Generalized Gradient Norm Clipping & Non-Euclidean (L_0, L_1)-Smoothness

Thomas Pethick[*,1], Wanyun Xie[*,1], Mete Erdogan[1],
Kimon Antonakopoulos[1], Antonio Silveti-Falls[2], Volkan Cevher[1]


[1] EPFL, [2] Inria
[*] Equal contribution

What is non-Euclidean methods?

Algorithmic Template

$$x^{k+1} \in \arg\min_x \gamma\langle d^k, x\rangle + h_k(x)$$

with gradient/dual feedback d^k.

- **Special case**: Gradient descent

$$h_k(x) = \tfrac{1}{2}\|x - x^k\|_2^2 \;\Rightarrow\; x^{k+1} = x^k - \gamma d^k$$

- **Processing of gradients**: can change the direction and magnitude

Why do we care about non-Euclidean methods?

They unify a wide range of widely used methods!

$$x^{k+1} \in \arg\min_{x \in \mathcal{X}} \gamma_k \langle d^k, x \rangle + h_k(x) \quad \text{with} \quad h_k(x) = \tfrac{1}{2}\|x - x^k\|^2_{2,H_k}.$$

Algorithm	**Dual feedback** d^k	**Preconditioner** H_k
AdaGrad / RMSProp	g^k	$[H_k]_{ii} = [H_{k-1}]_{ii} + (g_i^k)^2; \forall i \in [p]$
Adam / AdamW	$\beta_1 d^{k-1} + (1-\beta_1) g^k$	$[H_k]_{ii} = \beta_2 [H_{k-1}]_{ii} + (1-\beta_2)(g_i^k)^2; \forall i \in [p]$
ℓ_∞-descent	g^k	$[H_k]_{ii} = (g_i^k)^2; \forall i \in [p]$
SignSGD / Signum	$g^k / \beta_1 d^{k-1} + (1-\beta_1) g^k$	$[H_k]_{ii} = (g_i^k)^2; \forall i \in [p]$
Spectral descent	$G_k = \mathrm{mat}(g^k)$	$R_k^{1/2} \otimes L_k^{1/2} : \begin{cases} R_k = G_k G_k^\top \\ L_k = G_k^\top G_k \end{cases}$
Shampoo	$G_k = \mathrm{mat}(g^k)$	$R_k^{1/2} \otimes L_k^{1/2} : \begin{cases} R_k = \beta_2 R_{k-1} + (1-\beta_2) G_k G_k^\top \\ L_k = \beta_2 L_{k-1} + (1-\beta_2) G_k^\top G_k \end{cases}$
Scion / Muon (spectral)	$\beta_1 d^{k-1} + (1-\beta_1) G_k$	$R_k^{1/2} \otimes L_k^{1/2} : \begin{cases} R_k = G_k G_k^\top \\ L_k = G_k^\top G_k \end{cases}$

$\beta_2 \to 0$:

- RMSProp $\Rightarrow$ ℓ_∞-descent/SignSGD
- Adam $\Rightarrow$ Signum
- Shampoo $\Rightarrow$ Spectral descent

Table from "Training Neural Networks at Any Scale" [Pet+25]

Why do we care about non-Euclidean methods?

Assumption Lipschitz: $\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^d$.

Theorem

Steepest descent, $h_k(x) = \|x - x^k\|^2$, *with* $\gamma = 1/L$ *satisfies*

$$\min_{1 \leq k \leq n} \|\nabla f(x^k)\|_* \leq \sqrt{\frac{2L(f(x^1) - f^*)}{n}}$$

Favorable geometry when $L < L_2$ and $\|\nabla f\|_*/\|\nabla f\|_2$ large:

- Max-norm ℓ_∞: $\|\nabla f\|_1/\|\nabla f\|_2$ large (effective sparsity)
- Schatten-∞/Spectral: $\|\nabla f\|_{\mathcal{S}_1}/\|\nabla f\|_F$ large (effective rank)

Examples:

- Max-norm ℓ_∞: under log-sum-exp loss [Car+15]
- Schatten-∞/Spectral: grads in DL are high rank initially [DD25] [1]

$\Rightarrow$ Improved dimensionality dependency aka scaling law coefficient

[1] Over single sample: rank-1, so Spectral descent $\equiv$ Frobenius descent.
Explains recent successful SGD pre-training for LLMs [SGO25; Mar+25]

Goal of today

Algorithmic Template

$$x^{k+1} \in \arg\min_x \gamma\langle d^k, x\rangle + h_k(x)$$

with gradient/dual feedback d^k.

Steepest Descent

$h_k(x) = \frac{1}{2}\|x - x^k\|^2$

$\hookrightarrow$ Gradient Descent for ℓ_2:

$x^{k+1} = x^k - \gamma d^k$

Pro Descent method

Con Small enough stepsize

Conditional Gradient

$h_k(x) = \iota_{\rho\mathcal{D}}(x - x^k)$

with norm-ball $\mathcal{D} = \{x \colon \|x\| \leq 1\}$

$\hookrightarrow$ Normalized GD for ℓ_2

Pro Stable with large stepsize

Con Overshoots the solution

How to get best of both worlds?

Limitations of steepest descent

How do we model changing smoothness?

Assumption Non-Euclidean (L_0, L_1)-smooth if for $\|x - y\| \leq \frac{1}{L_1}$,

$$\|\nabla f(x) - \nabla f(y)\|_* \leq (L_0 + L_1 \|\nabla f(x)\|_*)\|x - y\|$$

Solution Conditional gradient = *Normalized* steepest descent

Theorem

Conditional gradient with $\gamma\rho \leq 1/2L_1$ *satisfies*

$$\min_{1 \leq k \leq n} \|\nabla f(x^k)\|_* \leq \frac{2\Delta}{\gamma\rho n} + 2L_0\gamma\rho$$

with $\Delta := f(x^1) - f^\star$.

Problem Not a descent methods (requires diminishing stepsize $\gamma \to \infty$).

How to maintain descent?

Steepest descent within a trust region:

$$x^{k+1} \in \underset{\|x-x^k\| \leq \rho}{\arg\min}\ \gamma \langle d^k, x \rangle + \tfrac{1}{2}\|x - x^k\|^2$$

$$\Downarrow$$

Generalized gradient norm clipping (GGNC)

$$x^{k+1} = x^k + \gamma \eta_k \operatorname{lmo}(d^k) \quad \text{with} \quad \eta_k := \min\{\rho, \|d^k\|_*\}$$

and linear minimization oracle, $\operatorname{lmo}(d) := \arg\min_{x:\|x-x^k\| \leq 1} \langle d, x \rangle$.

- Dual norm can easily be computed: $\|d^k\|_* = -\langle d^k, \operatorname{lmo}(d^k) \rangle$
- Particularly easy in Modula: `dual_norm = -d.dot(dualize(d))`

What is the benefit of clipping?

Answer Descent!

Theorem

GGNC with $\rho = \frac{L_0}{L_1}$ *and* $\gamma = \frac{1}{L_0}$ *satisfies*

$$\min_{1 \leq k \leq n} \|\nabla f(x^k)\|_* \leq \frac{2L_1\Delta}{n} + \sqrt{\frac{L_0\Delta}{n}}$$

with $\Delta := f(x^1) - f^\star$.

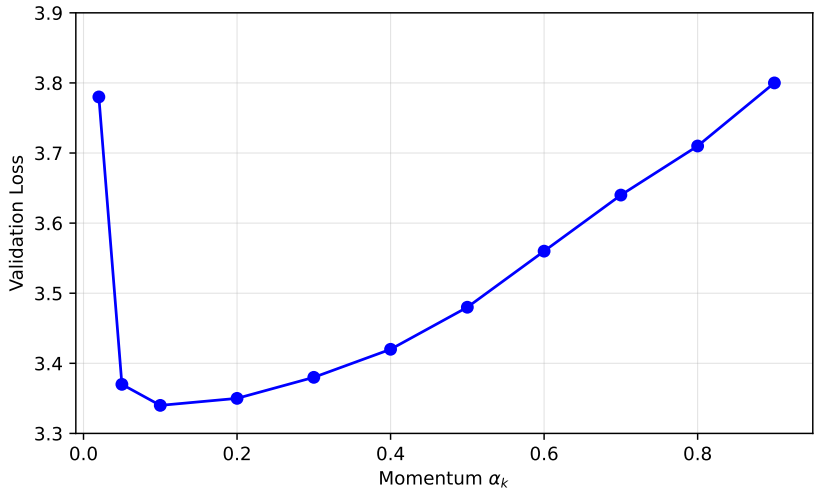
Extensions

Stochasticity How to deal with noise?

Momentum estimator

$$d^k = (1 - \alpha_k) d^{k-1} + \alpha_k \nabla f(x^k, \xi^k)$$

- Reduces variance of d^k to avoid bias of $\mathrm{lmo}(d^k)$ due to non-linearity
- $\Rightarrow$ Achieves order optimal $O(n^{-1/4})$ rate.



From "Training Neural Networks at Any Scale" [Pet+25]

Extensions

How to regularize? Weight decay:

$$x^{k+1} = (1-\lambda)x^k + \lambda\beta\,\mathrm{lmo}(d^k)$$

- Interpretation: Frank-Wolfe for constrained problems
- Iterates are constrained in the *chosen norm* (e.g., Spectral for Muon [Pet+25; Liu+25])
- $\beta > 0$ dictates the radius of the constraint, $\|x^k\| \leq \beta$

How to ensure descent?

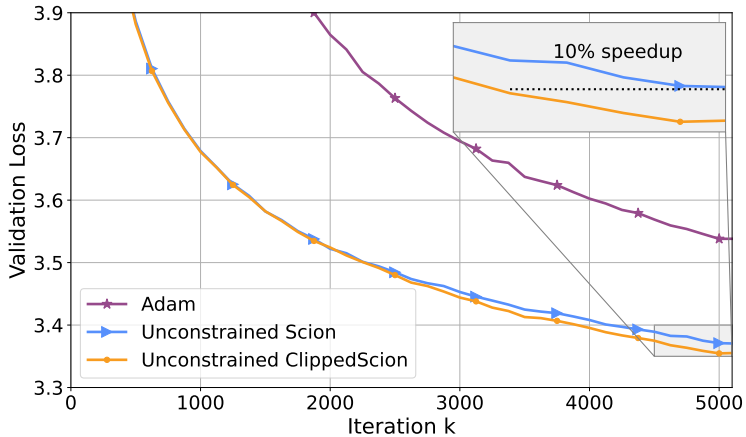
Frank-Wolfe short step

$$v^k = x^k - \beta\,\mathrm{lmo}(d^k)$$

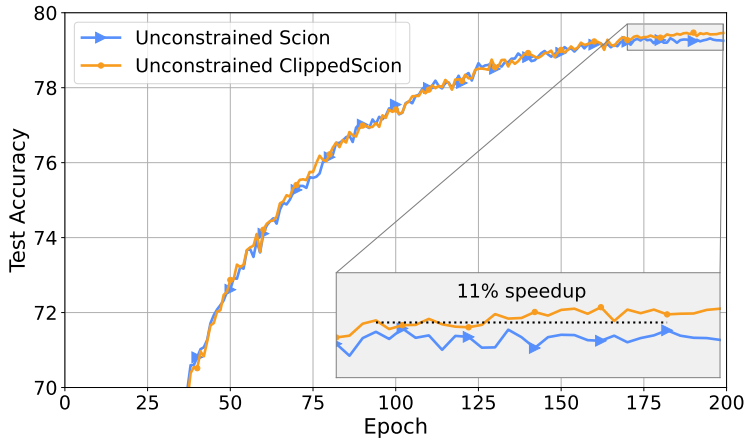
$$x^{k+1} = x^k - \gamma\eta_k v^k \quad \text{with} \quad \eta_k = \min\{\rho, \tfrac{\langle d^k, v^k\rangle}{\|v^k\|^2}\}$$

Experiments

NanoGPT 1B



DeiT-base

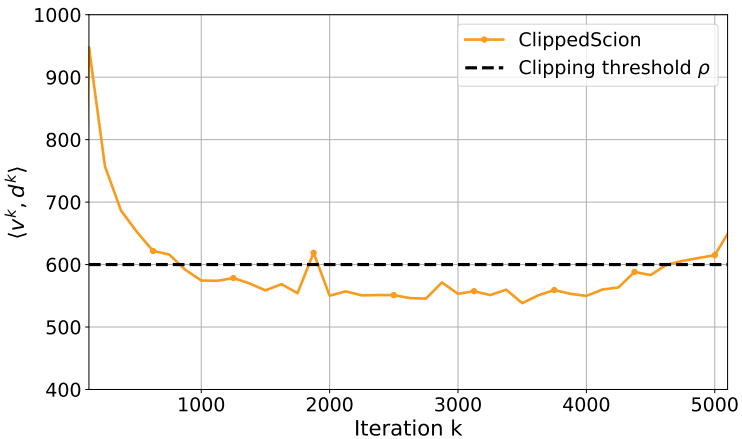


Norm behaviour

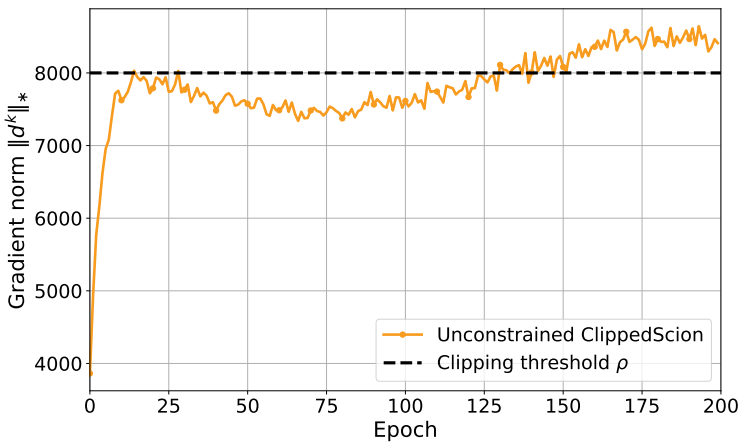
Expect Decrease of first order stationarity conditions (e.g., grad. norm)

In practice: Strange norm behaviour

NanoGPT with linear lr decay*



DeiT-base



* Partially investigated with AdamC paper [Def25].

Some shout-outs

Model truncation [Cra+25]

$$x^{k+1} = \underset{\|x-x^k\| \leq \rho}{\arg\min} \ \max\{\tilde{f}_k + \langle d^k, x - x^k \rangle, f^\star\} = x^k + \min\{\rho, \tfrac{\tilde{f}_k - f^\star}{\|d^k\|_*}\} \operatorname{lmo}(d^k)$$

- Incorporates knowledge of $f^\star$ to avoid tuning stepsize γ
- Also uses product norm over layers

Non-Euclidean (L_0, L_1)-smoothness empirically verified [Ria+25]

- Assumption holds along the optimization trajectory
- Based on the same norm choice: (Sign $\to$ Spectral $\to$ Sign)

Outlook

Towards parameter-agnosticity

- Clipping decouples L_0 and L_1
- L_0 (easy to adapt to) and L_1 (seems hard to adapt to [Hüb+24])

Towards a smoothness model for neural networks

Modular norm [Lar+24] If $\|x\| \leq 1$ then L-Lipschitz in the same norm.

- In practice, constraint seem to be for *regularization* not *optimization*
- Derive (L_0, L_1)-smooth for unconstrained modular norm?

Find us

in Exhibit Hall C,D,E #909

from 4:30 p.m. - 7:30 p.m.

References i

[Car+15] David E Carlson et al. "Preconditioned spectral descent for deep learning". In: *Advances in neural information processing systems* 28 (2015).

[Cra+25] Michael Crawshaw et al. "An exploration of non-euclidean gradient descent: Muon and its many variants". In: *arXiv preprint arXiv:2510.09827* (2025).

[DD25] Damek Davis and Dmitriy Drusvyatskiy. "When do spectral gradient updates help in deep learning?" In: (2025).

[Def25] Aaron Defazio. "Why Gradients Rapidly Increase Near the End of Training". In: *arXiv preprint arXiv:2506.02285* (2025).

References ii

[Hüb+24] Florian Hübler et al. “Parameter-agnostic optimization under relaxed smoothness”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2024, pp. 4861–4869.

[Lar+24] Tim Large et al. “Scalable Optimization in the Modular Norm”. In: *arXiv preprint arXiv:2405.14813* (2024).

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

[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).

References iii

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

[Ria+25] Artem Riabinin et al. “Gluon: Making Muon & Scion Great Again!(Bridging Theory and Practice of LMO-based Optimizers for LLMs)”. In: *arXiv preprint arXiv:2505.13416* (2025).

[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).