NORMALIZE YOUR NORMALIZED LAYERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Normalization, both in the forward pass and in the optimizer, is a central component of modern deep learning. But where and how should we normalize? We study this question from a purely optimization-based perspective and find that the answer prescribes a placement of normalization layers consistent with state-of-the-art architectures. Exploiting the scale invariance induced by normalization layers, we develop NorSODA, a weight-normalized method that achieves noise-robust accelerated rates by carefully incorporating weight decay. Our key technical result is an online-to-batch conversion that wraps any no-regret algorithm while only making assumptions on the sphere. Interestingly, the result extends to non-Euclidean geometries, yielding a close cousin of MuonH that removes the need to normalize the update direction.

1 INTRODUCTION

Weight normalization is increasingly used in large-scale training of deep learning models to ensure training stability (Loshchilov et al., 2025; Wen et al., 2026; Xie et al., 2026), yet its interaction with optimization guarantees is not well understood. In this work we take an optimization-based perspective and ask:

When can we normalize weights without breaking convergence, and while maintaining optimal rates?

Our analysis is based on the structural assumption of *scale invariance* of the objective, which states that rescaling a parameter group by a positive scalar leaves the objective unchanged. This occurs, for instance, when the rescaling is absorbed by a subsequent normalization layer in which case weight normalization leaves the function value unchanged.

A recurring observation is that weight normalization can avoid the need for weight decay, as reported for nGPT (Loshchilov et al., 2025) and MuonH (Wen et al., 2026). As a warm-up, we show that weight decay can provably be removed for weight-normalized steepest descent. This, however, does not explain the presence of Nesterov momentum in modern methods.

Our main result closes this gap with a weight normalized method that enjoys optimal noise-robust accelerated rates by integrating Nesterov momentum. Standard formulations of Nesterov acceleration typically relies on assumptions at auxiliary points, which is at odds with an analysis confined to the sphere. We resolve this with a new online-to-batch conversion that only requires assumptions to hold on the sphere, and instantiate it with Optimistic Dual Averaging (Rakhlin & Sridharan, 2013) to obtain NorSODA. Remarkably, NorSODA, which is developed to attain a theoretical optimal rate, almost exactly recovers state-of-the-art methods such as MuonH.

Contributions We make the following contributions:

- We identify parameter groups for which normalization layers induce scale invariance. For example, post-sublayer norm and QK-normalization as used in the Gemma and OLMo series renders all weights in attention and MLP blocks fully scale invariant for homogeneous activation functions. Conversely, output layers and the non-homogeneous gate projection of SwiGLU blocks are generally not scale invariant (Section B).
- Under scale invariance, we show convergence in function-value for weight-normalized steepest descent under Hölder smoothness and star-convexity without weight decay. The normalization step automatically implies boundedness, which is essential to establishing the result without weight decay.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- Reintroducing weight decay, we derive our main result, which is a tight online-to-batch conversion that wraps any online no-regret algorithm in a normalized averaged sequence (Lemma 5.3). Remarkably, the proof consists almost exclusively of equalities. Instantiating the conversion with Optimistic Dual Averaging yields noise-robust accelerated rates for finite $p \geq 2$ norms, including Schatten-p geometries for matrix parameters, while keeping gradient queries and objective assumptions on the sphere.
- For $p \to \infty$ NorSODA recovers a variant of MuonH, which avoids normalizing the update direction. Empirically, we show that update normalization can be dropped entirely without hurting accuracy. A practical NorSODA variant improves over MuonH across training horizons using the same MuonH-tuned hyperparameters.

2 Related work

Layer normalization Modern architectures have noted the importance of normalization placement for training stability. In particular, recent architectures such as Chameleon and OLMo 2 adopt norm reordering together with QK-normalization to control norm growth during training (Chameleon Team, 2024; Team OLMo et al., 2025). OLMo 3 continues this design choice (Olmo Team, 2025). The reordering places the normalization *after* the block (called "reordered" or "post-sublayer" norm)

$$h \leftarrow h + \mathsf{RMSNorm}(\mathsf{Block}(h)).$$

This reordering was originally introduced in Swin Transformer V2 (Liu et al., 2021), while QK-normalization was used as a key ingredient for stable large-scale vision transformer training (Dehghani et al., 2023). The Gemma series similarly normalizes after the attention and feedforward sublayers, and adds QK-normalization from Gemma 3 onward (Gemma Team, 2024; 2025; 2026). In nGPT (Loshchilov et al., 2025), RMSNorm is similarly applied after the block, but differs by additionally applying RMSNorm after the residual update instead of before the block as in the Gemma series. The nGPT model also optionally performs rowwise normalization of weight matrices on the forward pass, which renders the matrices scale-invariant row-wise.

Interestingly, this post-block normalization placement directly gives the scale invariance of weight matrices exploited in our work. More specifically, all hidden matrices of the attention and MLP blocks (query, key, value, output, and MLP projections) satisfy the scale invariance property in these architectures.

Weight normalization Spectral weight normalization was introduced in the context of Generative Adversarial Networks (GANs), originally motivated by the 1-Lipschitz constraint required by the Wasserstein GAN formulation (Miyato et al., 2018), and later adopted more broadly as a stability technique. Recently, there has been a flurry of work on weight normalization. MuonH (Wen et al., 2026) uses Frobenius normalization of both the update and the weights, with extensions such as HyperP (Ren et al., 2026). SpectralSphere (Xie et al., 2026) instead uses the spectral norm. Other approaches such as weight normalization reparameterization (Salimans & Kingma, 2016) and nGPT (Loshchilov et al., 2025) use a Frobenius and row-wise normalization scheme inside the architecture itself instead of the optimizer. Instead of using backpropagation this type of reparameterization can also be performed inside the optimizer itself as exploited recently in Hägele et al. (2026); Hübler et al. (2026a).

The original Muon implementation in the CIFAR-10 speedrun (Jordan, 2024) also uses Frobenius weight normalization for the Conv2D weights. This observation aligns with our framework because the BatchNorm-based CNN network used there is scale invariant for those parameter groups. A variant of Muon, called half-split CompositionalMuon with isotropic approximation (Keigwin et al., 2026), divides the matrix sign update in Muon by the Frobenius norm of the partner weight matrix. Interestingly, if weight normalization is employed, then this scaling becomes a no-op.

Online-to-batch and acceleration Accelerated rates can be obtained by converting optimistic online learners into offline methods (Wang & Abernethy, 2018; Cutkosky, 2019; Kavis et al., 2019; Joulani et al., 2020). Our conversion is a normalized version of anytime online-to-batch (Cutkosky, 2019), in which gradients are queried at the normalized average. Since $\mathsf{N}_{\omega,\rho}$ is a retraction, a Riemannian view is also possible, but Riemannian acceleration relies on geodesic convexity (Zhang & Sra, 2018; Criscitiello & Boumal, 2022), which we avoid.

3 PRELIMINARIES

The core structural assumption in this work is that the loss is insensitive to positive rescaling of each parameter group.

Assumption 3.1 (Scale invariance). *Write the parameters as $x = (x_1, \dots, x_G)$ and let $\mathcal{I} \subseteq \{1, \dots, G\}$ be the set of normalized groups. The objective f is differentiable on the domain where $x_i \neq 0$ for all $i \in \mathcal{I}$, and*

$$f(y) = f(x) \quad \textit{whenever} \quad y_i = a_i x_i$$

with $a_i > 0$ for $i \in \mathcal{I}$ and $y_i = x_i$ for $i \notin \mathcal{I}$.

This assumptions is interesting to us because it is satisfied by a large family of modern deep learning models due to the presence of normalization layers.

Example 3.1 (Normalized networks). *This assumption is satisfied groupwise by standard normalization layers. For example, let $x = (W_1, W_2)$. When σ is positively homogeneous (e.g., ReLU), the map*

$$z \mapsto \mathsf{RMSNorm}(\mathsf{MLP}(z)), \quad \textit{where} \quad \mathsf{MLP}(z) = W_2\sigma(W_1 z)$$

is invariant to positive rescaling of either W_1 or W_2, since the normalization removes the scale of its input. Consequently, any loss depending on the network only through this normalized output is scale invariant with respect to those weight blocks.

Likewise, QK-normalization makes the query and key projections scale invariant (even per head), while a post-sublayer normalization makes the value and output projections scale invariant. In all cases the normalization can be applied per group, so no global normalization over all parameters is required. See Section A for a more detailed overview of various normalization layers and more sophisticated activation functions such as SwiGLU.

> **Insight 3.1**. From an optimization perspective, a normalization layer placed right after the MLP blocks using homogeneous activation functions and after attention blocks (known as "reordered" or "post-sublayer" norm) makes all hidden matrices scale invariant and hence amenable to weight normalization. This placement is consistent with modern architectures such as OLMo and Gemma.

Limitations We are only providing sufficient condition and not necessary conditions for placing a normalization layer. We are *not* claiming that normalization layers should not be place anywhere else. In particular, pre-norm is also used in our experiments, which have other justifications (Xiong et al., 2020).

The following consequences are used throughout, which is a block-wise version of Roburin et al. (2022, Lemma 1).

Lemma 3.2. *Suppose Assumption 3.1 holds. Then, for every group $i \in \mathcal{I}$ and every $a > 0$,*

$$a\nabla_i f(x_1, \dots, ax_i, \dots, x_G) = \nabla_i f(x), \qquad \textit{and} \qquad \langle \nabla_i f(x), x_i \rangle = 0.$$

Consequently, if $y_i = a_i x_i$ for some $a_i > 0$, $i \in \mathcal{I}$, and $y_i = x_i$ for $i \notin \mathcal{I}$, then

$$f(y) = f(x), \qquad \langle \nabla f(x), y - x \rangle = 0.$$

Algorithmic primitives Let $\mathcal{X}$ be a finite-dimensional normed space with dual $\mathcal{X}^*$. For a norm $\|\cdot\|$ on the primal space, define the dual norm by

$$\|g\|_* := \sup_{\|x\| \leq 1} \langle g, x \rangle .$$

Let ω_i be norms used for normalization of the groups $i \in \mathcal{I}$ (possibly different from $\|\cdot\|$), with radii $\rho_i > 0$, and write $\omega = (\omega_i)_{i \in \mathcal{I}}$ and $\rho = (\rho_i)_{i \in \mathcal{I}}$. Define the product sphere and the groupwise weight-normalization map by

$$\mathcal{S}_{\omega,\rho} := \{x : \omega_i(x_i) = \rho_i,\ i \in \mathcal{I}\}, \qquad \mathsf{N}_{\omega,\rho}(x)_i := \begin{cases} \frac{\rho_i}{\omega_i(x_i)} x_i, & i \in \mathcal{I}, \\ x_i, & i \notin \mathcal{I}. \end{cases}$$

By Assumption 3.1, $f \circ \mathsf{N}_{\omega,\rho} = f$.

4 WARMUP: WEIGHT NORMALIZED STEEPEST DESCENT

It is well known that (normalized) steepest descent requires weight decay in order to establish convergence in function values (see e.g., Kovalev (2025, Sec. 4) and Mokhtari et al. (2020, Thm. 3)). Essentially, in order to establish theoretical guarantees, the approach is to instead run Frank-Wolfe on a ball with a carefully chosen radius that depends on the desired accuracy, the distance to the solution, and the Lipschitz constant (Kovalev, 2025, Cor. 3).

In this section we show that we can do away with weight decay by instead using weight normalization when the scale-invariance property (Assumption 3.1) holds. This has the added benefit of only requiring assumptions to hold on the chosen sphere.

Define the linear minimization oracle (lmo) on the unit ball by

$$\mathrm{lmo}(d) \in \arg\min_{\|s\| \leq 1} \langle d, s \rangle .$$

We consider the weight-normalized lmo-based update

$$\begin{aligned} s^k &= \mathrm{lmo}(\nabla f(\bar{x}^k)), \\ \bar{x}^{k+1} &= \mathsf{N}_{\omega,\rho}\left(\bar{x}^k + \gamma_k s^k\right). \end{aligned} \tag{1}$$

Without the normalization step, this update rule goes by many names, including normalized steepest descent (Boyd et al., 2004), unconstrained conditional gradient (Pethick et al., 2025), trust-region gradient method (Kovalev, 2025), and continuous greedy when $x^0 = 0$ and $\gamma_k = 1/n$.

4.1 CONVERGENCE ANALYSIS

The scale-invariance property will allow us to make our assumption only on iterates lying on the chosen sphere. Specifically, we make the following assumptions.

Assumption 4.1 (Hölder smoothness)**.** *For the iterates generated by (1), the objective f is differentiable at each $\bar{x}^k$. There are constants $\nu \in (0, 1]$ and $L_\nu > 0$ such that, for every k,*

$$f(\bar{x}^k + \gamma_k s^k) \leq f(\bar{x}^k) + \gamma_k \langle \nabla f(\bar{x}^k), s^k \rangle + \tfrac{L_\nu}{1+\nu} \gamma_k^{1+\nu} \left\| s^k \right\|^{1+\nu} .$$

Assumption 4.2 (μ-KL)**.** *There is a reference value $f^\star$ and a constant $\mu > 0$ such that*

$$f(\bar{x}^k) - f^\star \leq \tfrac{1}{\mu} \left\| \nabla f(\bar{x}^k) \right\|_* .$$

Remark 4.1. This is the μ-KL condition of Islamov et al. (2026, Assumption 3.3). It is implied by star-convexity along the iterates (Assumption 5.1 with comparator $x^\star$) on a bounded domain: if $\left\| \bar{x}^k - x^\star \right\| \leq R$ for all normalized iterates, then $f(\bar{x}^k) - f^\star \leq \langle \nabla f(\bar{x}^k), \bar{x}^k - x^\star \rangle \leq R \left\| \nabla f(\bar{x}^k) \right\|_*$, so Assumption 4.2 holds with $\mu = \frac{1}{R}$. *A key observation is that normalizing the iterates onto $\mathcal{S}_{\omega,\rho}$ enforces this bounded domain, so we do not have to enforce it through weight decay.* Moreover, since $f(\mathsf{N}_{\omega,\rho}(x^\star)) = f(x^\star)$, the comparator may be chosen on $\mathcal{S}_{\omega,\rho}$, in which case $R \leq \mathrm{diam}(\mathcal{S}_{\omega,\rho})$ is known in advance when all groups are normalized.

Since Assumption 4.2 also holds for any $\mu' \leq \mu$, the following step size only requires a lower bound on μ, such as $1/\mathrm{diam}(\mathcal{S}_{\omega,\rho})$, and no knowledge of the smoothness constant.

Theorem 4.2 (Function-value convergence)**.** *Suppose Assumptions 3.1, 4.1 and 4.2 hold. If $\gamma_k = \frac{2}{\mu(k+2)}$, then for every $k \geq 1$,*

$$f(\bar{x}^k) - f^\star \leq \tfrac{C_\nu}{(k+1)^\nu}, \qquad C_\nu := \tfrac{2^{1+\nu} L_\nu}{(1+\nu)(2-\nu)\mu^{1+\nu}}.$$

Extension to the (heavy-tailed) stochastic case is straightforward by introducing a heavy-ball momentum based gradient estimator as in Mokhtari et al. (2020); Pethick et al. (2025). The error control in Hübler et al. (2026b, Lem. C.2) is directly applicable since it only relies on the gradient estimator and not the update itself. We do not pursue this here, since in the next section we will instead show that Nesterov momentum, when combined with weight decay, can simultaneously handle noise and accelerate convergence.

5 ACCELERATION AND ONLINE-TO-BATCH CONVERSION

In this section we reintroduce weight decay and use it to accelerate convergence. The key difficulty in extending normalization to accelerated methods is that Nesterov acceleration has a remarkable number of forms, including primal momentum or extrapolation schemes, encompassing Nesterov's original AGM1 form and its look-ahead momentum variants (Nesterov, 2013; Sutskever et al., 2013); one-prox Bregman schemes (Auslender & Teboulle, 2006; Tseng, 2008); linear coupling, or two-prox, schemes (Allen-Zhu & Orecchia, 2014; Bansal & Gupta, 2017; Lan & Zhou, 2018); and accelerated dual-averaging, or cumulative-model, schemes (Nesterov, 2013; Tseng, 2008). Furthermore, this non-exhaustive list does not even include stochastic extensions. It is thus not immediately clear which formulation of Nesterov acceleration is most amenable to incorporate normalization.

In particular, the usual AGM1 form of Nesterov acceleration, involving a convex combination of the previous iterate and a gradient step, turns out to not be well suited to our normalized setting. The main issue is that the standard argument relies on assumptions that do not only involve the (normalized) averaged iterate. Instead, we consider an approach based on online-to-batch conversion, which fortunately is also much more general. In essence, by reintroducing weight decay, the modified scheme can achieve an accelerated convergence rate while still only making assumptions on the sphere.

5.1 METHOD

In order to appropriately place the normalization operation, it turns out to be useful to clearly distinquish between the primal and the dual. Let $\mathcal{X}$ be a finite-dimensional normed space with dual $\mathcal{X}^*$. We rely on the Fenchel conjugate to map gradient information back to the primal space. For a proper closed convex reference function $h : \mathcal{X} \to \mathbb{R} \cup \{\infty\}$, the Fenchel conjugate over $\mathcal{X}$ is defined as

$$h^*(m) = \sup_{x \in \mathcal{X}} \{\langle m, x\rangle - h(x)\}, \qquad m \in \mathcal{X}^*.$$

This dualization also conveniently allows us to interpolate between a Euclidean SGD-like method and a Schatten-∞-based method such as Muon (Jordan et al., 2024b) by generalizing to general p-norms as captured by the following assumption.

Definition 5.1 (p-uniform convexity)**.** *For $p \geq 2$, a proper closed convex function $h : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ is p-uniformly convex with constant $\mu_h > 0$ with respect to $\|\cdot\|$ if, for all $u, v \in \mathcal{X}$ and $s \in \partial h(v)$,*

$$D_h(u, v; s) := h(u) - h(v) - \langle s, u - v\rangle \geq \tfrac{\mu_h}{p} \|u - v\|^p .$$

The central object of study is a normalized version of the SODA framework (Pethick et al., 2026), which we call NorSODA. Let $\mathsf{N}_{\omega,\rho}$ be the normalization map from Section 3. Initialize $m^0 = 0$, $z^0 \in \partial h^*(0)$, and $\bar{x}^0 = \mathsf{N}_{\omega,\rho}(z^0)$. For $k \geq 0$,

$$\begin{aligned}
m^{k+1} &= (1 - \alpha_k)m^k + \alpha_k \nabla f(\bar{x}^k, \xi_k), \\
\bar{m}^{k+1} &= (1 - \bar{\alpha}_k)m^{k+1} + \bar{\alpha}_k \nabla f(\bar{x}^k, \xi_k), \\
z^{k+1} &\in \partial h^*(-\gamma_k \bar{m}^{k+1}) = \arg\min_{z \in \mathcal{X}} \left\{\gamma_k \langle \bar{m}^{k+1}, z\rangle + h(z)\right\}, \\
\bar{x}^{k+1} &= \mathsf{N}_{\omega,\rho}((1 - \lambda_k)\bar{x}^k + \lambda_k z^{k+1}) \\
&= \mathsf{N}_{\omega,\rho}(\bar{x}^k + \beta_k z^{k+1})
\end{aligned} \tag{NorSODA}$$

Here $\alpha_k, \bar{\alpha}_k, \lambda_k \in [0, 1]$, $\gamma_k > 0$ and $\beta_k = \frac{\lambda_k}{1-\lambda_k}$. Crucially, all gradients are queried at the normalized iterates. The last equality holds since $\mathsf{N}_{\omega,\rho}$ is invariant to positive rescaling of the normalized groups (on groups outside $\mathcal{I}$ the first form is used). Thus the contraction $(1 - \lambda_k)$, which acts like weight decay, is absorbed by the normalization into the step size β_k.

Schatten-p. For matrix-valued parameters, a canonical unconstrained choice is

$$h(X) = \tfrac{1}{p} \|X\|_{S_p}^p , \qquad p > 1.$$

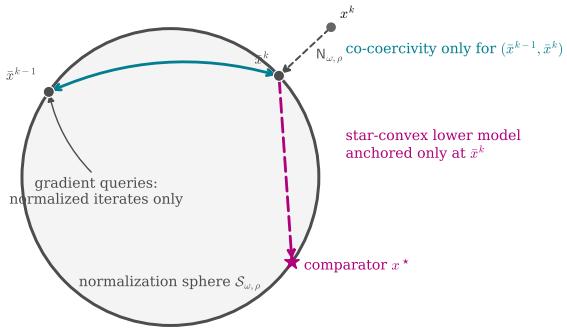


Figure 1: The assumptions on the objective f only need to hold along the iterates on the sphere.

Let $\bar{M} = U\,\mathrm{diag}(s)V^\top$, $s_i \geq 0$, be a singular value decomposition. Then, with $q = p/(p-1)$, the corresponding mirror map is

$$\partial h^*(\bar{M}) = \arg\max_X \left\{ \langle \bar{M}, X\rangle - \tfrac{1}{p}\,\|X\|_{S_p}^p \right\} = U\,\mathrm{diag}\left(s_i^{q-1}\right) V^\top.$$

Insight 5.1. NorSODA is closely related to MuonH, which corresponds to the limit $p \to \infty$ with Frobenius normalization $\mathsf{N}_{\omega,\rho}$, constant momentum parameters $\alpha_k = \bar{\alpha}_k = \alpha$ and stepsize β_k using linear decay. The main difference is that NorSODA does not normalize the update direction z^{k+1}. Figure 2 shows that this normalization can be removed without loss.

5.2 Online-to-batch

The analysis uses the following normalized online-to-batch conversion. Define the Bregman divergence

$$D_f(u, v) := f(u) - f(v) - \langle \nabla f(v), u - v\rangle\,. \tag{2}$$

We will only need our assumptions to hold on the sphere $\mathcal{S}_{\omega,\rho}$, specifically for the iterates $\bar{x}^k \in \mathcal{S}_{\omega,\rho}$. The assumptions are illustrated in Figure 1.

Assumption 5.1 (Star-convexity along the iterates)**.** *For the comparator $x \in \mathcal{X}$ and the iterates $\bar{x}^k \in \mathcal{S}_{\omega,\rho}$ generated by (3) (e.g., by NorSODA),*

$$f(x) \geq f(\bar{x}^k) + \langle \nabla f(\bar{x}^k), x - \bar{x}^k\rangle\,, \qquad k = 0, \ldots, n-1.$$

Assumption 5.2 (Hölder co-coercive along the iterates)**.** *There exist $\nu \in (0, 1]$ and $L_\nu > 0$ such that, with $r_\nu := (1+\nu)/\nu$, for $k = 1, \ldots, n-1$, the iterates $\bar{x}^k \in \mathcal{S}_{\omega,\rho}$ generated by (3) satisfy*

$$D_f(\bar{x}^{k-1}, \bar{x}^k) \geq \tfrac{1}{r_\nu L_\nu^{1/\nu}} \left\|\nabla f(\bar{x}^k) - \nabla f(\bar{x}^{k-1})\right\|_*^{r_\nu}.$$

Remark 5.2 (Sufficient condition). Define

$$w^k \in \arg\min_{w \in \mathcal{X}} \left\{ \langle \nabla f(\bar{x}^{k-1}) - \nabla f(\bar{x}^k), w - \bar{x}^{k-1}\rangle + \tfrac{L_\nu}{1+\nu} \left\|w - \bar{x}^{k-1}\right\|^{1+\nu} \right\}.$$

It suffices that convexity holds for $(\bar{x}^k, w^k)$ and that Hölder smoothness (Assumption 4.1) holds from $\bar{x}^{k-1}$ to w^k.

Assumption 5.3 (Unbiased)**.** *Let $\mathcal{F}_k$ be the natural filtration. The stochastic gradients satisfy*

$$\mathbb{E}[g^k \mid \mathcal{F}_{k-1}] = \nabla f(\bar{x}^k).$$

We are now ready to state our key result, which is an online-to-batch conversion result, that takes in an (optimistic) no-regret online learning algorithm and turns it into an accelerated offline algorithm whos iterates are restrained to the sphere. The result builds on Pethick (2026), but crucially only maintains a single averaged sequence that is in turn normalized.

Lemma 5.3 (Online-to-batch Hölder refinement). *Let an online no-regret algorithm generate $z^k = \mathcal{A}_k(g^0, \dots, g^{k-1}) \in \mathcal{X}$ before receiving the feedback $g^k = \nabla f(\bar{x}^k, \xi_k) \in \mathcal{X}^*$. Use the normalization map $\mathsf{N}_{\omega,\rho}$ defined in Section 3. Initialize $\bar{x}^0 = \mathsf{N}_{\omega,\rho}(z^0)$. Given positive weights $a_0, \dots, a_{n-1}$, define*

$$x^k = (1 - \lambda_{k-1})\bar{x}^{k-1} + \lambda_{k-1} z^k, \qquad \bar{x}^k = \mathsf{N}_{\omega,\rho}(x^k), \tag{3}$$

where

$$\lambda_{k-1} = \tfrac{a_k}{A_k}, \qquad A_k = \textstyle\sum_{i=0}^{k} a_i.$$

Suppose Assumptions 3.1 and 5.1 to 5.3 hold for a fixed comparator $x \in \mathcal{X}$. Then

$$\begin{aligned} A_{n-1}\mathbb{E}[f(\bar{x}^{n-1}) - f(x)] \le \mathbb{E}\left[\textstyle\sum_{k=0}^{n-1} a_k \langle g^k, z^k - x\rangle\right] \\ - \tfrac{1}{r_\nu L_\nu^{1/\nu}} \textstyle\sum_{k=1}^{n-1} A_{k-1}\mathbb{E}\left[\left\|\nabla f(\bar{x}^k) - \nabla f(\bar{x}^{k-1})\right\|_*^{r_\nu}\right]. \end{aligned} \tag{4}$$

What is remarkable about this lemma is the tightness of the inequality. Essentially the entire proof consists of equalities except for the one step where Assumption 5.2 is invoked (*c.f.* Section D.2). This yields a tighter bound than the one in Pethick (2026), while extending to weight normalization.

Equation (4) provides a way to extract offline convergence guarantees from any no-regret algorithm while ensuring that all gradients are queried only at normalized iterates. We next use this to obtain a noise-robust accelerated rate by combining it with Optimistic Dual Averaging (ODA).

5.3 Acceleration

In this section we show noise-robust acceleration of NorSODA by exploiting that the method can be seen as ODA wrapped in the online-to-batch meta-algorithm from Lemma 5.3. We make the following heavy-tailed noise assumption.

Assumption 5.4 (Gradient variation). *For some $q \in (1, 2]$, the gradients satisfy, for $k \ge 0$, $\tau_q \ge 1$, and $\sigma_q \ge 0$,*

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

For $k = 0$, use the convention $\bar{x}^{-1} = x^\star$.

Theorem 5.4 (Accelerated NorSODA under Hölder smoothness). *Let $\nu \in (0, 1]$, let $p \ge 2$ with $p > 1 + \nu$, and let $q = p/(p-1)$. Consider NorSODA with $h(x) = \frac{1}{p}\left\|x - z^0\right\|_p^p$, let $x^\star \in \arg\min_x f(x)$, and let $R_p := \left\|x^\star - z^0\right\|_p$. For every $k = 0, \dots, n-1$, choose*

$$\alpha_k = \tfrac{2}{k+2}, \qquad \bar{\alpha}_k = \lambda_k = \tfrac{2}{k+3}, \qquad \gamma_k = \eta\tfrac{(k+2)(k+3)}{2}.$$

Suppose Assumptions 3.1 and 5.1 to 5.4 hold with $\|\cdot\| = \|\cdot\|_p$, this value of q, and comparator $x^\star$. Choose

$$\eta \asymp_{p,\nu} \min\left\{\tfrac{R_p^{p-1-\nu}}{\tau_q^{1+\nu} L_\nu n^{2-\kappa_{\mathrm{acc}}}}, \tfrac{R_p^{p-1}}{\sigma_q\, n^{(2p-1)/p}}\right\} \quad \text{with} \quad \kappa_{\mathrm{acc}} := \tfrac{1+(p+1)\nu}{p}.$$

Then, for every $n \ge 1$,

$$\mathbb{E}[f(\bar{x}^{n-1}) - f(x^\star)] = O_{p,\nu}\left(\tfrac{\tau_q^{1+\nu} L_\nu R_p^{1+\nu}}{n^{\kappa_{\mathrm{acc}}}} + \tfrac{\sigma_q R_p}{n^{1/p}}\right).$$

Remark 5.5 (Replacing R_p by the sphere radius). Since $f(\mathsf{N}_{\omega,\rho}(x^\star)) = f(x^\star)$, the comparator can be taken on $\mathcal{S}_{\omega,\rho}$, so the hard-to-estimate R_p appearing in the stepsize can be replaced by the known radius through $R_p \le 2\rho$. This relationship tell us that the scale-invariance should ideally match the problem geometry. This choice trivially side-steps the difficulty of an unknown distance to the solution, which is what the literature on parameter-free methods attempts to address (Orabona & Pál, 2016). Being adaptive to σ_q and L_ν as well is an interesting direction that we leave for future work.

6 Experiments

Setup. We use the modded-nanoGPT speed-run (Jordan et al., 2024a) which uses a 12-layer, width-768 transformer with rotary embeddings, QK RMSNorm, ReLU-squared MLPs, and a tied token embedding/output head. Training uses 1,024-token sequences from FineWeb10B, a global batch

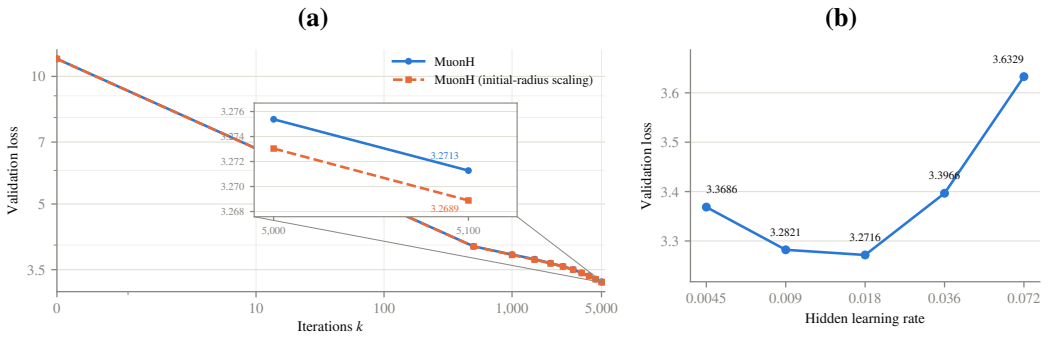


Figure 2: **(a)** MuonH with Frobenius normalization update directions versus static scaling based on the initial radius. Replacing the normalization by a static scale matches MuonH while reducing the per-step computational cost. **(b)** MuonH learning rate tuning on the sandwich-RMSNorm model. The head learning rate is scaled proportionally to the hidden rate, and all other settings are fixed.

size of 512, and, unless stated otherwise, 5,100 updates (our $1\times$ Chinchilla horizon). We replace the original residual normalization by *sandwich RMSNorm*,

$$h \leftarrow h + \mathsf{RMSNorm}\big(\mathsf{Block}(\mathsf{RMSNorm}(h))\big),$$

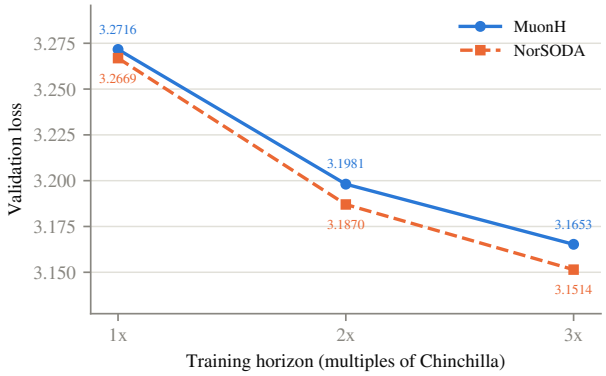
for both the attention and MLP branches. This makes the six hidden matrices in each block scale invariant, so $\mathcal{I}$ in Assumption 3.1 consists of the hidden matrices, while the tied embedding/output head is not scale invariant.

Optimizers. The baseline optimizer, MuonH (Wen et al., 2026), uses momentum 0.95, a 12-step Newton–Schulz polar direction for each hidden matrix, no weight decay, and a linear-decay learning-rate schedule. The tied head is updated with Lion without Frobenius weight normalization. Since this is a sign update, we graft the tuned output learning rate from Scion (Pethick et al., 2025), and then tune a single global multiplier that scales both hidden and head learning rates. A sweep confirms that the hidden learning rate 0.018 of the tuned MuonH modded-nanoGPT submission remains optimal for the modified architecture (Figure 2(b)).

NorSODA inherits all tuned MuonH hyperparameters including the learning rates, momentum, Newton–Schulz steps, and linear decay without further tuning. Instead of Frobenius normalizing the update direction as in MuonH, for an $m \times n$ hidden matrix, we apply the usual Muon scaled Newton–Schulz direction and then additionally scale it by $R_0/\sqrt{m}$ with R_0 being the Frobenius norm of the initialized weights. Relative to the raw Newton–Schulz output, this corresponds to scaling the update by $R_0/\sqrt{\min(m,n)}$. This calibration lets us transfer MuonH's scalar learning rate to NorSODA without learning rate retuning. NorSODA does not normalize the final head, where there is no scale invariance to exploit, and anchors the head at its initialization (z^0 in NorSODA) following (Pethick et al., 2026). We make the practical choice of $\lambda_k = 2/(k+\Omega)$ with $\Omega = 20000$, which asymptotically satisfies the theoretical requirements. The Ω is not picked particularly carefully and can be further tuned (*c.f.* Figure 4(b)).

Removing update normalization. MuonH additionally divides every hidden polar direction by its measured Frobenius norm. To test whether this is needed, we replace that normalization by the static initialization scale $R_0/\sqrt{\min(m,n)}$ while keeping the weight normalization and every hyperparameter fixed. As shown in Figure 2(a), the static variant reaches validation loss 3.269, compared with 3.271 for MuonH. The update direction normalization is therefore unnecessary here and slightly hurts performance, consistent with our formulation which only normalizes the iterates. This simplification should not be seen as too surprising given that the Frobenius norm of a perfectly orthogonalize matrix is $\sqrt{\min(m,n)}$.

NorSODA improves with training horizon. We compare MuonH and NorSODA at $1\times$, $2\times$, and $3\times$ Chinchilla training in Figure 3, setting the linear decay schedule based on the corresponding horizon while keeping all other hyperparameters fixed. NorSODA performs better at every horizon,



$1\times$ **Chinchilla (3 seeds)**

Method	Mean $\pm$ s.d.
MuonH	3.2695 ± 0.0018
NorSODA	$\mathbf{3.2646} \pm 0.0022$
Difference	-0.0049 ± 0.0010

Differences are paired by seed.

Figure 3: Validation loss as the training horizon increases. NorSODA uses the same MuonH-tuned hyperparameters and improves relative to MuonH with longer training. The table shows that the conclusions are consistent across three seeds.

and its advantage grows from 0.005 to 0.011 and 0.014 validation loss. We test across three seeds to ensure that the conclusions are not explained by noise and find that NorSODA is indeed consistently better across all seeds. To further investigate this gap, we use the empirically derived scaling rules for the learning rate of MuonH developed by Marin and use the for the $2\times$ and $3\times$ Chinchilla horizons (*c.f.* Figure 4(a)). We find that this partially closes the gap.

7 Conclusion

We have provided a theoretical basis for the combination of Nesterov acceleration and weight normalization in MuonH by showing that a proper parameterization enjoys a noise-robust accelerated rate while only requiring assumptions to hold on the sphere. We then used this connection to further simplify MuonH by avoiding Frobenius normalization the update direction.

There is a whole range of questions that opens up from this theoretical perspective. Is it possible to develop a practical *anytime* method similar to the schedule-free wrapper (Defazio et al., 2024) so we not have to know the training horizon beforehand? Additionally, our Theorem 5.4 suggests scheduling the momentum parameters α_k and $\bar{\alpha}_k$ (in tandem with the stepsize β_k) in a way that is asymptotically equivalent to logarithmic-time schedules (Ferbach et al., 2026), which has shown strong empirical performance. Is it possible to gain similar improvements for weight normalized schemes? Lastly, we have seen theoretically how normalization alleviates the need for knowing the initial distance to the solution (Remark 5.5). Can we further develop a fully parameter-agnostic method that also do not need knowledge of L and σ?

Reproducibility Statement

Complete proofs of all theoretical results are given in the appendix. The architecture, optimizer hyperparameters, and training horizons are listed in Section E, and the practical NorSODA update is described in Section 6. We will release the code for NorSODA and all experiments upon publication.

AI Use Statement

In this work, we used generative AI tools for generating the plots, figures and their accompanying code and in helping generalize the result to Hölder smoothness. It was also extensively used for rearranging and improving the presentation by streamlining notation. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*, 2014. (cited on p. 5)

Alfred Auslender and Marc Teboulle. Interior gradient and proximal methods for convex and conic optimization. *SIAM J. Opt.*, 16(3):697–725, 2006. (cited on p. 5)

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL [url]. (cited on p. 13)

Nikhil Bansal and Anupam Gupta. Potential-function proofs for first-order methods. *arXiv:1712.04581*, 2017. (cited on p. 5)

S. Boyd, S.P. Boyd, L. Vandenberghe, and Cambridge University Press. *Convex Optimization*. Berichte uber verteilte messysteme. Cambridge University Press, 2004. ISBN 9780521833783. URL [url]. (cited on p. 4)

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. URL [url]. (cited on p. 2)

Christopher Criscitiello and Nicolas Boumal. Negative curvature obstructs acceleration for strongly geodesically convex optimization, even with exact first-order oracles. In *Conference on Learning Theory*, 2022. (cited on p. 2)

Ashok Cutkosky. Anytime online-to-batch, optimism and acceleration. In *International conference on machine learning*, pp. 1446–1454. PMLR, 2019. (cited on p. 2)

Aaron Defazio, Xingyu Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. In *Advances in Neural Information Processing Systems*, volume 37, pp. 9974–10007, 2024. doi: 10.52202/079017-0320. URL [url]. (cited on p. 9)

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. *arXiv preprint arXiv:2302.05442*, 2023. URL [url]. (cited on p. 2)

Damien Ferbach, Courtney Paquette, Gauthier Gidel, Katie Everett, and Elliot Paquette. Logarithmic-time schedules for scaling language models with momentum. *arXiv preprint arXiv:2602.05298*, 2026. (cited on p. 9)

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL [url]. (cited on p. 2)

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL [url]. (cited on p. 2)

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. URL [url]. (cited on p. 2)

Alexander Hägele, Alejandro Hernández-Cano, Atli Kosson, and Martin Jaggi. Improving neural network training by decoupling the magnitude and direction of weight vectors. *arXiv preprint arXiv:2606.25971*, 2026. (cited on p. 2)

Florian Hübler, Kai Lion, Antonio Orvieto, and Niao He. Muown implicitly performs angular step-size decay. *arXiv preprint arXiv:2606.23637*, 2026a. (cited on p. 2)

Florian Hübler, Thomas Pethick, and Suvrit Sra. Free heavy-tailed lunch for muon: A theoretical justification of empirical success. *arXiv preprint arXiv:2606.14560*, 2026b. (cited on p. 4)

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456. JMLR.org, 2015. (cited on p. 13)

Rustem Islamov, Roman Machacek, Aurelien Lucchi, Antonio Silveti-Falls, Eduard Gorbunov, and Volkan Cevher. On the role of batch size in stochastic conditional gradient methods. *arXiv preprint arXiv:2603.21191*, 2026. (cited on p. 4)

Keller Jordan. 94% on CIFAR-10 in 3.29 seconds on a single GPU. *arXiv preprint arXiv:2404.00498*, 2024. (cited on p. 2)

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 [url]. (cited on p. 7)

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cecista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024b. (cited on p. 5)

Pooria Joulani, Anant Raj, Andras Gyorgy, and Csaba Szepesvári. A simpler approach to accelerated optimization: iterative averaging meets optimism. In *International conference on machine learning*, pp. 4984–4993. PMLR, 2020. (cited on p. 2)

Ali Kavis, Kfir Y Levy, Francis Bach, and Volkan Cevher. Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization. *Advances In Neural Information Processing Systems 32 (Nips 2019)*, 32(CONF), 2019. (cited on p. 2)

Ben Keigwin, Li Yang, Yunzhe Zhang, Dhruv Pai, and Alec DeWulf. Towards compositional steepest descent. [url], 2026. Tilde Research blog. (cited on p. 2)

Dmitry Kovalev. Understanding gradient orthogonalization for deep learning via non-euclidean trust-region optimization. *arXiv preprint arXiv:2503.12645*, 2025. (cited on p. 4)

Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *Mathematical Programming*, 171(1–2):167–215, 2018. doi: 10.1007/s10107-017-1173-0. (cited on p. 5)

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer V2: Scaling up capacity and resolution. *arXiv preprint arXiv:2111.09883*, 2021. URL [url]. (cited on p. 2)

Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. ngpt: Normalized transformer with representation learning on the hypersphere. In *International Conference on Learning Representations*, volume 2025, pp. 74014–74038, 2025. (cited on pp. 1 and 2)

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. (cited on p. 2)

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *Journal of Machine Learning Research*, 2020. (cited on p. 4)

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer US, 2013. ISBN 9781441988539. URL [url]. (cited on p. 5)

Olmo Team. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025. URL [url]. (cited on p. 2)

Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. In *Advances in Neural Information Processing Systems*, 2016. (cited on p. 7)

Thomas Pethick. When to use what Schatten-p norm in deep learning? *arXiv preprint arXiv:2606.15268*, 2026. (cited on pp. 6, 7, 16, and 17)

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained LMOs. In *International Conference on Machine Learning*, 2025. (cited on pp. 4, 8, and 14)

Thomas Pethick, Wanyun Xie, Roman Machacek, and Volkan Cevher. Optimistic dual averaging unifies modern optimizers. *arXiv preprint arXiv:2605.11172*, 2026. (cited on pp. 5 and 8)

Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory*, pp. 993–1019. PMLR, 2013. (cited on p. 1)

Liliang Ren, Yang Liu, Yelong Shen, and Weizhu Chen. Rethinking language model scaling under transferable hypersphere optimization. *arXiv preprint arXiv:2603.28743*, 2026. (cited on p. 2)

Simon Roburin, Yann de Mont-Marin, Andrei Bursuc, Renaud Marlet, Patrick Perez, and Mathieu Aubry. Spherical perspective on learning with normalization layers. *Neurocomputing*, 487:66–74, 2022. (cited on p. 3)

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, 2016. (cited on p. 2)

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. URL [url]. (cited on p. 14)

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 1139–1147, 2013. (cited on p. 5)

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2025. URL [url]. (cited on p. 2)

Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2(3), 2008. (cited on p. 5)

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. URL [url]. (cited on p. 13)

Jun-Kun Wang and Jacob D. Abernethy. Acceleration through optimistic no-regret dynamics. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 3828–3838, 2018. URL [url]. (cited on p. 2)

Kaiyue Wen, Xingyu Dang, Kaifeng Lyu, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them ii: Hyperball optimization. *arXiv preprint arXiv:2606.16899*, 2026. (cited on pp. 1, 2, 8, 14, and 19)

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision*, pp. 3–19, 2018. URL [url]. (cited on p. 13)

Tian Xie, Haoming Luo, Haoyu Tang, Yiwen Hu, Jason Klein Liu, Qingnan Ren, Yang Wang, Wayne Xin Zhao, Rui Yan, Bing Su, et al. Controlled llm training on spectral sphere. *arXiv preprint arXiv:2601.08393*, 2026. (cited on pp. 1 and 2)

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pp. 10524–10533. PMLR, 2020. (cited on p. 3)

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *arXiv preprint arXiv:1910.07467*, 2019. URL [url]. (cited on p. 13)

Hongyi Zhang and Suvrit Sra. Towards Riemannian accelerated gradient methods. *arXiv preprint arXiv:1806.02812*, 2018. (cited on p. 2)

APPENDIX CONTENTS

A Normalization-induced scale invariance **13**

A.1 Normalization layers and scale invariance 13

A.2 SwiGLU blocks and scale invariance 14

A.3 Scale-invariance identities 14

B Unembedding layer **14**

C Weight-normalized steepest descent **15**

D Accelerated normalized SODA **15**

D.1 Method 15

D.2 Online-to-batch 16

D.3 Acceleration 16

E Experimental hyperparameters **19**

F Additional optimizer ablations **19**

A NORMALIZATION-INDUCED SCALE INVARIANCE

A.1 NORMALIZATION LAYERS AND SCALE INVARIANCE

Let $Y \in \mathbb{R}^{B\times C\times H\times W}$ denote the input to a normalization layer, where B is the batch dimension, C is the channel or feature dimension, and H, W are spatial or position dimensions. Vector activations are covered by setting unused spatial dimensions to one. A normalization layer partitions the coordinates of Y into normalization groups S. Ignoring the stabilizing constant ε, centered normalization uses

$$\mu_S(Y) = \tfrac{1}{|S|}\textstyle\sum_{i\in S} Y_i, \qquad \sigma_S(Y) = \left(\tfrac{1}{|S|}\textstyle\sum_{i\in S}(Y_i - \mu_S(Y))^2\right)^{1/2},$$

and maps

$$\mathsf{N}(Y)_i = \frac{Y_i - \mu_S(Y)}{\sigma_S(Y)}, \qquad i \in S.$$

RMSNorm omits the mean and divides by the root mean square instead.

Scale invariance If each normalization group is rescaled by its own positive scalar, $\tilde{Y}_i = a_S Y_i$ for all $i \in S$, then

$$\mathsf{N}(\tilde{Y}) = \mathsf{N}(Y).$$

The following table applies this rule using the normalization sets of Wu & He (2018, Figure 2). It covers BatchNorm (Ioffe & Szegedy, 2015), LayerNorm (Ba et al., 2016), InstanceNorm (Ulyanov et al., 2016), GroupNorm (Wu & He, 2018), and RMSNorm (Zhang & Sennrich, 2019).

Normalization	Statistics over S	Scale invariance
BatchNorm	$S_c = \{(b,c,h,w) : b,h,w\}$	$\tilde{Y}_{b,c,h,w} = a_c Y_{b,c,h,w}$ [1]
LayerNorm	$S_b = \{(b,c,h,w) : c,h,w\}$	$\tilde{Y}_{b,c,h,w} = a_b Y_{b,c,h,w}$
RMSNorm	$S_b = \{(b,c,h,w) : c,h,w\}$ (no gains)	$\tilde{Y}_{b,c,h,w} = a_b Y_{b,c,h,w}$
InstanceNorm	$S_{b,c} = \{(b,c,h,w) : h,w\}$	$\tilde{Y}_{b,c,h,w} = a_{b,c} Y_{b,c,h,w}$
GroupNorm	$S_{b,g} = \{(b,c,h,w) : c \in C_g, h,w\}$	$\tilde{Y}_{b,c,h,w} = a_{b,g} Y_{b,c,h,w}$ for $c \in C_g$

[1] In evaluation mode, the running mean and variance must be rescaled consistently.

The table shows that the weights $A^{C\times d}$ of a linear layer $x \mapsto Ax$ followed by a BatchNorm layer can use a distinct weight normalization for each channel dimension, corresponding to row-wise weight normalization, whereas, for example, LayerNorm and RMSNorm are scale invariant per batch example. Trivially, global weight normalization is also feasible as a special case in both cases.

A.2 SwiGLU blocks and scale invariance

SwiGLU feedforward blocks (Shazeer, 2020) are commonly written as

$$F(z) = W_{\text{down}}\big(\mathsf{SiLU}(W_{\text{gate}}z) \odot W_{\text{up}}z\big).$$

If the block output is subsequently normalized as $\mathsf{RMSNorm}(F(z))$, then positive rescaling of W_{down} or W_{up} is removed by the normalization:

$$F_{aW_{\text{down}}}(z) = aF(z), \qquad F_{aW_{\text{up}}}(z) = aF(z), \qquad a > 0.$$

Thus the normalized output is invariant to positive rescaling of these two weight blocks. The same exact invariance does not apply to W_{gate}, since $\mathsf{SiLU}(at) \neq a\mathsf{SiLU}(t)$ in general.

A.3 Scale-invariance identities

Proof of Lemma 3.2. Fix a group $i \in \mathcal{I}$. Then differentiating $f(x_1, \dots, ax_i, \dots, x_G) = f(x_1, \dots, x_i, \dots, x_G)$ with respect to x_i gives the gradient scaling identity. For fixed nonzero x, the map $a \mapsto f(x_1, \dots, ax_i, \dots, x_G)$ is constant, so differentiating at $a = 1$ gives $\langle \nabla_i f(x), x_i \rangle = 0$. The final two identities follow by applying Assumption 3.1 to y and summing $\langle \nabla_i f(x), (a_i - 1)x_i \rangle = 0$ over $i \in \mathcal{I}$, since $y_i - x_i = 0$ for $i \notin \mathcal{I}$. □

B Unembedding layer

The final linear layer differs from hidden layers followed by normalization, since its scale directly controls the softmax margins. The following result applies whenever its matrix stays in a bounded set, regardless of whether that constraint is enforced by a Frank-Wolfe type update (Pethick et al., 2025) or by hyperball normalization (Wen et al., 2026).

Theorem B.1 (Constrained weights implies constrained confidence)**.** *Let $\phi \in \mathbb{R}^d$ be a feature vector and $W \in \mathbb{R}^{m\times d}$, $m \geq 2$, be an unembedding matrix with rows w_i. The logits are $z = W\phi$, and the class probabilities are $\pi = \mathrm{softmax}(z)$. Suppose $\|\phi\|_2 \leq H$ and $\|w_i\|_2 \leq B$ for every i, where $H, B < \infty$. Then, for every class y, the softmax probability and cross-entropy loss obey*

$$\pi_y \leq \frac{e^{2BH}}{e^{2BH} + m - 1}, \qquad -\log \pi_y \geq \log\big(1 + (m-1)e^{-2BH}\big) > 0.$$

In particular, achieving $\pi_y \geq 1 - \delta$ for $0 < \delta < 1 - 1/m$ requires

$$BH \geq \tfrac{1}{2}\log\Big(\tfrac{(m-1)(1-\delta)}{\delta}\Big).$$

Proof. Cauchy–Schwarz gives $|z_i| \leq BH$, and hence $z_i - z_y \geq -2BH$ for every $i \neq y$. Therefore

$$\pi_y^{-1} = 1 + \sum_{i\neq y} e^{z_i - z_y} \geq 1 + (m-1)e^{-2BH}.$$

Taking reciprocals and logarithms proves the first two bounds. Solving the probability bound for BH gives the stated necessary condition. □

For an RMS-normalized feature vector with unit gain, we have $\|\phi\|_2 \leq \sqrt{d}$. In this case a row-wise constraint,

$$\|W\|_{\text{RMS}\to\infty} := \sup_{\|x\|_2 \leq \sqrt{d}} \|Wx\|_\infty = \sqrt{d}\max_i \|w_i\|_2 \leq \rho,$$

gives the theorem's bound with $H = \sqrt{d}$ and $B = \rho/\sqrt{d}$, namely $\pi_y \leq e^{2\rho}/(e^{2\rho} + m - 1)$. Thus, the maximum probability is controlled by the chosen weight radius ρ. For large vocabularies this upper bound is small unless ρ grows on the order of $\log m$. In other words, the final layer needs sufficiently large row norms to express highly confident predictions.

C WEIGHT-NORMALIZED STEEPEST DESCENT

Lemma C.1. *Suppose Assumptions 3.1 and 4.1 hold. Then the iterates generated by (1) satisfy*

$$f(\bar{x}^{k+1}) \le f(\bar{x}^k) - \gamma_k \left\|\nabla f(\bar{x}^k)\right\|_* + \tfrac{L_\nu}{1+\nu}\gamma_k^{1+\nu}.$$

Proof of Lemma C.1. Let

$$s^k := \mathrm{lmo}(\nabla f(\bar{x}^k)), \qquad x^{k+1} := \bar{x}^k + \gamma_k s^k.$$

By the definition of the linear minimization oracle,

$$\langle \nabla f(\bar{x}^k), s^k\rangle = -\left\|\nabla f(\bar{x}^k)\right\|_*, \qquad \left\|s^k\right\| \le 1.$$

Apply Hölder smoothness along the trajectory from Assumption 4.1. Writing $g^k := \nabla f(\bar{x}^k)$ gives

$$\begin{aligned} f(x^{k+1}) &\le f(\bar{x}^k) + \gamma_k \langle g^k, s^k\rangle + \tfrac{L_\nu \gamma_k^{1+\nu}}{1+\nu}\left\|s^k\right\|^{1+\nu} \\ &\le f(\bar{x}^k) + \gamma_k \langle g^k, s^k\rangle + \tfrac{L_\nu \gamma_k^{1+\nu}}{1+\nu} \\ &= f(\bar{x}^k) - \gamma_k \left\|g^k\right\|_* + \tfrac{L_\nu \gamma_k^{1+\nu}}{1+\nu}, \end{aligned}$$

Scale invariance gives $f(\bar{x}^{k+1}) = f(x^{k+1})$, proving the claim. □

Proof of Theorem 4.2. Let $\Delta_k := f(\bar{x}^k) - f^\star$ and $g^k := \nabla f(\bar{x}^k)$. By Assumption 4.2, $\Delta_k \le \frac{\|g^k\|_*}{\mu}$. Combining this with Lemma C.1 yields, with $A_\nu := \frac{L_\nu}{1+\nu}$,

$$\Delta_{k+1} \le (1-\mu\gamma_k)\Delta_k + A_\nu \gamma_k^{1+\nu}.$$

With $\gamma_k = \frac{2}{\mu(k+2)}$,

$$\Delta_{k+1} \le \tfrac{k}{k+2}\Delta_k + \tfrac{2^{1+\nu}A_\nu}{\mu^{1+\nu}(k+2)^{1+\nu}}.$$

Let $B_\nu := \frac{2^{1+\nu}A_\nu}{\mu^{1+\nu}}$, so $C_\nu = \frac{B_\nu}{2-\nu}$. The case $k=0$ in the preceding recursion gives

$$\Delta_1 \le \tfrac{A_\nu}{\mu^{1+\nu}} \le \tfrac{C_\nu}{2^\nu}.$$

For the induction step, assume $\Delta_k \le \frac{C_\nu}{(k+1)^\nu}$ and write $m = k+1$. Since $(1+\frac{1}{m})^\nu \le 1 + \frac{\nu}{m}$,

$$\tfrac{m-1}{m+1}\tfrac{(m+1)^\nu}{m^\nu} \le 1 - \tfrac{2-\nu}{m+1}.$$

Therefore

$$\begin{aligned} \Delta_{k+1} &\le \tfrac{C_\nu(m-1)}{(m+1)m^\nu} + \tfrac{B_\nu}{(m+1)^{1+\nu}} \\ &\le \left(C_\nu\left(1 - \tfrac{2-\nu}{m+1}\right) + \tfrac{B_\nu}{m+1}\right)\tfrac{1}{(m+1)^\nu} \\ &\le \tfrac{C_\nu}{(m+1)^\nu} = \tfrac{C_\nu}{(k+2)^\nu}. \end{aligned}$$

This proves the claim. □

D ACCELERATED NORMALIZED SODA

D.1 METHOD

Spectral-ball variant. If instead of an unconstrained update we have

$$h(X) = \tfrac{1}{p}\left\|X\right\|_{S_p}^p + \iota_{\{\|X\|_{S_\infty}\le R\}}(X), \qquad p > 1, \quad R > 0,$$

then, with $q = p/(p-1)$ and $\bar{M}^{k+1} = U\,\mathrm{diag}(s)V^\top$, $s_i \ge 0$, the mirror map is

$$\begin{aligned} Z^{k+1} &= \operatorname*{arg\,min}_{\|X\|_{S_\infty}\le R}\left\{\gamma_k\langle \bar{M}^{k+1}, X\rangle + \tfrac{1}{p}\left\|X\right\|_{S_p}^p\right\} \\ &= -U\,\mathrm{diag}\Big(\min\left\{\gamma_k^{q-1}s_i^{q-1}, R\right\}\Big)V^\top. \end{aligned}$$

Proof. By von Neumann's trace inequality, for fixed singular values $t_i = \sigma_i(X)$ the linear term is minimized by aligning the singular vectors of X against those of $\bar{M}^{k+1}$, so that $X = -U\,\mathrm{diag}(t)V^\top$ and

$$\gamma_k \langle \bar{M}^{k+1}, X\rangle + \tfrac{1}{p}\|X\|_{S_p}^p = \sum_i \left(-\gamma_k s_i t_i + \tfrac{1}{p} t_i^p\right).$$

The constraint $\|X\|_{S_\infty} \le R$ is exactly $0 \le t_i \le R$. Thus the minimization separates across singular values, and each scalar problem has solution $t_i = \min\{(\gamma_k s_i)^{1/(p-1)}, R\}$. □

D.2 Online-to-batch

Proof of Lemma 5.3. Fix $k \ge 1$ and abbreviate $\bar{g}^k := \nabla f(\bar{x}^k)$. Using (2) twice gives

$$\begin{aligned}
&A_k\big(f(\bar{x}^k) - f(x)\big) - A_{k-1}\big(f(\bar{x}^{k-1}) - f(x)\big)\\
&\quad = a_k\big(f(\bar{x}^k) - f(x)\big) + A_{k-1}\big(f(\bar{x}^k) - f(\bar{x}^{k-1})\big)\\
&\quad = a_k\langle \bar{g}^k, \bar{x}^k - x\rangle + A_{k-1}\langle \bar{g}^k, \bar{x}^k - \bar{x}^{k-1}\rangle - a_k D_f(x, \bar{x}^k) - A_{k-1} D_f(\bar{x}^{k-1}, \bar{x}^k)\\
&\quad = \langle \bar{g}^k, A_k\bar{x}^k - A_{k-1}\bar{x}^{k-1} - a_k x\rangle - a_k D_f(x, \bar{x}^k) - A_{k-1} D_f(\bar{x}^{k-1}, \bar{x}^k).
\end{aligned}$$

Since $\bar{x}^k$ is obtained from x^k by positive rescaling of the groups in $\mathcal{I}$ (and coincides with it on the others), scale invariance and radial orthogonality (Assumption 3.1 and Lemma 3.2) give

$$f(\bar{x}^k) = f(x^k), \qquad \langle \bar{g}^k, \bar{x}^k - x^k\rangle = 0.$$

Therefore the final inner product above is unchanged if $A_k\bar{x}^k$ is replaced by $A_k x^k$. Moreover,

$$A_k x^k = A_{k-1}\bar{x}^{k-1} + a_k z^k,$$

because $A_k(1-\lambda_{k-1}) = A_{k-1}$ and $A_k\lambda_{k-1} = a_k$. Hence

$$\begin{aligned}
&A_k\big(f(\bar{x}^k) - f(x)\big) - A_{k-1}\big(f(\bar{x}^{k-1}) - f(x)\big)\\
&\quad = a_k\langle \nabla f(\bar{x}^k), z^k - x\rangle - A_{k-1} D_f(\bar{x}^{k-1}, \bar{x}^k) - a_k D_f(x, \bar{x}^k).
\end{aligned} \tag{5}$$

For $k = 0$, the same identity follows from $A_0 = a_0$ and $\bar{x}^0 = \mathsf{N}_{\omega,\rho}(z^0)$:

$$\begin{aligned}
A_0\big(f(\bar{x}^0) - f(x)\big) &= a_0\langle \nabla f(\bar{x}^0), \bar{x}^0 - x\rangle - a_0 D_f(x, \bar{x}^0)\\
&= a_0\langle \nabla f(\bar{x}^0), z^0 - x\rangle - a_0 D_f(x, \bar{x}^0),
\end{aligned}$$

where the second equality again uses Lemma 3.2.

By construction, z^k and $\bar{x}^k$ are $\mathcal{F}_{k-1}$-measurable. Therefore, using the conditional unbiasedness in Assumption 5.3,

$$\mathbb{E}\left[\langle g^k, z^k - x\rangle \mid \mathcal{F}_{k-1}\right] = \langle \nabla f(\bar{x}^k), z^k - x\rangle.$$

Summing and taking expectations yields

$$\begin{aligned}
A_{n-1}\mathbb{E}[f(\bar{x}^{n-1}) - f(x)] \le\ &\mathbb{E}\left[\textstyle\sum_{k=0}^{n-1} a_k\langle g^k, z^k - x\rangle\right]\\
&- \textstyle\sum_{k=1}^{n-1} A_{k-1}\mathbb{E} D_f(\bar{x}^{k-1}, \bar{x}^k) - \sum_{k=0}^{n-1} a_k \mathbb{E} D_f(x, \bar{x}^k).
\end{aligned} \tag{6}$$

Assumption 5.1 gives $D_f(x, \bar{x}^k) \ge 0$. Applying Assumption 5.2 to $D_f(\bar{x}^{k-1}, \bar{x}^k)$ in (6), and dropping the remaining nonpositive terms, proves (4). □

D.3 Acceleration

We use the following form of ODA with gradient feedback g^k (where the gradients are computed is only specified by the online-to-batch conversion):

$$\begin{aligned}
m^{k+1} &= (1-\alpha_k)m^k + \alpha_k g^k,\\
\bar{m}^{k+1} &= (1-\bar{\alpha}_k)m^{k+1} + \bar{\alpha}_k g^k,\\
z^{k+1} &\in \partial h^*(-\gamma_k \bar{m}^{k+1}) = \operatorname*{arg\,min}_{z\in\mathcal{X}} \left\{\gamma_k\langle \bar{m}^{k+1}, z\rangle + h(z)\right\}.
\end{aligned} \tag{ODA}$$

Here $\alpha_k, \bar{\alpha}_k \in [0,1]$ and $\gamma_k > 0$.

The following regret bound is Lemma A.1 of Pethick (2026).

Lemma D.1 (ODA regret (Pethick, 2026, Lem. A.1))**.** *Consider ODA. Choose positive weights* $a_0, \dots, a_{n-1}$ *and write*

$$A_k := \textstyle\sum_{i=0}^{k} a_i, \qquad A_{-1} := 0.$$

We use the convention $g^{-1} = 0$ *and set* $z^0 \in \partial h^*(0)$*. For* $k = 0, \dots, n-2$*, choose*

$$\alpha_k = \tfrac{a_k}{A_k}, \quad \bar{\alpha}_k = \tfrac{a_{k+1}}{A_{k+1}}, \quad \gamma_k = \eta \tfrac{A_k}{1-\bar{\alpha}_k}$$

for some $\eta > 0$*. Let* $p \geq 2$*, let* $q = p/(p-1)$*, and let* h *be* p*-uniformly convex with constant* μ_h *in the sense of Definition 5.1. Then, for every* $x \in \mathcal{X}$*,*

$$\textstyle\sum_{k=0}^{n-1} a_k \langle g^k, z^k - x\rangle \leq \frac{h(x) - \inf_{u\in\mathcal{X}} h(u)}{\eta} + \frac{\eta^{q-1}}{q\mu_h^{q-1}} \sum_{k=0}^{n-1} a_k^q \left\|g^k - g^{k-1}\right\|_*^q. \tag{7}$$

We are now ready to prove the convergence result for NorSODA. The analysis follows Pethick (2026) closely, but uses the tighter online-to-batch conversion result in Lemma 5.3 for the normalized update, which corresponds to setting $c_\nu = 1/r_\nu$ and $\bar{\lambda}_k = 0$ in the proof of Pethick (2026, A.3).

Proof of Theorem 5.4. We first prove the result for a general reference function h that is p-uniformly convex with constant μ_h (Definition 5.1), with $D := h(x^\star) - \inf h$, and specialize to $h(x) = \frac{1}{p}\left\|x - z^0\right\|_p^p$ at the end. Choose $a_k = k+1$, so that $A_k = (k+1)(k+2)/2$. Then

$$\alpha_k = \tfrac{a_k}{A_k} = \tfrac{2}{k+2}, \qquad \bar{\alpha}_k = \tfrac{a_{k+1}}{A_{k+1}} = \tfrac{2}{k+3}, \qquad \gamma_k = \eta \tfrac{A_k}{1-\bar{\alpha}_k} = \eta \tfrac{(k+2)(k+3)}{2}.$$

Moreover, $\lambda_k = a_{k+1}/A_{k+1} = 2/(k+3)$, so NorSODA is the normalized online-to-batch wrapper in (3) applied to ODA with $g^k = \nabla f(\bar{x}^k, \xi_k)$. Combining Lemma D.1 with Lemma 5.3 gives

$$\begin{aligned} A_{n-1}\mathbb{E}[f(\bar{x}^{n-1}) - f(x^\star)] \leq \tfrac{D}{\eta} + \tfrac{\eta^{q-1}}{q\mu_h^{q-1}} \textstyle\sum_{k=0}^{n-1} (k+1)^q \mathbb{E}\left\|g^k - g^{k-1}\right\|_*^q \\ - \tfrac{1}{r_\nu L_\nu^{1/\nu}} \textstyle\sum_{k=1}^{n-1} A_{k-1} \mathbb{E}\left[\left\|\nabla f(\bar{x}^k) - \nabla f(\bar{x}^{k-1})\right\|_*^{r_\nu}\right]. \end{aligned}$$

We next split the regret noise term using Assumption 5.4. For $k \geq 1$,

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

For $k = 0$, using the boundary convention $g^{-1} = 0$ gives

$$\mathbb{E}\left\|g^0 - g^{-1}\right\|_*^q \leq \tau_q^q \mathbb{E}\left\|\nabla f(\bar{x}^0) - \nabla f(x^\star)\right\|_*^q + \sigma_q^q.$$

Since $\sum_{k=0}^{n-1}(k+1)^q \lesssim_q n^{q+1}$,

$$\begin{aligned} A_{n-1}\mathbb{E}[f(\bar{x}^{n-1}) - f(x^\star)] \lesssim_q \tfrac{D}{\eta} + \tfrac{\eta^{q-1}\sigma_q^q n^{q+1}}{\mu_h^{q-1}} \\ + \tfrac{\tau_q^q \eta^{q-1}}{\mu_h^{q-1}} \textstyle\sum_{k=1}^{n-1} (k+1)^q \mathbb{E}\left[\left\|\nabla f(\bar{x}^k) - \nabla f(\bar{x}^{k-1})\right\|_*^q\right] \\ - \tfrac{1}{r_\nu L_\nu^{1/\nu}} \textstyle\sum_{k=1}^{n-1} A_{k-1} \mathbb{E}\left[\left\|\nabla f(\bar{x}^k) - \nabla f(\bar{x}^{k-1})\right\|_*^{r_\nu}\right] \\ + \tfrac{\tau_q^q \eta^{q-1}}{\mu_h^{q-1}} \mathbb{E}\left[\left\|\nabla f(\bar{x}^0) - \nabla f(x^\star)\right\|_*^q\right]. \end{aligned} \tag{8}$$

The last term of (8) is dominated by the first term by the same argument as in the non-accelerated proof, since the deterministic branch of η is at most

$$\tfrac{\mu_h^{(1+\nu)/p} D^{(p-1-\nu)/p}}{\tau_q L_\nu}.$$

It remains to control the positive and negative variation sums. For $k \geq 1$, $(k+1)^q \lesssim_q k^q$ and $A_{k-1} \gtrsim k^2$. Since $p > 1 + \nu$, we have $r_\nu = (1+\nu)/\nu > p/(p-1) = q$, as $(1+\nu)(p-1) > \nu p$. Thus, for all $A, B > 0$,

$$\sup_{s\geq 0}\{As^q - Bs^{r_\nu}\} \leq C_{p,\nu} A^{r_\nu/(r_\nu - q)} B^{-q/(r_\nu - q)}.$$

Applying this bound with

$$A = \tfrac{\tau_q^q \eta^{q-1}}{\mu_h^{q-1}} k^q, \qquad B = \tfrac{k^2}{r_\nu L_\nu^{1/\nu}},$$

and absorbing the factor r_ν into $C_{p,\nu}$ yields

$$\sup_{s\geq 0}\left\{\frac{\tau_q^q\eta^{q-1}}{\mu_h^{q-1}}k^q s^q - \frac{k^2}{r_\nu L_\nu^{1/\nu}}s^{r_\nu}\right\} \leq C_{p,\nu}\frac{\tau_q^{p(1+\nu)/(p-1-\nu)}L_\nu^{p/(p-1-\nu)}}{\mu_h^{(1+\nu)/(p-1-\nu)}}\eta^{(1+\nu)/(p-1-\nu)}k^{p(1-\nu)/(p-1-\nu)}. \tag{9}$$

The exponent of k in (9) comes from

$$q\frac{r_\nu}{r_\nu-q} - 2\frac{q}{r_\nu-q} = \frac{p(1-\nu)}{p-1-\nu}.$$

Define

$$\theta_\nu := \frac{1+\nu}{p-1-\nu}, \qquad \xi_\nu := \frac{p(1-\nu)}{p-1-\nu}, \qquad M_\nu := \frac{L_\nu^{p/(p-1-\nu)}}{\mu_h^{(1+\nu)/(p-1-\nu)}}.$$

With

$$s_k := \left\|\nabla f(\bar{x}^k) - \nabla f(\bar{x}^{k-1})\right\|_*,$$

(9) gives, for each $k \geq 1$,

$$\frac{\tau_q^q\eta^{q-1}}{q\mu_h^{q-1}}(k+1)^q\mathbb{E}[s_k^q] - \frac{1}{r_\nu L_\nu^{1/\nu}}A_{k-1}\mathbb{E}[s_k^{r_\nu}] \leq O_{p,\nu}\left(\tau_q^{p(1+\nu)/(p-1-\nu)}M_\nu\eta^{\theta_\nu}k^{\xi_\nu}\right).$$

Summing these remainders and using

$$\textstyle\sum_{k=1}^{n-1}k^{\xi_\nu} \lesssim_{p,\nu} n^{1+\xi_\nu}, \qquad A_{n-1} = \Theta(n^2),$$

we obtain

$$\mathbb{E}[f(\bar{x}^{n-1}) - f(x^\star)] \leq O_{p,\nu}\left(\frac{D}{\eta n^2} + \tau_q^{p(1+\nu)/(p-1-\nu)}M_\nu\eta^{\theta_\nu}n^{\xi_\nu-1} + \frac{\eta^{q-1}\sigma_q^q n^{q-1}}{\mu_h^{q-1}}\right). \tag{10}$$

Equating the deterministic and Hölder-variation terms in (10) gives

$$\frac{D}{\eta n^2} \asymp_{p,\nu} \tau_q^{p(1+\nu)/(p-1-\nu)}M_\nu\eta^{\theta_\nu}n^{\xi_\nu-1},$$

or equivalently,

$$\eta^{1+\theta_\nu} \asymp_{p,\nu} \frac{D}{\tau_q^{p(1+\nu)/(p-1-\nu)}M_\nu n^{1+\xi_\nu}}.$$

Since

$$1+\theta_\nu = \frac{p}{p-1-\nu}, \qquad (1+\xi_\nu)\frac{p-1-\nu}{p} = 2-\kappa_{\mathrm{acc}},$$

this yields

$$\eta_{\mathrm{det}} \asymp_{p,\nu} \frac{\mu_h^{(1+\nu)/p}D^{(p-1-\nu)/p}}{\tau_q^{1+\nu}L_\nu n^{2-\kappa_{\mathrm{acc}}}}, \qquad \kappa_{\mathrm{acc}} = \frac{1+(p+1)\nu}{p}.$$

Substituting this value of η_{det} into $D/(\eta n^2)$ gives the deterministic rate in the theorem.

Equating the deterministic and stochastic terms in (10) gives

$$\frac{D}{\eta n^2} \asymp_p \frac{\eta^{q-1}\sigma_q^q n^{q-1}}{\mu_h^{q-1}}, \qquad \text{hence} \qquad \eta^q \asymp_p \frac{\mu_h^{q-1}D}{\sigma_q^q n^{q+1}}.$$

Using $q = p/(p-1)$, this gives

$$\eta_{\mathrm{stoch}} \asymp_p \frac{\mu_h^{1/p}D^{(p-1)/p}}{\sigma_q n^{(2p-1)/p}},$$

and substitution into $D/(\eta n^2)$ gives the stochastic rate $\sigma_q D^{1/p}/(\mu_h^{1/p}n^{1/p})$. Hence, with $\eta \asymp_{p,\nu} \min\{\eta_{\mathrm{det}}, \eta_{\mathrm{stoch}}\}$,

$$\mathbb{E}[f(\bar{x}^{n-1}) - f(x^\star)] = O_{p,\nu}\left(\frac{\tau_q^{1+\nu}L_\nu D^{(1+\nu)/p}}{\mu_h^{(1+\nu)/p}n^{\kappa_{\mathrm{acc}}}} + \frac{\sigma_q D^{1/p}}{\mu_h^{1/p}n^{1/p}}\right).$$

Finally, for $h(x) = \frac{1}{p}\left\|x - z^0\right\|_p^p$ we have $\inf h = 0$, so $D = R_p^p/p$, and h is p-uniformly convex with respect to $\|\cdot\|_p$ with $\mu_h = 2^{2-p}$. Substituting these values into η_{det}, η_{stoch}, and the rate above, and absorbing the factors that depend only on p into the constants, gives the step size and the rate stated in Theorem 5.4. □

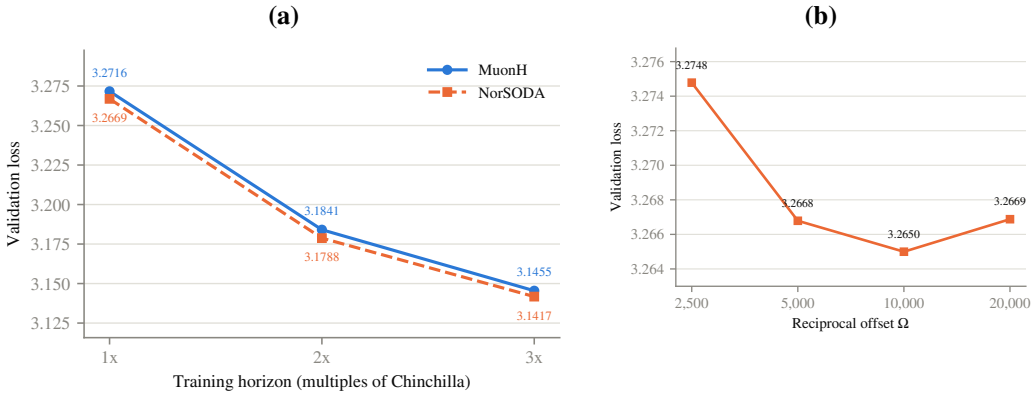


Figure 4: **(a)** MuonH and NorSODA across training horizons when both methods use the same Marin horizon-scaled hidden and head learning rates at $2\times$ and $3\times$. **(b)** NorSODA validation loss as a function of the reciprocal offset Ω at the standard $1\times$ horizon.

E Experimental hyperparameters

Table 1 shows the hyperparameters used for the $1\times$-Chinchilla experiments. For the horizon comparison, the only modification is that K is changed to 10,200 or 15,300.

Table 1: Hyperparameters used for the Muon, MuonH, and NorSODA experiments. Here $[a]_+ := \max\{a, 0\}$, K is the training horizon, and O_k is the Newton–Schulz direction.

Hyperparameter	Muon	MuonH	NorSODA
Transformer layers		12	
Width / att. heads / head dim.		768 / 6 / 128	
Activation		$2\ \mathrm{ReLU}(\cdot)^2$	
Normalization		sandwich RMSNorm, QK RMSNorm	
Vocabulary		50,304	
Dataset		FineWeb10B	
Sequence length / global batch		1,024 / 512	
Horizon K		5,100 ($1\times$ Chinchilla)	
Momentum		0.95	
Newton–Schulz steps		12	
Output peak learning rate		$2^{-12}(3000/768) = 0.0009536743$	
Hidden peak learning rate γ_0	0.025	0.018	0.018
Weight decay	0.0125	0	0
Hidden-weight normalization	none	initial Frobenius sphere	initial Frobenius sphere
Hidden direction scale	Muon aspect correction	$R_0/\|O_k\|_F$	$R_0/\sqrt{\min(m,n)}$
Learning-rate schedule	$\gamma_k = \gamma_0[1-k/K]_+$	$\gamma_k = \gamma_0[1-k/K]_+$	$\gamma_k = \gamma_0\frac{\Omega}{k+\Omega}[1-k/K]_+$
Primal averaging λ_k	—	—	$\frac{2}{k+\Omega}[1-k/K]_+$
Reciprocal offset Ω	—	—	20,000
Output anchor	—	—	initialization

F Additional optimizer ablations

Figure 4(a) repeats the training-horizon comparison after scaling both the hidden and head learning rates as $(K/5100)^{-0.395}$ for $K > 5100$, following the empirical horizon-scaling rule for MuonH from Marin (Wen et al., 2026). The $1\times$ points coincide with the main-text comparison because their learning-rate multiplier is one. Figure 4(b) varies NorSODA's reciprocal offset Ω at the $1\times$ horizon while fixing all other hyperparameters.