Matrix Procrustes Problem and Its Applications in Deep Learning

Maxim Rakhuba

HSE University

CS department
October, 2025

Outline

Matrix Procrustes Problem

Newton-Schulz and Beyond

Muon's polynomial choice

Outline

Orthogonal Procrustes Problem: Formulation

Problem: Given $A, B \in \mathbb{R}^{m \times n}$, $m \geqslant n$, solve

$$\min \left\{ \|A - BQ\|_F : Q^\top Q = I \right\}.$$

When $B = I$, we get the nearest orthogonal matrix problem from Muon.

Orthogonal Procrustes Problem: Formulation

Problem: Given $A, B \in \mathbb{R}^{m \times n}$, $m \geqslant n$, solve

$$\min \left\{ \|A - BQ\|_F : Q^\top Q = I \right\}.$$

When $B = I$, we get the nearest orthogonal matrix problem from Muon.

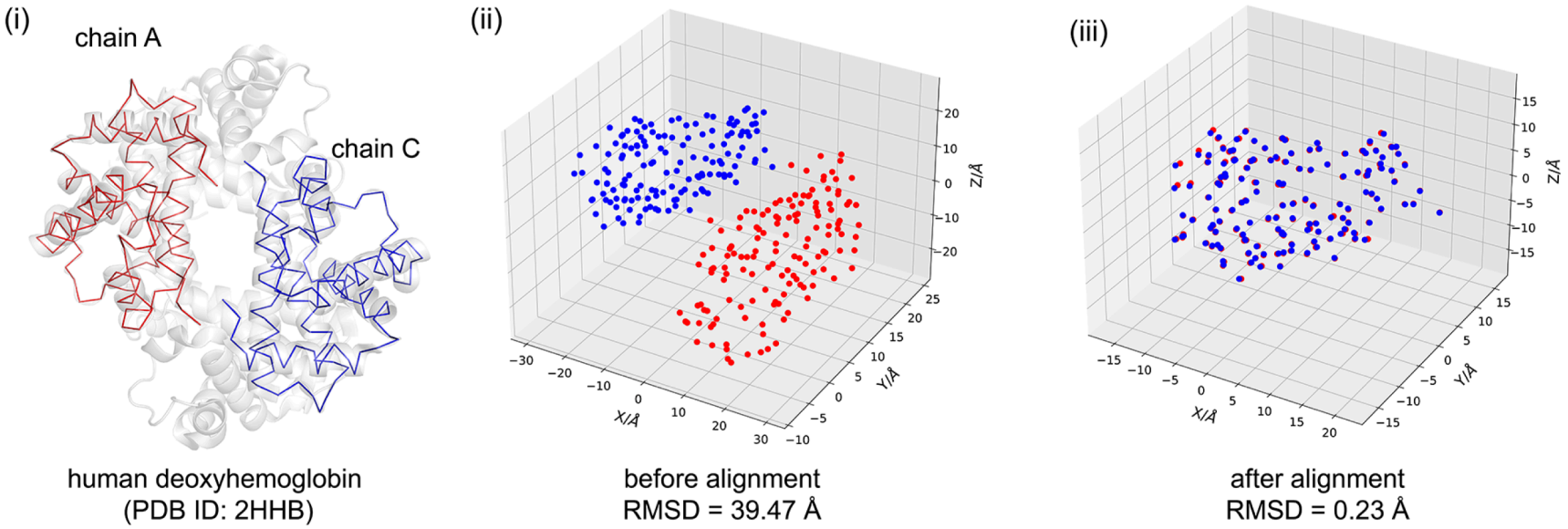


Figure: Protein structure alignment. Example from the documentation of `procrustes` library: `https://procrustes.readthedocs.io/_/downloads/en/latest/pdf/`

Orthogonal Procrustes Problem: Solution

$$\min \left\{ \|A - BQ\|_F : Q^\top Q = I \right\}.$$

How to solve?

1. $B^\top A = U\Sigma V^\top$ (SVD);
2. $Q = UV^\top$.

Why Procrustes?

Setting $Q = UV^\top$ is like replacing $\Sigma \to I$: "cutting" $\sigma_i > 1$ and "stretching" $\sigma_i < 1$.

Figure: In Greek myth, Procrustes was a thief who forced his victims to fit his bed, either by stretching them or cutting their limbs.

Muon: high-level computational idea

Muon (**M**oment**U**m **O**rthogonalized by **N**ewton-Schulz)
`https://kellerjordan.github.io/posts/muon/`

Muon: high-level computational idea

Muon (**M**oment**U**m **O**rthogonalized by **N**ewton-Schulz)
`https://kellerjordan.github.io/posts/muon/`

Setting: 2D weight matrices in hidden layers (and conv kernels flattened).
Core move: *Orthogonalize* momentum M_t by a fast iteration (Newton–Schulz).

Muon: high-level computational idea

Muon (**M**oment**U**m **O**rthogonalized by **N**ewton-Schulz)
`https://kellerjordan.github.io/posts/muon/`

Setting: 2D weight matrices in hidden layers (and conv kernels flattened).
Core move: *Orthogonalize* momentum M_t by a fast iteration (Newton–Schulz).

Muon (layerwise, schematic)

Given gradient g_t and momentum $m_t \leftarrow \beta m_{t-1} + (1-\beta) g_t$:

(1) $M_t \leftarrow \texttt{matricise}(m_t)$

(2) $\widehat{O}_t \leftarrow \texttt{NewtonSchulz5}(M_t) \approx \underset{O^\top O = I}{\arg\min} \|O - M_t\|_F$

(3) $W_{t+1} \leftarrow W_t - \eta\, \widehat{O}_t$

Outline

Matrix Procrustes Problem

Newton-Schulz and Beyond

Muon's polynomial choice

Newton-Schulz iteration vs. SVD

$$UV^\top \in \operatorname*{arg\,min}_{Q:\ Q^\top Q = I} \|A - Q\|_F.$$

Methods

1. Using SVD: $\mathcal{O}(mn^2)$ FLOPs.
   - $+$ High reliability, controlled accuracy.
   - $-$ Large constant in $\mathcal{O}(\cdot)$.

Newton-Schulz iteration vs. SVD

$$UV^\top \in \operatorname*{arg\,min}_{Q:\ Q^\top Q = I} \|A - Q\|_F.$$

Methods

1. Using SVD: $\mathcal{O}(mn^2)$ FLOPs.
   - \+ High reliability, controlled accuracy.
   - − Large constant in $\mathcal{O}(\cdot)$.
2. Newton-Schulz iteration:

$$X_{k+1} = \frac{1}{2} X_k (3I - X_k^\top X_k), \quad X_0 = A.$$

   - \+ GPU-friendly (matrix multiplications).
   - − Convergence can be an issue.

Newton–Schulz iteration convergence

$$X_{k+1} = \frac{1}{2} X_k (3I - X_k^\top X_k), \quad X_0 = A.$$

Each step is a polynomial transform of singular values

For $X_k = U \Sigma_k V^\top$ (SVD):

$$X_{k+1} =$$

Newton–Schulz iteration convergence

$$X_{k+1} = \frac{1}{2} X_k (3I - X_k^\top X_k), \quad X_0 = A.$$

Each step is a polynomial transform of singular values

For $X_k = U\Sigma_k V^\top$ (SVD):

$$X_{k+1} = \frac{1}{2}\left(3U\Sigma_k V^\top - U\Sigma_k V^\top V\Sigma_k U^\top U\Sigma_k V^\top\right) =$$

Newton–Schulz iteration convergence

$$X_{k+1} = \frac{1}{2} X_k (3I - X_k^\top X_k), \quad X_0 = A.$$

Each step is a polynomial transform of singular values

For $X_k = U \Sigma_k V^\top$ (SVD):

$$X_{k+1} = \frac{1}{2}\left(3U\Sigma_k V^\top - U\Sigma_k V^\top V \Sigma_k U^\top U \Sigma_k V^\top\right) = \frac{1}{2} U(3\Sigma_k - \Sigma_k^3) V^\top = U\, p(\Sigma_k)\, V^\top,$$

Newton–Schulz iteration convergence

$$X_{k+1} = \frac{1}{2} X_k (3I - X_k^\top X_k), \quad X_0 = A.$$

Each step is a polynomial transform of singular values

For $X_k = U \Sigma_k V^\top$ (SVD):

$$X_{k+1} = \frac{1}{2}\left(3U\Sigma_k V^\top - U\Sigma_k V^\top V \Sigma_k U^\top U \Sigma_k V^\top\right) = \frac{1}{2} U(3\Sigma_k - \Sigma_k^3) V^\top = U\, p(\Sigma_k)\, V^\top,$$

where $p(\sigma) = \frac{1}{2}\sigma(3 - \sigma^2)$.

Newton–Schulz iteration convergence

$$X_{k+1} = \frac{1}{2} X_k (3I - X_k^\top X_k), \quad X_0 = A.$$

Each step is a polynomial transform of singular values

For $X_k = U \Sigma_k V^\top$ (SVD):

$$X_{k+1} = \frac{1}{2}\left(3U\Sigma_k V^\top - U\Sigma_k V^\top V \Sigma_k U^\top U \Sigma_k V^\top\right) = \frac{1}{2} U(3\Sigma_k - \Sigma_k^3) V^\top = U\, p(\Sigma_k)\, V^\top,$$

where $p(\sigma) = \frac{1}{2}\sigma(3 - \sigma^2)$. Thus,

$$X_{k+1} = U\, p(p(\ldots p(\Sigma_0)))\, V^\top.$$

Newton–Schulz iteration convergence

$$X_{k+1} = \frac{1}{2} X_k (3I - X_k^\top X_k), \quad X_0 = A.$$

Each step is a polynomial transform of singular values

For $X_k = U \Sigma_k V^\top$ (SVD):

$$X_{k+1} = \frac{1}{2}\left(3U\Sigma_k V^\top - U\Sigma_k V^\top V \Sigma_k U^\top U \Sigma_k V^\top\right) = \frac{1}{2} U(3\Sigma_k - \Sigma_k^3) V^\top = U\, p(\Sigma_k)\, V^\top,$$

where $p(\sigma) = \frac{1}{2}\sigma(3 - \sigma^2)$. Thus,

$$X_{k+1} = U\, p(p(\ldots p(\Sigma_0)))\, V^\top.$$

So we want

$$p(p(\ldots p(x))) \to 1 \quad \text{on} \quad [\sigma_{\min}, \sigma_{\max}].$$

Is it possible to do better?

1. Grishina, Ekaterina, Matvey Smirnov, and Maxim Rakhuba. *Accelerating Newton-Schulz Iteration for Orthogonalization via Chebyshev-type Polynomials.* arXiv:2506.10935, 2025.
2. Amsel, Noah, et al. *The polar express: Optimal matrix sign methods and their application to the Muon algorithm.* arXiv:2505.16932, 2025.

Is it possible to do better?

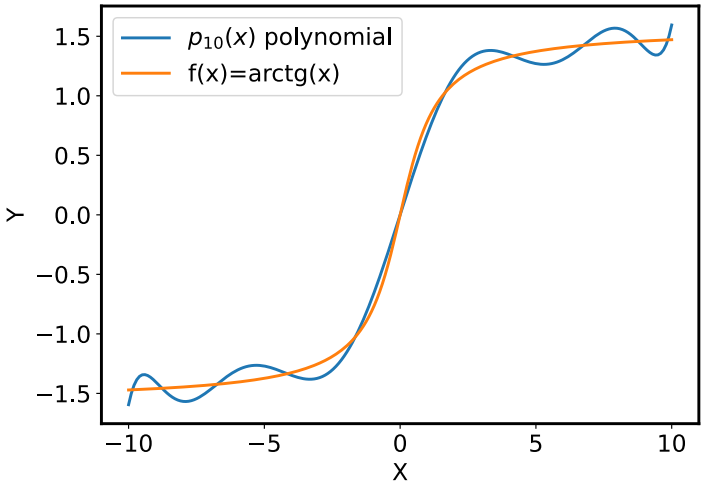
Theorem (Chebyshev)

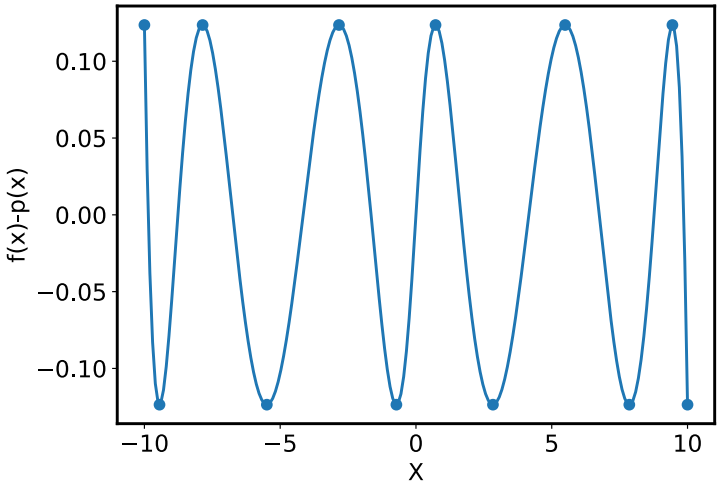
$f \in C[a,b]$ admits unique best approximation $p \in \mathbb{P}_n$:

$$p = \arg\min_{q \in \mathbb{P}_n} \|f - q\|_{C[a,b]} .$$

Moreover, p is best $\iff$ there exist $x_0 < x_1 < \cdots < x_{n+1}$ (alternance points):

$$|p(x_j) - f(x_j)| = \|p - f\|_{C[a,b]}, \quad p(x_j) - f(x_j) = -(p(x_{j-1}) - f(x_{j-1}))$$





Chebyshev-acceleated Newton-Schulz (CANS)

For

$$\mathbb{L}_2 = \left\{\alpha_1 x + \alpha_3 x^3 : \alpha_1, \alpha_3 \in \mathbb{R}\right\},$$

one can find optimal α_1 and α_3 in terms of $[a, b]$.

Optimized Newton-Schulz

- $A \to A/\|A\|_2$ and let $[a_0, b_0] := [\sigma_n/\sigma_1, 1]$

Chebyshev-acceleated Newton-Schulz (CANS)

For

$$\mathbb{L}_2 = \left\{\alpha_1 x + \alpha_3 x^3 : \alpha_1, \alpha_3 \in \mathbb{R}\right\},$$

one can find optimal α_1 and α_3 in terms of $[a, b]$.

Optimized Newton-Schulz

- $A \to A/\|A\|_2$ and let $[a_0, b_0] \coloneqq [\sigma_n/\sigma_1, 1]$
- Find optimal p_0 on $[a_0, b_0]$.

$$\delta_0 = \|p_0 - 1\|_{C[a_0, b_0]}.$$

Chebyshev-acceleated Newton-Schulz (CANS)

For

$$\mathbb{L}_2 = \left\{\alpha_1 x + \alpha_3 x^3 : \alpha_1, \alpha_3 \in \mathbb{R}\right\},$$

one can find optimal α_1 and α_3 in terms of $[a, b]$.

Optimized Newton-Schulz

- $A \to A/\|A\|_2$ and let $[a_0, b_0] \coloneqq [\sigma_n/\sigma_1, 1]$
- Find optimal p_0 on $[a_0, b_0]$.

$$\delta_0 = \|p_0 - 1\|_{C[a_0, b_0]}.$$

- Find p_1 on $[a_1, b_1] \coloneqq [1 - \delta_0, 1 + \delta_0]$ and let

$$\delta_1 = \|p_1 - 1\|_{C[a_1, b_1]}.$$

Chebyshev-acceleated Newton-Schulz (CANS)

For

$$\mathbb{L}_2 = \left\{\alpha_1 x + \alpha_3 x^3 : \alpha_1, \alpha_3 \in \mathbb{R}\right\},$$

one can find optimal α_1 and α_3 in terms of $[a, b]$.

Optimized Newton-Schulz

- $A \to A/\|A\|_2$ and let $[a_0, b_0] := [\sigma_n/\sigma_1, 1]$
- Find optimal p_0 on $[a_0, b_0]$.

$$\delta_0 = \|p_0 - 1\|_{C[a_0, b_0]}.$$

- Find p_1 on $[a_1, b_1] := [1 - \delta_0, 1 + \delta_0]$ and let

$$\delta_1 = \|p_1 - 1\|_{C[a_1, b_1]}.$$

- Find p_2 on $[a_2, b_2] := [1 - \delta_1, 1 + \delta_1]$ ...

Chebyshev-accelerated Newton-Schulz (CANS)

As a result, we obtain CANS:

$$X_{k+1} = p_k(X_k),$$

Proposition

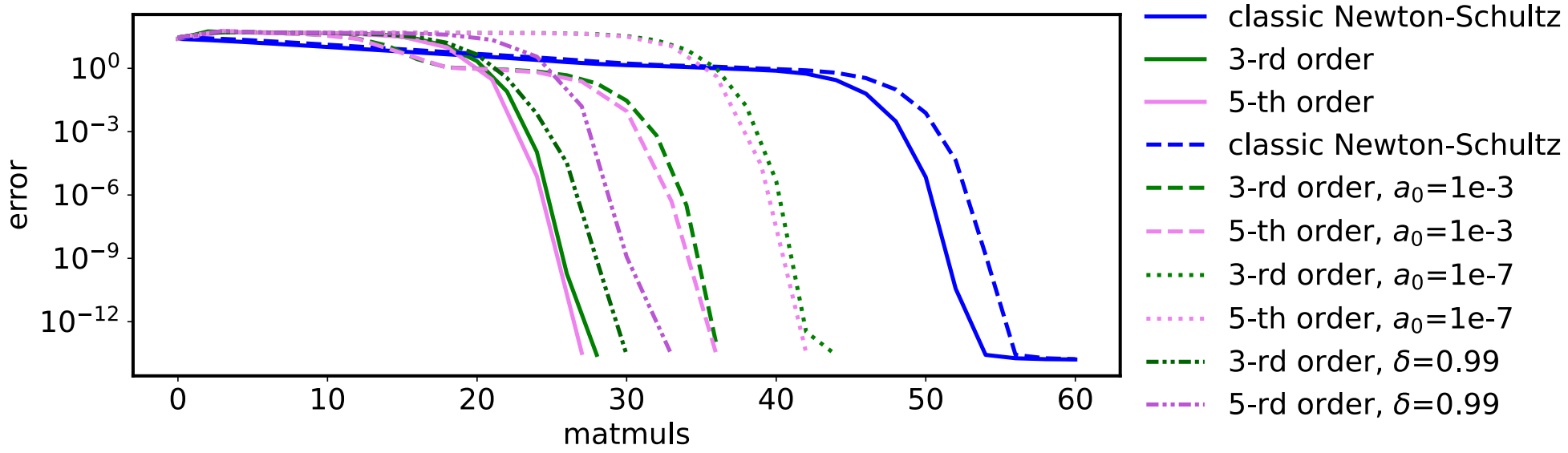
$$\varepsilon_k = \|X_k - UV^\top\|_2 \to 0, \quad k \to \infty,$$

and

$$\varepsilon_{k+1} \leqslant \varepsilon_k^2, \qquad \lim_{k\to\infty} \frac{\varepsilon_{k+1}}{\varepsilon_k^2} = \frac{3}{4}.$$

Chebyshev-acceleated Newton-Schulz (CANS)

Convergence on a random matrix:

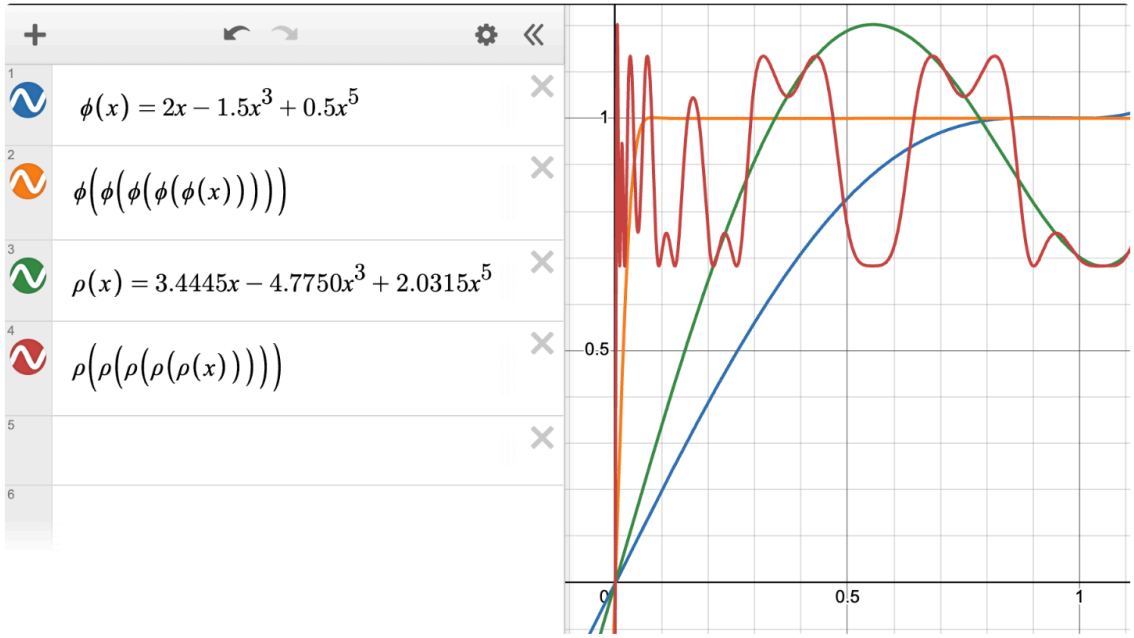


Outline

Muon inflates small singular values faster

Hand-tuned coefficients (used for 5 steps) that steepens the slope near 0:

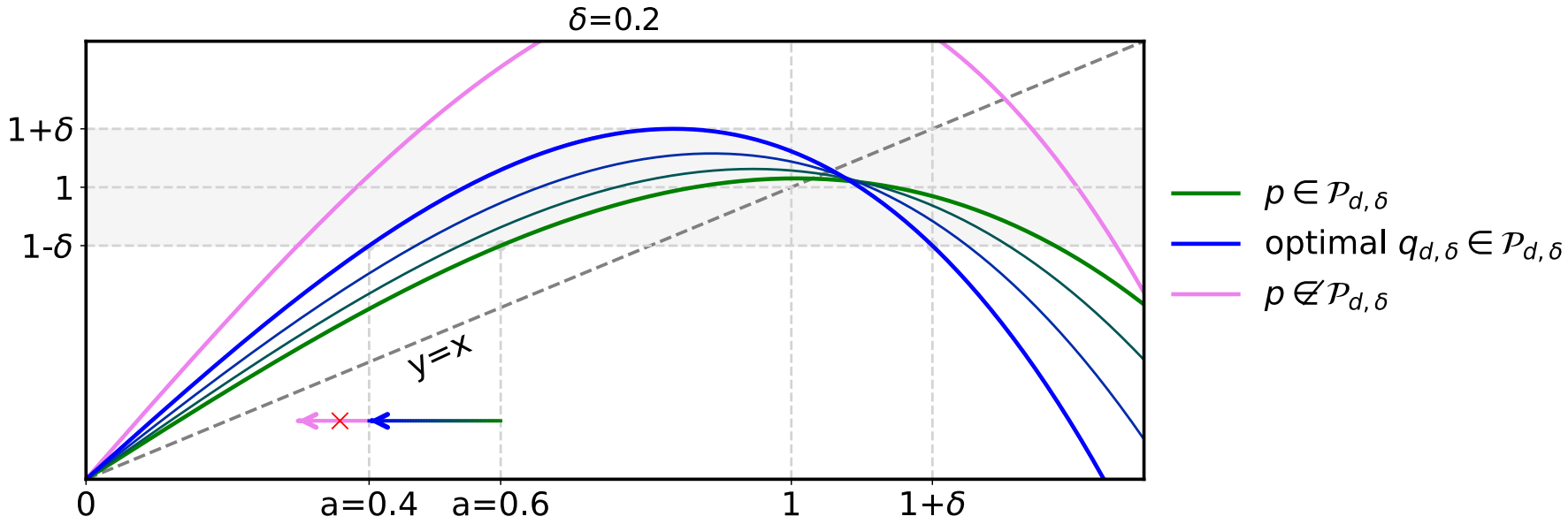
$$\varphi(x) = 3.4445\,x \;-\; 4.7750\,x^3 \;+\; 2.0315\,x^5.$$



Source: Blog post: `https://kellerjordan.github.io/posts/muon/`

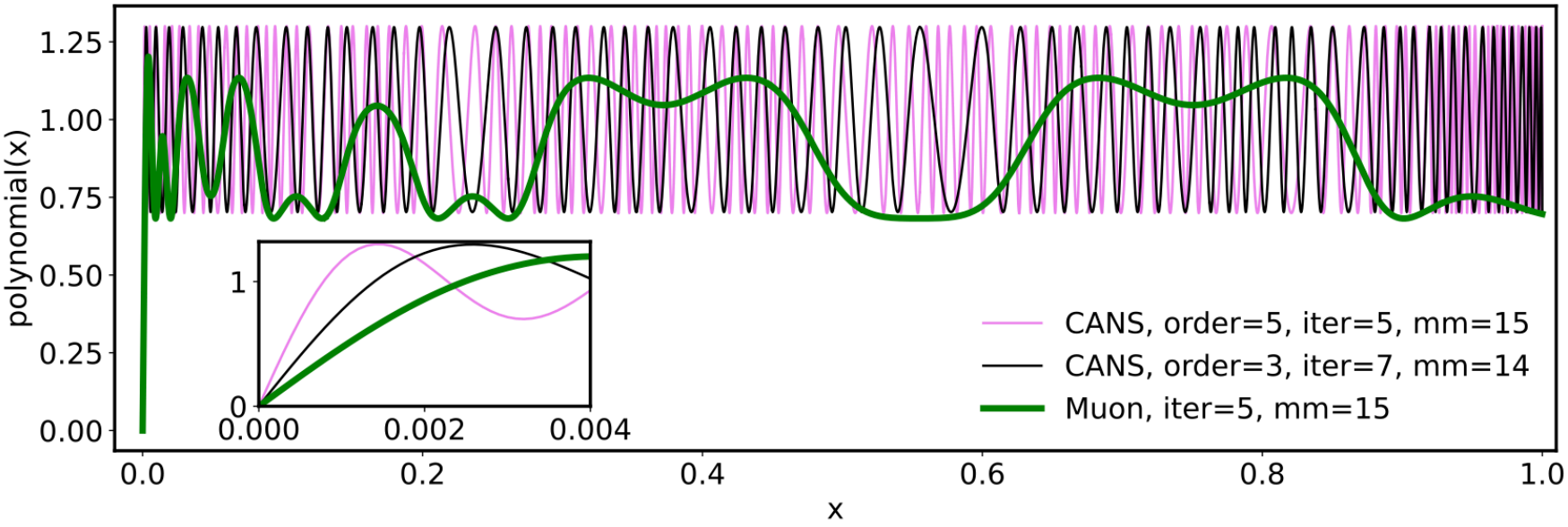
How to maximize derivative at 0?

Optimal polynomial on $[a, 1]$ with the smallest possible a to fit $[1-\delta, 1+\delta]$.



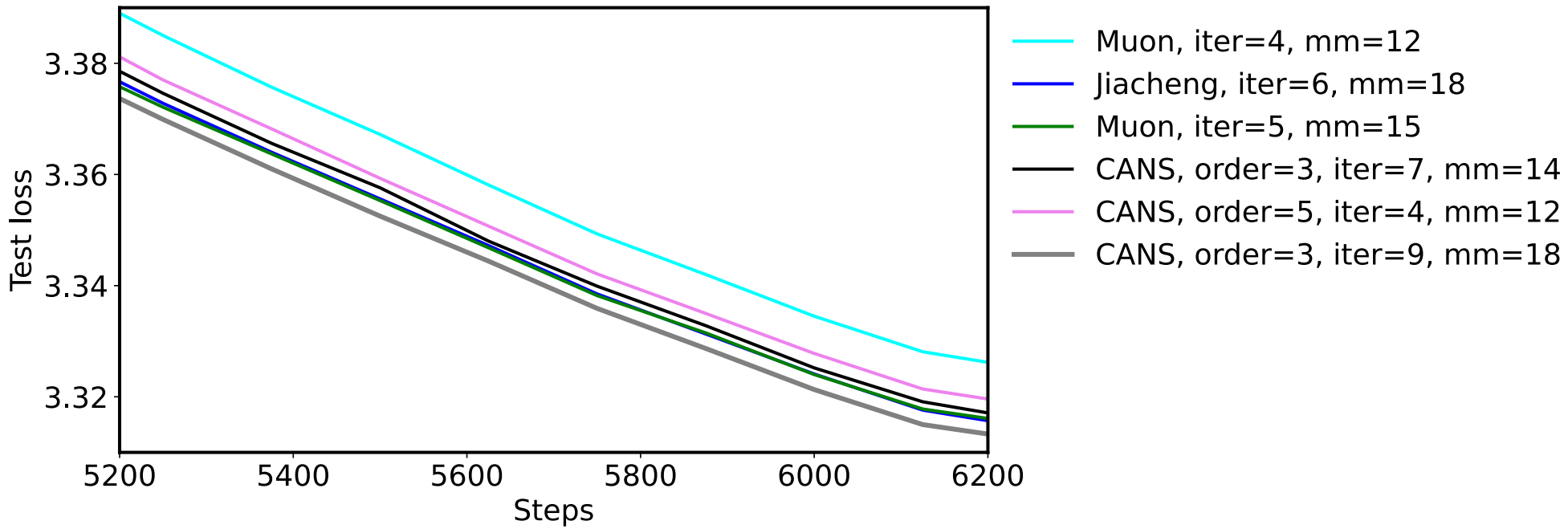
Maximizes $p'(0)$ for $\deg = 3$ and under certain assumption for $\deg \geqslant 5$.

How to maximize derivative at 0?



Muon optimizer using our polynomials

Test loss of NanoGPT trained using Muon and different polynomials.



Conclusion

1. New method CANS.

2. Application in Muon optimizer.

3. Other applications:
   Retraction in Riemannian optimization;
   ProcrustesGPT: compression of neural networks [arXiv:2506.02818].

Thank you for your attention!