optimization performance (although only for one iteration)
- Differentiates between homogenization and orthogonalization

References. Thank you!

1. *PolarGrad: A Class of Matrix-Gradient Optimizers from a Unifying Preconditioning Perspective*
   Tim Lau, Qi Long, and Weijie Su. arXiv:2505.21799
2. *Isotropic Curvature Model for Understanding Deep Learning Optimization: Is Gradient Orthogonalization Optimal?*
   Weijie Su. arXiv:2511.00674

References I

N. Amsel, D. Persson, C. Musco, and R. M. Gower. The polar express: Optimal matrix sign methods and their application to the muon algorithm. *arXiv preprint arXiv:2505.16932*, 2025.

J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2): 223–311, 2018.

D. Carlson, Y.-P. Hsieh, E. Collins, L. Carin, and V. Cevher. Stochastic spectral descent for discrete graphical models. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):296–311, 2016.

G. E. Dahl, F. Schneider, Z. Nado, N. Agarwal, C. S. Sastry, P. Hennig, S. Medapati, R. Eschenhagen, P. Kasimbeg, D. Suo, J. Bae, J. Gilmer, A. L. Peirson, B. Khan, R. Anil, M. Rabbat, S. Krishnan, D. Snider, E. Amid, K. Chen, C. J. Maddison, R. Vasudev, M. Badura, A. Garg, and P. Mattson. Benchmarking neural network training algorithms. *arXiv preprint arXiv:2306.07179*, 2023.

V. Gupta, T. Koren, and Y. Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

K. Jordan, Y. Jin, V. Boza, Y. Jiacheng, F. Cecista, L. Newhouse, and J. Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL `https://kellerjordan.github.io/posts/muon/`.

P. Kasimbeg, F. Schneider, R. Eschenhagen, J. Bae, C. S. Sastry, M. Saroufim, B. Feng, L. Wright, E. Z. Yang, Z. Nado, S. Medapati, P. Hennig, M. Rabbat, and G. E. Dahl. Accelerating neural network training: An analysis of the AlgoPerf competition. In *International Conference on Learning Representations (ICLR)*, 2025.

D. P. Kingma and J. L. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

References II

J. Liu, J. Su, X. Yao, Z. Jiang, G. Lai, Y. Du, Y. Qin, W. Xu, E. Lu, J. Yan, Y. Chen, H. Zheng, Y. Liu, S. Liu, B. Yin, W. He, H. Zhu, Y. Wang, J. Wang, M. Dong, Z. Zhang, Y. Kang, H. Zhang, X. Xu, Y. Zhang, Y. Wu, X. Zhou, and Z. Yang. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

F. Orabona. Neural networks (maybe) evolved to make Adam the best optimizer, 2020. URL `https://parameterfree.com/2020/12/06/neural-network-maybe-evolved-to-make-adam-the-best-optimizer/`.

H.-J. M. Shi, T.-H. Lee, S. Iwasaki, J. Gallego-Posada, Z. Li, K. Rangadurai, D. Mudigere, and M. Rabbat. A distributed data-parallel PyTorch implementation of the distributed Shampoo optimizer for training neural networks at-scale. *arXiv preprint arXiv:2309.06497*, 2023.

N. Vyas, D. Morwani, R. Zhao, I. Shapira, D. Brandfonbrener, L. Janson, and S. Kakade. SOAP: Improving and stabilizing Shampoo using Adam. In *International Conference on Learning Representations (ICLR)*, 2025.

K. Wen, D. Hall, T. Ma, and P. Liang. Fantastic pretraining optimizers and where to find them. *arXiv preprint arXiv:2509.02046*, 2025.