TUON: A TENSOR-AWARE OPTIMIZER UTILIZING TENSOR-SVD AND POLARIZATION FOR LARGE LANGUAGE MODEL PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large scale language model (LLM) pretraining is increasingly bottlenecked by optimizer efficiency, driving optimizer design from vector-aware methods such as AdamW to matrix-aware methods. Matrix-aware optimizers such as Muon exploit matrix structure but treat layers independently, neglecting cross-layer correlations. Tensor-aware optimizers are therefore needed, yet existing ones rely on a matricization-based tensor algebra and remain restricted to shallow stacking and specific modules. In this paper, we propose **TUON**, a novel optimizer built on a transform-based tensor algebra derived from the Tensor-SVD (T-SVD). Unlike methods that polarize a globally unfolded matrix, TUON mixes layers via a unitary transform $\boldsymbol{Q}$ and then polarizes each transformed slice, and inverts the transform to recover the update. This mix-then-separate strategy captures cross-layer correlations while preserving per-layer diversity, avoiding the structural dimension imbalance and cross-layer misalignment of unfolding. Thus it successfully scales to more layers and diverse modules beyond QKV. Theoretically, we prove that TUON performs steepest descent under a $\boldsymbol{Q}$-transformed tensor spectral norm, and identify its update as the tensor polar factor of the T-SVD. We further establish transform non-redundancy, norm and smoothness comparisons with Muon and TEON, and a convergence guarantee. Empirically, TUON improves consistently over Muon and TEON on GPT-124M/350M and Qwen-0.6B/4B, with ablations confirming the transformation mechanism, cross-layer structure beyond two-layer stacking, adaptability beyond QKV, and robustness to the polarization method.

1 INTRODUCTION

As model and data scales continue to grow, the resource intensity of large scale language model (LLM) pretraining has placed optimization efficiency at the forefront of research Liu et al. (2025c;a); Zhang et al. (2026b). In this context, optimizers are critical for training efficiency. The past years have witnessed optimizer advances along different structural granularity levels, from *vector-aware* to *matrix-aware* methods. Among them, Adam (Kingma & Ba, 2015) and AdamW (Loshchilov & Hutter, 2017) remain the default and most widely-used optimizers. It treats parameters via coordinate-wise momentum statistics, placing it within the *vector-aware* family (Xie et al., 2024; Zhang et al., 2024). Then, *matrix-aware* optimizers (Gupta et al., 2018; Vyas et al., 2025) are emerging as a promising paradigm for next-generation efficient training algorithms. As a representative, Muon (Jordan et al., 2024) has achieved record-breaking pretraining performance on LLMs (Liu et al., 2025a; Team et al., 2025). Compared with vector-aware optimizers, the improvement may stem from matrix-aware optimizers preserving more information from the matrix structure in Transformers.

A Transformer naturally has richer structure than the matrix view suggests. Each module repeats across depth with compatible shapes, forming a tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{m\times n\times K}$ indexing layers. Yet to fit into optimizer frameworks, this structure is usually split into separate matrices $\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(K)} \in \mathbb{R}^{m\times n}$, losing cross-layer information. Existing matrix-aware methods operate layer-wise and thus neglect cross-layer correlations. This has motivated *tensor-aware* optimizers. TEON (Zhang et al., 2026a) matricizes the third-order gradient tensors along the first and second modes and orthogonalizes them, extending Muon to the tensor level. Tensorion (Bogachev et al., 2026) instead relaxes the

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.

tensor spectral norm into a tractable surrogate, operating on native higher-order parameters within a single layer and adaptively selecting the unfolding.

Despite their empirical success, existing tensor-aware optimizers exhibit two practical limitations. First, they are restricted in scope. TEON is applied for QKV-specific structures and performs worse on other modules such as output projections and MLPs, so its benefits remain confined to a narrow subset of the network. Tensorion has been applied to vision tasks, leaving its applicability to LLMs unclear. Second, they are confined to shallow stacking. TEON supports only two-layer stacking, and its performance degrades as more layers are concatenated, preventing it from realizing the theoretical information gain that larger stacking sizes are expected to yield (Zhang et al., 2026a).

These limitations stem from two structural problems. The first is a dimension imbalance induced by unfolding. As the number of layers grows, the unfolded matrix becomes increasingly rectangular, so its number of singular directions is bounded by its smaller dimension, and may be far smaller than the larger dimension. Thus, these directions span only a small subspace of the full parameter space, causing information loss. The second is cross-layer misalignment. Even within the same module type, each slice has its own semantic role and evolves differently during training, which means each dominant subspace may not align. Global unfolding concatenates all slices into one matrix to solve for a single shared subspace that approximates all slices collectively. This compromise subspace smooths out slice-specific characteristics and suppresses head- or layer-specific signals. Ultimately, these problems stem from the tensor algebra underlying existing tensor-aware optimizers. Addressing them requires a different tensor algebra, which induces a geometry better suited to cross-layer optimization. This raises a natural question:

Can we design a tensor-aware optimizer that captures cross-layer structure beyond two-layer stacking, adapts to diverse module types beyond QKV, and scales across model sizes, while being theoretically interpretable under a proper tensor geometry?

In this work, we answer this question affirmatively by introducing **TUON**, a **T**ensor-aware optimizer **U**tilizing tens**O**r-SVD and polarizatio**N**, built upon a *transform-based tensor algebra* grounded in the Tensor-SVD (T-SVD) (Kilmer & Martin, 2011; Kernfeld et al., 2015; Zheng et al., 2026). For third-order tensors in Transformer, TUON applies a unitary transform $\boldsymbol{Q}$ along the layer mode to exploit cross-layer structure, then polarizes each transformed slice independently, and inverts the transform to recover the update. While the algebra only requires $\boldsymbol{Q}$ to be invertible, we further require it to decorrelate layer-wise coupling to prevent mutual interference between slices during polarization. Thus we propose **DecorQ**, a data-dependent transform that diagonalizes the layer Gram matrix and decouples the update into independent orthogonal subproblems. This *mix-then-separate* design captures cross-layer structure while preserving per-layer diversity, avoiding the information loss of global unfolding and scaling gracefully to many layers and diverse modules beyond QKV.

Theoretically, we first give the tensor spectral under $\boldsymbol{Q}$-transformation and its dual. Then we prove that TUON is steepest descent for this norm. We further show that the update is the tensor polar factor of a Q-product T-SVD and prove that the transform is non-redundant. At last, we establish norm comparability and smoothness comparison with Muon and TEON, and provide a convergence guarantee. Empirically, TUON delivers consistent improvements across GPT-124M/350M and Qwen-0.6B/4B. Ablations further confirm the effectiveness of the transform DecorQ, its extension beyond two-layer stacking, its adaptability beyond QKV, and its robustness to the polarization method.

Our main contributions are as follows:

- We propose TUON, a tensor-aware optimizer based on transform-based tensor algebra that better exploits cross-layer information, with the DecorQ transform to decouple layers without mutual interference during polarization. The optimizer applies across layer depths, module types, and model sizes.
- We prove that TUON performs steepest descent under the transform-induced norm. Moreover, we identify the update as the tensor polar factor of a $\boldsymbol{Q}$-product T-SVD, and prove the non-redundancy of this transform.
- We establish geometry comparisons with Muon and TEON. The transform distorts the Muon geometry by at most a factor of $\sqrt{K}$, and at a common operator radius, the TUON oracle is at least as strong as the TEON oracle, and can be up to $\sqrt{K}$ stronger.

- We provide comprehensive experiments. TUON improves over Muon and TEON on GPT-124M/350M and Qwen-0.6B/4B, with ablations confirming the transformation mechanism, cross-layer structure beyond two-layer stacking, adaptability beyond QKV, and robustness to the polarization method.

§2 presents the TUON method, §3 its theoretical properties, §4 the experimental results. §5 concludes with future directions. **Notation**, **preliminaries**, and a comprehensive **related work** review are provided in Appendices A, B, and C, respectively.

2 Methodology of TUON

2.1 Momentum tensor and layer transform

For one compatible group, stack the momenta at t-th iteration as $\boldsymbol{\mathcal{M}}_t(:,:,k) = \boldsymbol{M}_t^{(k)}$ and $\boldsymbol{\mathcal{M}}_t = \beta\boldsymbol{\mathcal{M}}_{t-1} + (1-\beta)\boldsymbol{\mathcal{G}}_t$ where $\boldsymbol{\mathcal{M}}_t \in \mathbb{R}^{m\times n\times K}$. Given a unitary $\boldsymbol{Q} \in \mathbb{C}^{K\times K}$, $\boldsymbol{Q}^{\mathsf{H}}\boldsymbol{Q} = I$, where $\boldsymbol{Q}^{\mathsf{H}}$ denotes the conjugate transpose of $\boldsymbol{Q}$, our tensor-product-transform treats every tube as a length-K column,

$$\widehat{\boldsymbol{\mathcal{M}}}_t = \boldsymbol{\mathcal{M}}_t \times_3 \boldsymbol{Q}, \quad \text{where the } j\text{-th slice of } \widehat{\boldsymbol{\mathcal{M}}}_t \quad \widehat{\boldsymbol{M}}_{t,j} = \widehat{\boldsymbol{\mathcal{M}}}_t(:,:,j) = \sum_{k=1}^{K} \boldsymbol{Q}_{jk}\boldsymbol{M}_t^{(k)}. \tag{1}$$

Choice of the layer transform $\boldsymbol{Q}$. In this tensor algebra derived from T-SVD, the layer transform $\boldsymbol{Q}$ only needs to be unitary (or invertible). Many standard transforms satisfy this requirement, including the DCT, the Haar wavelet transform, and the FFT (discussed in Appendix E). These fixed transforms are simple and computationally efficient. However, a fixed transform imposes a prior mismatched to actual layer statistics. To see this, let $\mathsf{M}_t \in \mathbb{R}^{K\times mn}$ be the mode-3 unfolding of the momentum tensor, whose k-th row is $\mathrm{vec}(\boldsymbol{M}_t^{(k)})^{\mathsf{T}}$, and define the layer-wise Gram matrix

$$\mathsf{C}_t := \mathsf{M}_t\mathsf{M}_t^{\mathsf{T}} \in \mathbb{R}^{K\times K}, \qquad (\mathsf{C}_t)_{ij} = \langle \boldsymbol{M}_t^{(i)}, \boldsymbol{M}_t^{(j)}\rangle_{\mathrm{F}}. \tag{2}$$

This matrix encodes the pairwise coupling among layers. Large off-diagonal entries indicate strongly correlated update directions, while near-zero entries indicate decoupled evolution. A good choice of $\boldsymbol{Q}$ should decorrelate this structure so that the transformed slices evolve independently.

This motivates our default choice, **DecorQ**. We set $\boldsymbol{Q} = U_3^{\mathsf{T}}$, where U_3 is the left singular matrix of the SVD $\mathsf{M}_t = \boldsymbol{U}_3\boldsymbol{\Sigma}_3\boldsymbol{V}_3^{\mathsf{T}}$. Under this transform, the Gram matrix of the transformed slices becomes

$$\widehat{\mathsf{C}}_t = \boldsymbol{Q}\mathsf{M}_t\mathsf{M}_t^{\mathsf{T}}\boldsymbol{Q}^{\mathsf{T}} = \boldsymbol{U}_3^{\mathsf{T}}\big(\boldsymbol{U}_3\boldsymbol{\Sigma}_3^2\boldsymbol{U}_3^{\mathsf{T}}\big)\boldsymbol{U}_3 = \boldsymbol{\Sigma}_3^2, \tag{3}$$

which is diagonal. Hence the transformed slices satisfy $\langle \widehat{\boldsymbol{M}}_{t,i}, \widehat{\boldsymbol{M}}_{t,j}\rangle_{\mathrm{F}} = 0$ for $i \neq j$, which are mutually orthogonal. The original coupled K-layer of a tensor is therefore decoupled into K independent single-channel problems, *each evolving in its own orthogonal subspace without mutual interference*. DecorQ is data-dependent and is refreshed from the momentum statistics during training, so the transform adapts to the evolving cross-layer structure rather than imposing a fixed prior. We compare DecorQ against DCT, Haar, and DFT in experiments in §4.2, where DecorQ consistently achieves the best performance. Unlike existing approaches that orthogonalize the unfolded matrix as a whole, TUON uses unfolding only to construct $\boldsymbol{Q}$. Full derivations and additional transform choices are provided in Appendix E.

2.2 Polarization and inverse transform

Then for each transformed slice $\widehat{\boldsymbol{M}}_{t,j}$, we polarize (or orthogonalize) them, defined as

$$\widehat{\boldsymbol{P}}_{t,j} = \mathrm{polar}(\widehat{\boldsymbol{M}}_{t,j}) = \boldsymbol{U}_{t,j}\boldsymbol{V}_{t,j}^{\top}, \tag{4}$$

where $\boldsymbol{U}_{t,j}$ and $\boldsymbol{V}_{t,j}$ correspond to the singular value decomposition (SVD) of $\widehat{\boldsymbol{M}}_{t,j} = \boldsymbol{U}_{t,j}\boldsymbol{\Sigma}_{t,j}\boldsymbol{V}_{t,j}^{\top}$. Then the polarized tensor $\widehat{\boldsymbol{\mathcal{P}}}_t$ is made up by each polarized slice $\widehat{\boldsymbol{P}}_{t,j}$, i.e., $\widehat{\boldsymbol{P}}_{t,j} = \widehat{\boldsymbol{\mathcal{P}}}_t(:,:,j), (j = 1,\cdots,K)$. Instead of exact SVD, the polar factor is approximated by fast iterative methods such as Newton–Schulz (NS, Appendix F).

Algorithm 1 TUON

Require: compatible matrices $\{\boldsymbol{W}_t^{(k)}\}_{k=1}^K$, gradients $\{\boldsymbol{G}_t^{(k)}\}_{k=1}^K$ and momentum $\{\boldsymbol{M}_t^{(k)}\}_{k=1}^K$; learning rate η_t, momentum coefficient β, weight decay coefficient λ, NS steps J

$\boldsymbol{\mathcal{G}}_t \leftarrow \mathrm{stack}_3(\boldsymbol{G}_t^{(1)}, \ldots, \boldsymbol{G}_t^{(K)})$
$\boldsymbol{\mathcal{M}}_t \leftarrow \beta \boldsymbol{\mathcal{M}}_{t-1} + (1-\beta)\boldsymbol{\mathcal{G}}_t$
Compute the Gram matrix $\mathsf{C}_t := \mathsf{M}_t \mathsf{M}_t^{\mathsf{T}} \in \mathbb{R}^{K\times K}$ with $(\mathsf{C}_t)_{k\ell} = \langle \boldsymbol{\mathcal{M}}_t(:,:,k), \boldsymbol{\mathcal{M}}_t(:,:,\ell)\rangle_{\mathrm{F}}$
Compute eigendecomposition $\mathsf{C}_t = \mathbf{U}_t \boldsymbol{\Lambda}_t \mathbf{U}_t^{\mathsf{T}}$
Set $\boldsymbol{Q}_t \leftarrow \mathbf{U}_t^{\mathsf{T}}$ ▷ Refresh per 200 iterations
$\widehat{\boldsymbol{\mathcal{M}}}_t \leftarrow \boldsymbol{\mathcal{M}}_t \times_3 \boldsymbol{Q}_t$
for $j = 1, \ldots, K$ **in parallel do**
    $\widehat{\boldsymbol{\mathcal{P}}}_t(:,:,j) \leftarrow \mathrm{polarization}(\widehat{\boldsymbol{\mathcal{M}}}_t(:,:,j))$ ▷ Newton–Schulz, J steps
end for
$\boldsymbol{\mathcal{P}}_t \leftarrow \widehat{\boldsymbol{\mathcal{P}}}_t \times_3 \boldsymbol{Q}_t^{\mathsf{H}}$
$\boldsymbol{\mathcal{W}}_{t+1} \leftarrow (1-\eta_t\lambda)\boldsymbol{\mathcal{W}}_t - \eta_t\gamma_t\boldsymbol{\mathcal{P}}_t$ with $\gamma_t \leftarrow 1/(\mathrm{RMS}(\boldsymbol{\mathcal{P}}_t) + \epsilon)$

After obtaining the approximate polar factors $\widehat{\boldsymbol{\mathcal{P}}}_t$, we invert the transform to recover the update in the original tensor domain as

$$\boldsymbol{\mathcal{P}}_t = \widehat{\boldsymbol{\mathcal{P}}}_t \times_3 \boldsymbol{Q}^{\mathsf{H}}. \tag{5}$$

To control the overall step magnitude while preserving the direction of the update, we then scale the entire stack by $\gamma_t = \frac{1}{\mathrm{RMS}(\boldsymbol{\mathcal{P}}_t)+\epsilon}$, where $\mathrm{RMS}(\boldsymbol{\mathcal{P}}_t) = \sqrt{\frac{1}{mnK}\sum_{i=1}^m\sum_{j=1}^n\sum_{k=1}^K |\boldsymbol{\mathcal{P}}_t(i,j,k)|^2}$ is the root-mean-square value (typically $1/\sqrt{\max(m,n)}$, as suggested by the rank–energy identity in Proposition 3), and ϵ avoids division by zero. Since γ_t is shared across all layers, it only changes the effective trust-region radius, not the relative direction across slices. Finally, with decoupled weight decay, $\boldsymbol{\mathcal{W}}_{t+1} = (1-\eta_t\lambda)\,\boldsymbol{\mathcal{W}}_t - \eta_t\,\widetilde{\boldsymbol{\mathcal{P}}}_t, \quad \widetilde{\boldsymbol{\mathcal{P}}}_t = \gamma_t\,\boldsymbol{\mathcal{P}}_t$ where η_t is the learning rate and λ is the weight-decay coefficient.

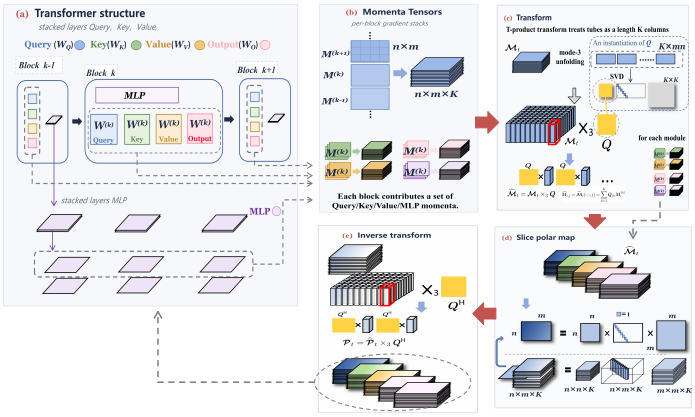


Figure 1: Illustration of the TUON update

Complexity analysis. Table 1 compares the resulting complexities. In the typical regime $K \ll \min(m,n)$, TUON has the comparable asymptotic cost as Muon and TEON. A detailed derivation of the per-iteration complexity is deferred to Appendix G.1.

Table 1: Per-iteration computational complexity for a single tensor group of $K \times m \times n$.

Optimizer	Per-iteration complexity	Dominant term
Muon	$O(K\,mn \cdot \min(m,n))$	$K\,mn \cdot \min(m,n)$
TEON	$O(K\,mn \cdot \min(m,nK))$	$Kmn \cdot \min(m,nK)$
TUON	$O\big(K\,mn \cdot \min(m,n) + K^2mn\big)$	$K\,mn \cdot \min(m,n)$

3 STEEPEST DESCENT THEORY AND COMPARATIVE ADVANTAGES

This section develops the complete theory of TUON. Preliminaries including Matrix SVD, operator/nuclear norm duality, polar factors, norm-induced LMO, T-SVD, and norm-smoothness are reviewed in Appendix B. The Muon and TEON geometries, including their trust-region updates, are stated in Appendix D. Detailed step-by-step proofs of every result in this section are given in Appendix H. The theoretical development answers **four questions**. **i)** Is the update in our method the exact steepest descent direction under some well-defined norm geometry (Sec. 3.1)? **ii)** Does the layer transform cancel algebraically with the inverse transform (Sec. 3.1)? **iii)** How does the geometry compare with Muon and TEON (Sec. 3.2)? **iv)** Can a strict convergence be certified (Sec. 3.3)?

Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{m\times n\times K}$ stack K compatible matrices, with frontal slices $\boldsymbol{X}_k := \boldsymbol{\mathcal{X}}(:,:,k)$. Define mode-1 unfolding as $\mathscr{U}(\boldsymbol{\mathcal{X}}) := [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{nK}] \in \mathbb{C}^{m\times nK}$. Under this convention, $\mathscr{U}(\boldsymbol{\mathcal{X}} \times_3 \boldsymbol{Q}) = \mathscr{U}(\boldsymbol{\mathcal{X}})R_{\boldsymbol{Q}}$, $R_{\boldsymbol{Q}} := \boldsymbol{Q}^{\mathsf{T}} \otimes I_n$, where $\otimes$ is the Kronecker product and I_n is the identity matrix of size n. Since both $\boldsymbol{Q}$ and I_n are unitary, their Kronecker product $R_{\boldsymbol{Q}}$ is unitary. Below, $\|\cdot\|_{\mathrm{Muon}}$ and $\|\cdot\|_{\mathrm{TEON}}$ denote the spectral norms used by Muon and TEON, respectively; their duals and optimizer maps are given in Appendix D. Appendix B.4 verifies directly that $\|\cdot\|_{\boldsymbol{Q},\mathrm{Ts}}$ is a norm. Proposition 1 identifies the dual norm associated with TUON, which measures the largest first-order alignment with a direction in the unit trust region. The proof can be found in Appendix H.

Definition 1 (Tensor spectral and nuclear norms under $\boldsymbol{Q}$-transform)**.** *Define* $\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} := \max_{1\le j\le K} \|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}}$ *and* $\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Tn}} := \sum_{j=1}^{K} \|\widehat{\boldsymbol{X}}_j\|_*$ *where* $\|\cdot\|_{\mathrm{op}}$ *and* $\|\cdot\|_*$ *denote the matrix operator (spectral) norm and nuclear norm, respectively.*

Proposition 1 (Dual norm)**.** *For unitary* $\boldsymbol{Q}$, $\|\cdot\|_{\boldsymbol{Q},\mathrm{Tn}}$ *is the dual norm of* $\|\cdot\|_{\boldsymbol{Q},\mathrm{Ts}}$*, defined as* $\sup_{\|\boldsymbol{\mathcal{D}}\|_{\boldsymbol{Q},\mathrm{Ts}}\le 1}\langle \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{D}}\rangle_{\mathrm{F}} = \|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}}$.

3.1 STEEPEST DESCENT AND T-SVD POLAR

For clarity, we fix the notation in this part as follows. Define the update of TUON by a general $\boldsymbol{Q}$ transform as $\boldsymbol{\mathcal{P}} := \big(\mathrm{polar}\big((\boldsymbol{\mathcal{M}} \times_3 \boldsymbol{Q})_{k\in\{1,\cdots,K\}}\big)\big) \times_3 \boldsymbol{Q}^{\mathsf{H}}$. From Theorem 1, we will show that TUON is normalized steepest descent under the $\boldsymbol{Q}$-transformed tensor spectral norm, rather than a heuristic composition of matrix normalizations. A shared positive scale on all slices only changes the effective trust-region radius ρ, not the update direction.

Theorem 1. *For every radius* $\rho > 0$, $-\rho\boldsymbol{\mathcal{P}} \in \arg\min_{\|\boldsymbol{\mathcal{D}}\|_{\boldsymbol{Q},\mathrm{Ts}}\le\rho}\langle \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{D}}\rangle_{\mathrm{F}}$ *and the optimal value is* $-\rho\|\boldsymbol{\mathcal{M}}\|_{\boldsymbol{Q},\mathrm{Tn}} = -\rho\sum_{j=1}^{K}\|\widehat{\boldsymbol{M}}_j\|_*$.

The update also admits a tensor-algebraic interpretation. Let $\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{U}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{S}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{V}}^{\top}$ be the T-SVD of $\boldsymbol{\mathcal{M}}$ (see Appendix B.5), where $\boldsymbol{\mathcal{S}}$ is f-diagonal with the nonzero singular values of the transformed slices on its diagonal. Let $\boldsymbol{\mathcal{J}}_{\mathrm{supp}}$ be the f-diagonal tensor obtained from $\boldsymbol{\mathcal{S}}$ by replacing every nonzero diagonal entry with 1. Analogously to the matrix polar factor $\boldsymbol{U}\boldsymbol{V}^{\top}$ of $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}$, the tensor polar factor is $\mathrm{tpolar}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}}) := \boldsymbol{\mathcal{U}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{J}}_{\mathrm{supp}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{V}}^{\top}$. The following proposition shows that this tensor-algebraic object is exactly the optimizer map from Theorem 1.

Proposition 2 (tensor polar equivalence)**.** *The optimizer map* $\boldsymbol{\mathcal{P}}$ *equals the partial-isometry factor of the generalized T-SVD:* $\boldsymbol{\mathcal{P}} = \mathrm{tpolar}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}}) = \boldsymbol{\mathcal{U}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{J}}_{\mathrm{supp}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{V}}^{\top}$. *For* $K = 1$*, this reduces to matrix Muon. For* $\boldsymbol{Q} = \boldsymbol{I}$*, it reduces to independent layer-wise polar maps.*

The proposed direction is thus both the exact LMO for a tensor-spectral trust region and the generalized polar factor of a third-order tensor. Our novelty lies in specializing the t-product and tensor polar decomposition to cross-layer optimization, which enables the comparative, convergence, and oracle analyses in § 3.2. These proofs can be found in Appendices H.2 and H.3.

Non-redundancy of the Layer Transform. A natural concern is that the layer transform $\boldsymbol{Q}$ might be algebraically redundant, since the update applies $\boldsymbol{Q}$ and later $\boldsymbol{Q}^{\mathsf{H}}$, the two might simply cancel, leaving no real effect. For a TEON-style unfolding polar, a unitary change of layer basis cancels exactly, so it cannot adapt the update to the transform. In TUON, however, the transform is applied *before* the nonlinear slice-wise polar maps, so it changes the tensor partition. This establishes an algebraic distinction between the two approaches. The following theorem clarifies when this cancellation occurs.

Theorem 2 (Unfolding cancellation versus slice-wise non-cancellation)**.** *Let $\boldsymbol{M}_{(1)} = \mathscr{U}(\boldsymbol{\mathcal{M}})$ and $\boldsymbol{R}_{\boldsymbol{Q}} = \boldsymbol{Q}^{\mathsf{T}} \otimes \boldsymbol{I}_n$. For the matrix polar map, $\mathrm{polar}(\boldsymbol{M}_{(1)}\boldsymbol{R}_{\boldsymbol{Q}})\boldsymbol{R}_{\boldsymbol{Q}}^{\mathsf{H}} = \mathrm{polar}(\boldsymbol{M}_{(1)})$. Hence a unitary layer transform is redundant under a global unfolding polar. In contrast, under the algebra used by TUON, $\big(\mathrm{polar}_{\mathrm{slice}}(\boldsymbol{\mathcal{M}} \times_3 \boldsymbol{Q})\big) \times_3 \boldsymbol{Q}^{\mathsf{H}} \neq \mathrm{polar}_{\mathrm{slice}}(\boldsymbol{\mathcal{M}})$, cancellation does not occur in general.*

Proposition 3. *Let $r_j := \mathrm{rank}(\widehat{\boldsymbol{M}}_j)$. Then $\|\boldsymbol{\mathcal{P}}\|_{\mathrm{F}}^2 = \sum_{j=1}^{K} r_j$ and $\mathrm{RMS}(\boldsymbol{\mathcal{P}}) = \sqrt{\sum_j r_j/(mnK)}$. If slices are full rank, then $\mathrm{RMS}(\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}) = 1/(\sqrt{\max(m,n)})$, independently of K.*

Since the polar map normalizes all nonzero singular values to one, the update energy depends only on the number of active singular directions, not on the momentum magnitude. Proposition 3 gives this energy exactly and shows that stacking more layers does not inflate the RMS by a factor of $\sqrt{K}$, which justifies the stack-scale factor for normalization in §2.2. Proofs are in Appendix H.4 and H.5.

3.2 Comparison with Muon and TEON

In this part, we show that any unitary layer transform distorts the Muon geometry by at most a factor of $\sqrt{K}$, where K is the number of stacked layers. The precise Muon norm pair and its polar LMO are reviewed in Appendix D.1. Here we only use the notation $\|\cdot\|_{\mathrm{Muon}}$ and $\|\cdot\|_{\mathrm{Muon},*}$.

Theorem 3. *Define $\alpha_{\boldsymbol{Q}} := \max_j \sum_{k=1}^{K} |\boldsymbol{Q}_{jk}|$ and $\beta_{\boldsymbol{Q}} := \max_k \sum_{j=1}^{K} |\boldsymbol{Q}_{jk}|$. Since $\boldsymbol{Q}$ is unitary, $\alpha_{\boldsymbol{Q}}, \beta_{\boldsymbol{Q}} \leq \sqrt{K}$. Then, for every tensor $\boldsymbol{\mathcal{X}}$ and for their duals $\boldsymbol{\mathcal{G}}$,*

$$\frac{1}{\beta_{\boldsymbol{Q}}}\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}} \leq \|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} \leq \alpha_{\boldsymbol{Q}}\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}}, \quad \frac{1}{\alpha_{\boldsymbol{Q}}}\|\boldsymbol{\mathcal{G}}\|_{\mathrm{Muon},*} \leq \|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}} \leq \beta_{\boldsymbol{Q}}\|\boldsymbol{\mathcal{G}}\|_{\mathrm{Muon},*}. \tag{6}$$

The bounds, $\frac{1}{\beta_{\boldsymbol{Q}}}\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}} \leq \|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} \leq \alpha_{\boldsymbol{Q}}\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}}$, ensure that the transform neither collapses nor explodes the trust-region scale. Their dual counterparts control the gradient's nuclear norm: the best first-order decrease under the $\boldsymbol{Q}$ geometry is at least $1/\sqrt{K}$ and at most $\sqrt{K}$ times that under the layer-wise Muon geometry. This is the key stability statement: *no unitary layer transform can distort the Muon geometry by more than a factor of $\sqrt{K}$, regardless of the data or the choice of $\boldsymbol{Q}$.*

Corollary 1 (Smoothness comparison)**.** *Let $L_{\boldsymbol{Q}}$ and L_{Muon} be the smallest constants satisfying $\|\nabla F(\boldsymbol{\mathcal{X}}) - \nabla F(\boldsymbol{\mathcal{Y}})\|_{\boldsymbol{Q},\mathrm{Tn}} \leq L_{\boldsymbol{Q}}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Y}}\|_{\boldsymbol{Q},\mathrm{Ts}}$ and the analogous inequality under the Muon norm pair. Then $\frac{1}{\alpha_{\boldsymbol{Q}}^2}L_{\mathrm{Muon}} \leq L_{\boldsymbol{Q}} \leq \beta_{\boldsymbol{Q}}^2 L_{\mathrm{Muon}}$.*

In Corollary 1, the induced smoothness constants differ by at most a factor K. This establishes the maximum scale at which a convergence improvement can arise, while also preventing an unsupported claim of arbitrary acceleration from a mere coordinate change.

Theorem 4 (LMO gain with TEON)**.** *For every unitary $\boldsymbol{Q}$ and tensor $\boldsymbol{\mathcal{X}}$, $\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} \leq \|\boldsymbol{\mathcal{X}}\|_{\mathrm{TEON}} \leq \sqrt{K}\,\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}}$. Consequently, $\|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*} \leq \|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}} \leq \sqrt{K}\,\|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*}$. Thus, at a common radius ρ, $\rho\|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*} \leq \rho\|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}} \leq \sqrt{K}\,\rho\|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*}$.*

Then, we compare the linearized decrease with TEON. The TEON norm/dual norm, and TEON polar update are stated in Appendix D.2. Thus, at equal operator radius, the exact TUON first-order oracle is never weaker than the TEON oracle and may be up to $\sqrt{K}$ stronger. These proofs can be found in Appendix H.8, H.9 and H.10.

3.3 Convergence

Theorem 5. *Let F be the training objective, lower bounded by $F_\star$, η is the step size and T is the number of iterations. Suppose F is lower bounded by $F_\star$ and is $L_{\boldsymbol{Q}}$-smooth under $\|\cdot\|_{\boldsymbol{Q},\mathrm{Ts}}$. For $\boldsymbol{\mathcal{W}}_{t+1} = \boldsymbol{\mathcal{W}}_t - \eta\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\nabla F(\boldsymbol{\mathcal{W}}_t))$, we have $\min_{0\leq t<T}\|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q},\mathrm{Tn}} \leq \frac{F(\boldsymbol{\mathcal{W}}_0)-F_\star}{\eta T} + \frac{L_{\boldsymbol{Q}}\eta}{2}$. Choosing $\eta = \sqrt{\frac{2(F(\boldsymbol{\mathcal{W}}_0)-F_\star)}{L_{\boldsymbol{Q}}T}}$ gives*

$$\min_{0\leq t<T}\|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\mathrm{F}} \leq \min_{0\leq t<T}\|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q},\mathrm{Tn}} \leq \sqrt{\frac{2L_{\boldsymbol{Q}}(F(\boldsymbol{\mathcal{W}}_0) - F_\star)}{T}}. \tag{7}$$

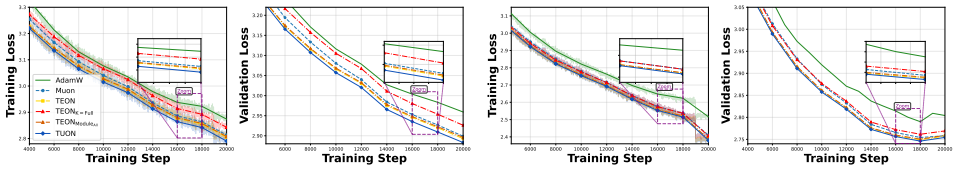
Corollary 1 bounds the smoothness gap between TUON and Muon by at most a factor K. Together with Theorem 5, this yields a convergence improvement of at most $\sqrt{K}$ and a tighter $O(T^{-1/2})$ stationarity certificate. The proof is in Appendix H.11.

4 EXPERIMENTS

We evaluate TUON across three dimensions to thoroughly assess its effectiveness and generality. (i) **Architecture and Scalability.** GPT-124M/350M and Qwen-0.6B/4B verify broad applicability and scalability (§4.1). (ii) **Ablation.** Ablations isolate the transformation mechanism, cross-layer stacking beyond two layers, module adaptability beyond QKV, and robustness to the polarization method (§4.2). (iii) **Cost.** Step-time comparison shows that TUON adds only marginal overhead and scales gracefully (§4.3).

Baselines. We compare TUON against representative optimizers from three levels of structural granularity. (i) **AdamW** (Loshchilov & Hutter, 2017) is the de facto vector-aware standard for large-scale pretraining. (ii) **Muon** (Jordan et al., 2024) is a matrix-aware optimizer that has recently demonstrated remarkable breakthroughs in LLM pretraining. (iii) **TEON** (Zhang et al., 2026a) is a recent tensor-aware optimizer that has consistently outperformed Muon across model scales and architectures. Tensorion is also a tensor-aware optimizer, but it has been validated mainly on vision tasks, so we use TEON rather than Tensorion as our tensor-aware baseline. Together, these baselines span vector-, matrix-, and tensor-level optimization. More details can be found in Appendix I.3.

Experimental details. All baselines follow **the same configurations** , ensuring a fair comparison. Comprehensive experimental details, such as computational resources, configurations, implementation specifics, are elaborated in Appendix I.2. Our implementation strictly follows the PyTorch official API. *Fully functional, production-ready code* is publicly released (see Appendix I.1 for link and implement details).



(a) Qwen 0.6B . (b) Qwen 4B.

Figure 2: Training curves of Qwen models.

4.1 RESULTS ON LLM PRETRAINING

Table 2: Validation loss on GPT-2 (124M, 350M; 50k iterations) and Qwen3 (0.6B, 4B; 20k iterations). $\Delta = \text{Val} - \text{Val}_{\text{AdamW}}$; negative Δ indicates improvement. Best per column in bold. Lower is better.

Optimizer	**124M$_{\text{GPT-2}}$**		**350M$_{\text{GPT-2}}$**		**0.6B$_{\text{Qwen3}}$**		**4B$_{\text{Qwen3}}$**	
	Val	Δ	Val	Δ	Val	Δ	Val	Δ
AdamW	**2.9230**	–	2.7307	–	2.9652	–	2.8108	–
Muon	3.0488	+0.1258	2.7404	+0.0097	2.9056	−0.0596	2.7636	−0.0472
TEON	2.9442	+0.0212	2.6935	−0.0372	2.8978	−0.0674	2.7604	−0.0504
TUON	2.9357	**+0.0127**	**2.6925**	**−0.0382**	**2.8887**	**−0.0765**	**2.7598**	**−0.0510**

Table 2 reports validation loss on GPT-2 (124M, 350M) and Qwen3 (0.6B, 4B). TUON achieves the lowest loss on three of the four settings and consistently outperforms Muon and TEON. Specifically, TUON reduces validation loss over AdamW by 0.0382, 0.0765, and 0.0510 on GPT-2 350M and the two Qwen3 scales, respectively. Moreover, TUON outperforms Muon on all four settings, with the largest gap at 124M (0.1131). This suggests that cross-layer structure is particularly beneficial.

Table 3: Effect of stacking size K. ↑/↓ denote K = full is worse/better than $K = 2$. 20k iterations.

Model	Muon	$\text{TEON}_{\text{mode-1}}$		$\text{TEON}_{\text{mode-2}}$		TUON_{DCT}		$\text{TUON}_{\text{DecorQ}}$	
		K=2	K=full	K=2	K=full	K=2	K=full	K=2	K=full
124M$_{\text{GPT-2}}$	3.0583	3.0255	3.0513(↑)	3.0332	3.0789(↑)	3.0309	3.0295(↓)	3.0299	**3.0233**(↓)
350M$_{\text{GPT-2}}$	2.8427	2.8427	2.8617(↑)	2.8643	2.8762(↑)	2.8150	2.8218(↑)	2.8183	**2.8143**(↓)
0.6B$_{\text{Qwen3}}$	2.9056	2.8994	2.9323(↑)	2.8978	2.9583(↑)	2.9203	2.9210(↑)	2.8938	**2.8887**(↓)
4B$_{\text{Qwen3}}$	2.7636	2.7627	2.7649(↑)	2.7604	2.7640(↑)	2.7606	2.8100(↑)	2.7629	**2.7598** (↓)

Furthermore, TUON improves over TEON on every model, consistent with our motivation and theory in §3.2. Fig. 2 presents the training and validation curves for Qwen3 0.6B and 4B. TUON outperforms AdamW, Muon, TEON, and the TEON variants that stack all layers ($\text{TEON}_{K=\text{Full}}$) or extend to QKVO and MLP ($\text{TEON}_{\text{Module}_{\text{All}}}$). Overall, TUON delivers consistent improvements across two architectures and four scales without per-scale tuning, supporting the claim that a transform-based tensor algebra offers a more effective geometry for cross-layer optimization.

4.2 Ablation Studies

We conduct ablation studies to isolate the effect of each design choice in our method. Unless otherwise noted, we train all models for 20k iterations for computational reasons.

Choices of $\boldsymbol{Q}$. The T-SVD-based algebra only requires $\boldsymbol{Q}$ to be invertible and does not specify which transform suits cross-layer optimization. Our analysis in § 2 suggests that $\boldsymbol{Q}$ should be guided by layer statistics, which we test here by comparing fixed transforms (DCT, Haar, DFT) against data-dependent DecorQ variants (static and residual); details are in Appendix E and I.6.

In Table 7, DecorQ achieves the best results, improving over DCT by 0.0502, while the other four choices perform comparably. It reflects that layer statistics are empirically superior. Moreover, static DecorQ reaches a competitive 2.7658 but remains slightly worse than the adaptive version, indicating that refreshing $\boldsymbol{Q}$ during training provides a consistent benefit. Overall, the ablation confirms that the layer mode should be treated as a structured dimension rather than an arbitrary algebraic index.

Stacking size K. A larger stacking size K exposes more cross-layer structure, and the theoretical analysis in § 3.2 and in (Zhang et al., 2026a) predicts a larger potential gain as K grows. Table 3 tests whether this potential is realized in practice. TEON performs best at $K = 2$ and consistently degrades at K = full for both mode-1 and mode-2 unfolding, exposing a gap between theory and practice. Although larger K promises more gain, unfolding methods lose performance as K grows. Two causes are likely. Unfolding conflates structural information, and aligning top singular vectors across many layers becomes increasingly difficult.

TUON with DecorQ is the only method that improves from $K = 2$ to K = full on all four models: 3.0299 to 3.0233 (GPT-2 124M), 2.8183 to 2.8143 (GPT-2 350M), 2.8938 to 2.8887 (Qwen3 0.6B), and 2.7629 to 2.7598 (Qwen3 4B), confirming its ability to exploit more layers. We also notice that TUON with DCT degrades on three of the four models, isolating the effect of $\boldsymbol{Q}$, i.e., it is not larger K alone that helps, but the ability to adapt the shared basis to the grouped layers. Overall, although the gain remains modest and falls short of the $\sqrt{K}$ upper bound by theory, TUON with DecorQ is the first that makes stacking more layers beneficial rather than harmful, a regime that prior methods cannot reach.

Unfolding mode. TEON requires an additional hyperparameter, i.e., unfold along mode-1 or mode-2. It reports that mode-1 works better in practice (Zhang et al., 2026a). Our experiments reveal that the better mode of TEON is *model-dependent* (Table 3). Specifically, GPT-2 favors mode-1, while Qwen3 favors mode-2. Moreover, choosing the wrong mode costs 0.0216 on GPT-2 350M. The gap is non-negligible, meaning TEON requires a per-model mode choice, and neither mode provides a universally correct alignment. **TUON avoids this hyperparameter by transform-based tensor algebra.** In all main experiments, we report the stronger of the two TEON modes, ensuring that the comparison is against the most favorable TEON configuration. Even under this conservative protocol, TUON outperforms the best TEON on every model in Table 2.

Table 4: Time per iteration (ms) on four models. Lower values indicate faster training.

Model	Time per Iteration (ms)			
	124M$_{\text{GPT-2}}$	**350M$_{\text{GPT-2}}$**	**0.6B$_{\text{Qwen3}}$**	**4B$_{\text{Qwen3}}$**
AdamW	**152.74**	**379.95**	**3265.5**	**5466.9**
Muon	160.88	404.81	3324.3	5794.7
TEON	161.81	408.30	3385.7	6081.5
TUON	162.12	411.64	3378.3	6150.7

Layer type. Our method applies uniformly to all layer types. We verify this by comparing Module$_{\text{QKV}}$ (attention projections only) with Module$_{\text{All}}$ (QKVO and MLP). Table 9 reports the results. Under mode-1, TEON degrades on all four models, and under mode-2 the degradation is even larger, e.g., from 3.0332 to 3.1027 on GPT-2 124M. This confirms that the matricization-based tensor algebra struggles to maintain alignment across heterogeneous layer types under either unfolding mode. TUON behaves differently, extending from Module$_{\text{QKV}}$ to Module$_{\text{All}}$ yields comparable or better losses on GPT-2 124M (3.0225 to 3.0233), GPT-2 350M (2.8291 to 2.8143), Qwen3 0.6B (2.8897 to 2.8887), and Qwen3 4B (2.7606 to 2.7598). Moreover, in the Module$_{\text{All}}$ setting, it outperforms the best TEON by 0.0298, 0.0474, 0.0131, and 0.0041, respectively. These results show that TUON's benefit extends beyond attention to MLP and output projections, where prior tensor-aware optimizers fail to transfer.

Polarization methods. Our optimizer relies on a polar map, approximated in practice by Newton–Schulz (NS) iterations. We test the robustness of TUON to the polarization method by comparing NS ($J = 5$) with PolarExpress (Amsel et al., 2026) on Qwen3-4B. In Table 11, the two methods perform comparably at both stacking sizes. Specifically, at $K = 2$, NS reaches 2.7629 and PolarExpress 2.7638; at full stacking, PolarExpress is slightly better (2.7569 vs. 2.7598). Both differences are an order of magnitude smaller than the gain from the $\boldsymbol{Q}$ mechanism (0.0502), confirming the robustness of TUON.

4.3 Training step time

We compare the per-iteration step time with identical hardware, batch size, sequence length, and parallelism configuration. Table 4 reports the average time per iteration (ms), excluding the first few iterations to remove compilation and initialization overhead. Relative to vector-aware AdamW, TUON is slower by 6.1%, 8.3%, 3.5%, and 12.5% on the four models per iteration, respectively. Compared with matrix-aware and tensor-aware optimizer, TUON is on par with Muon and TEON across all four models, within 0.2%–1.1% of TEON and 1.7%–6.1% of Muon. These results show that TUON introduces only marginal overhead and scales gracefully to larger models, making the per-iteration cost a reasonable price for the optimization benefits reported above.

5 Conclusion

We introduce **TUON**, a tensor-aware optimizer built on a transform-based tensor algebra grounded in the Tensor-SVD. For third-order tensors in Transformers, TUON applies a unitary transform $\boldsymbol{Q}$ along the layer mode and polarizes each transformed slice independently before inverting the transform, a mix-then-separate design that exploits cross-layer structure while preserving per-layer diversity and avoiding the information loss of global unfolding. Empirically, TUON improves consistently over Muon and TEON on GPT-124M/350M and Qwen-0.6B/4B, with ablations confirming its transformation mechanism, cross-layer stacking beyond two layers, adaptability beyond QKV, and robustness to polarization, at only marginal overhead. For future work, although DecorQ realizes the benefits of larger stacking sizes, these gains remain bounded and below the theoretical maximum. Further extending TUON's stacking capability is therefore a promising direction. Two additional directions are extending TUON to other parameter groups and heterogeneous architectures, and co-designing tensor-efficient kernels that exploit the layer-mode transform and slice-wise polarization to reduce memory traffic and improve hardware efficiency.

AI USE STATEMENT

In this work, we used generative AI tools solely for suggested-disclosure tasks to aid writing. These tools are used to improve readability, suggesting alternative phrasing, and refining the overall presentation of our work. We have not used generative AI tools for any required-disclosure tasks as defined by the ICLR policy, such as generating synthetic data, developing theoretical models or conceptual frameworks or mathematical claims, writing proofs, designing experiments or methodology, implementing methods, or interpreting results. We have reviewed all AI-assisted work. All AI-suggested text edits were critically reviewed, revised as needed, and checked for accuracy and consistency by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

See the ICLR 2027 AI Policy for Authors for more details. This statement should not be more than 1 page.

REPRODUCIBILITY STATEMENT

We have taken the following steps to ensure the reproducibility of our work.

Our implementation strictly follows the PyTorch official API,

```
torch.optim
```

. The complete production-ready code is publicly available.

For all results. The complete source code is provided as anonymous material and is also available at https://anonymous.4open.science/r/tuon-802A/README.md.

All computational resources, training details, configurations, baselines and TUON implementation specifics, model architectures and hyper-parameters are provided in Appendices I.1, I.2 and I.4.

For theoretical results. Complete proofs of all claims are provided in Appendix H.1.

We believe the above materials are sufficient for reproducing all main results reported in this paper. Any remaining implementation details not covered here are available upon reasonable request to the authors.

REFERENCES

Noah Amsel, David Persson, Christopher Musco, and Robert M Gower. The polar express: Optimal matrix sign methods and their application to the muon algorithm. In *International Conference on Learning Representations*, volume 2026, pp. 138323–138360, 2026.

Kang An, Yuxing Liu, Rui Pan, Shiqian Ma, Donald Goldfarb, and Tong Zhang. Asgo: Adaptive structured gradient optimization. *arXiv preprint arXiv:2503.20762*, 2025.

Amir Beck. *First-order methods in optimization*. SIAM, 2017.

Vladimir Bogachev, Vladimir Aletov, Alexander Molozhavenko, Sergei Kudriashov, and Maxim Rakhuba. Tensorion: A tensor-aware generalization of the muon optimizer. *arXiv preprint arXiv:2606.25975*, 2026.

Xi Chen, Kaituo Feng, Changsheng Li, Xunhao Lai, Xiangyu Yue, Ye Yuan, and Guoren Wang. Fira: Can we achieve full-rank training of LLMs under low-rank constraint? *arXiv preprint arXiv:2410.01623*, 2024.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233, 2023.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Pre-conditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1842–1850. PMLR, 2018.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Eric Kernfeld, Misha Kilmer, and Shuchin Aeron. Tensor–tensor products with invertible linear transforms. *Linear Algebra and its Applications*, 485:545–570, 2015.

Misha E. Kilmer and Carla D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011. doi: 10.1016/j.laa.2010.09.020.

Gyu Yeol Kim and Min-hwan Oh. Convergence of muon with newton–schulz. *arXiv preprint arXiv:2601.19156*, 2026.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3): 455–500, 2009.

Hao Kong, Canyi Lu, and Zhouchen Lin. Tensor q-rank: New data dependent definition of tensor rank. *Machine Learning*, 110:1867–1900, 2021. doi: 10.1007/s10994-021-05987-8.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*, 2023.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025a.

Liming Liu, Zhenghao Xu, Zixuan Zhang, Hao Kang, Zichong Li, Chen Liang, Weizhu Chen, and Tuo Zhao. Cosmos: A hybrid adaptive optimizer for memory-efficient training of LLMs. *arXiv preprint arXiv:2502.17410*, 2025b.

Ziyue Liu, Ruijie Zhang, Zhengyang Wang, Mingsong Yan, Zi Yang, Paul D Hovland, Bogdan Nicolae, Franck Cappello, Sui Tang, and Zheng Zhang. Cola: Compute-efficient pre-training of LLMs via low-rank activation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4627–4645, 2025c.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. Exact low tubal rank tensor recovery from gaussian measurements. *arXiv preprint arXiv:1806.02511*, 2018.

Canyi Lu, Xi Peng, and Yunchao Wei. Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms. In *2019 IEEE/CVF conference on computer vision and pattern recognition*, pp. 5989–5997. IEEE, 2019.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

Son Nguyen, Bo Liu, Lizhang Chen, and Qiang Liu. Improving adaptive moment optimization via preconditioner diagonalization. *arXiv preprint arXiv:2502.07488*, 2025.

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained lmos. *arXiv preprint arXiv:2502.07529*, 2025.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pp. 4596–4604. PMLR, 2018.

Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3): 279–311, 1966.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam for language modeling. In *International Conference on Learning Representations*, volume 2025, pp. 93423–93444, 2025.

Sike Wang, Pan Zhou, Jia Li, and Hua Huang. 4-bit shampoo for memory-efficient network training. *Advances in Neural Information Processing Systems*, 37:126997–127029, 2024.

Shuo Xie, Tianhao Wang, Sashank Reddi, Sanjiv Kumar, and Zhiyuan Li. Structured preconditioners in adaptive optimization: A unified analysis. *arXiv preprint arXiv:2503.10537*, 2025.

Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Yan Yang, Bin Gao, and Ya-xiang Yuan. Variational analysis of determinantal varieties. *arXiv preprint arXiv:2511.22613*, 2025.

Yan Yang, Bin Gao, and Ya-xiang Yuan. Optimization over the intersection of manifolds. *arXiv preprint arXiv:2605.22736*, 2026a.

Yan Yang, Bin Gao, and Ya-xiang Yuan. A space-decoupling framework for optimization on bounded-rank matrices with orthogonally invariant constraints: Y. yang et al. *Mathematical Programming*, pp. 1–53, 2026b.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

Ziming Yu, Pan Zhou, Sike Wang, Jia Li, Mi Tian, and Hua Huang. Zeroth-order fine-tuning of LLMs in random subspaces. *arXiv preprint arXiv:2410.08989*, 2024.

Ruijie Zhang, Yequan Zhao, Ziyue Liu, Zhengyang Wang, Dongyang Li, Yupeng Su, Sijia Liu, and Zheng Zhang. Teon: Tensorized orthonormalization beyond layer-wise muon for large language model pre-training. *arXiv preprint arXiv:2601.23261*, 2026a.

Ruijie Ray Zhang, Ziyue Alvin Liu, Zhengyang Wang, and Zheng Zhang. Lax: Boosting low-rank training of foundation models via latent crossing. *Advances in Neural Information Processing Systems*, 38:142920–142948, 2026b.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik P Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. *arXiv preprint arXiv:2406.16793*, 2024.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient LLM training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

Jingjing Zheng, Anda Tang, Qiangqiang Mao, Zhouchen Lin, and Yankai Cao. Refta: Breaking the weight reconstruction bottleneck in tensorized parameter-efficient fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26369–26378, 2026.

Yu-Bang Zheng, Ting-Zhu Huang, Xi-Le Zhao, Tai-Xiang Jiang, Teng-Yu Ji, and Tian-Hui Ma. Tensor n-tubal rank and its convex relaxation for low-rank tensor recovery. *Information Sciences*, 532:170–189, 2020.

Pan Zhou, Xingyu Xie, and Shuicheng Yan. Win: Weight-decay-integrated nesterov acceleration for adaptive gradient algorithms. 2023.

Pan Zhou, Xingyu Xie, Zhouchen Lin, Kim-Chuan Toh, and Shuicheng Yan. Win: Weight-decay-integrated nesterov acceleration for faster network training. *Journal of Machine Learning Research*, 25(83):1–74, 2024.

CONTENTS

1 Introduction **1**

2 Methodology of TUON **3**

2.1 Momentum tensor and layer transform 3

2.2 Polarization and inverse transform 3

3 Steepest descent theory and comparative advantages **5**

3.1 Steepest descent and T-SVD polar 5

3.2 Comparison with Muon and TEON 6

3.3 Convergence 6

4 Experiments **7**

4.1 Results on LLM pretraining 7

4.2 Ablation studies 8

4.3 Training step time 9

5 Conclusion **9**

A Unified notation **15**

B Preliminaries **16**

B.1 Tensor notation and unitary mode-3 transforms 16

B.2 SVD, matrix norms, and the polar factor 17

B.3 Dual norms, norm ordering, and linear minimization oracles (LMO) 19

B.4 The Q-transform spectral quantity is a norm 20

B.5 T-product and T-SVD tensor algebra 20

B.6 Smoothness and the descent lemma 21

C Related work **22**

D Muon and TEON Geometries **25**

D.1 Muon 25

D.2 TEON 25

D.3 How a layer transform acts on the TEON unfolding 26

E Selection of Q **26**

E.1 Momentum tensor and layer transform 26

F Polarization via Newton–Schulz iteration **29**

G Stack scaling and weight update **30**

G.1 Computational complexity analysis 30

H Detailed proofs for Section 3 **31**

H.1 Proof of Proposition 1: duality of the Q-block norms 31

H.2 Proof of Theorem 1: exact normalized steepest descent 32

H.3 Proof of Proposition 2: tensor-polar equivalence 32

H.4 Proof of Theorem 2: cancellation and non-cancellation 33

H.5 Proof of Proposition 3: exact update energy 34

H.6 Non-unitary transforms 35

H.7 A time-varying unitary basis 36

H.8 Proof of Theorem 3: norm comparability 37

H.9 Proof of Corollary 1: smoothness constants 39

H.10 Proof of Theorem 4: comparison with TEON 39

H.11 Proof of Theorem 5: convergence 41

I Experiments **42**

I.1 Code and reproducibility 42

I.2 Datasets, models and training configuration. 42

I.3 Baselines. 43

I.4 Hyperparameters 44

I.5 Results on LLM pretraining 44

I.6 Ablation studies 45

I.7 Training time comparison 49

A Unified notation

For one architecture-compatible group of K matrices, we use

Symbol	Meaning
$\boldsymbol{W}_t^{(k)}, \boldsymbol{G}_t^{(k)}, \boldsymbol{M}_t^{(k)} \in \mathbb{R}^{m\times n}$	parameter, stochastic gradient, and momentum of layer slice k
$\boldsymbol{\mathcal{W}}_t, \boldsymbol{\mathcal{G}}_t, \boldsymbol{\mathcal{M}}_t \in \mathbb{R}^{m\times n\times K}$	their stacked tensors
$\boldsymbol{Q}$ or $\boldsymbol{Q}_t$	unitary layer-mode transform
$\widehat{\boldsymbol{\mathcal{M}}}_t = \boldsymbol{\mathcal{M}}_t \times_3 \boldsymbol{Q}$	momentum in transformed layer coordinates
M_t	mode-3 unfolding of $\boldsymbol{\mathcal{M}}_t$
$\mathsf{C}_t = \mathsf{M}_t\mathsf{M}_t^{\mathsf{T}}$	layer Gram matrix
$\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}})$ or $\boldsymbol{\mathcal{P}}$	exact transform–slice-polar–inverse-transform map
$\mathrm{tpolar}_{\boldsymbol{Q}}(\cdot)$	generalized tensor polar factor
$\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{V}}$	T-SVD factors of $\boldsymbol{\mathcal{M}}$
$\boldsymbol{\mathcal{J}}_{\mathrm{supp}}$	support indicator tensor
γ_t	shared stack-scale factor
ϵ	small constant for numerical stability
J	number of Newton–Schulz iterations
I	refresh interval for $\boldsymbol{Q}_t$
η_t, β, λ	learning rate, momentum coefficient, and decoupled weight decay
η	fixed step size used in the theory
T	number of iterations in the convergence analysis
$F_\star$	lower bound of the objective
ρ_t	trust-region radius in the theory
$\|\cdot\|_{\boldsymbol{Q},\mathrm{Ts}}$	Q-block spectral norm
$\|\cdot\|_{\boldsymbol{Q},\mathrm{Tn}}$	Q-block nuclear norm (dual of $\|\cdot\|_{\boldsymbol{Q},\mathrm{Ts}}$)
$\|\cdot\|_{\mathrm{Muon}}, \|\cdot\|_{\mathrm{TEON}}$	Muon and TEON spectral norms
L_Q	smoothness constant under the Q-block norm pair
α_Q, β_Q	transform spread constants
$\mathscr{U}(\cdot)$	mode-1 unfolding operator
$R_Q = Q^{\mathsf{T}} \otimes I_n$	unfolding transformation matrix

This convention is used in the method, theory, experiments, and appendices.

B PRELIMINARIES

This appendix collects the elementary facts used in Section 3. They are stated explicitly so that the proofs can be followed using only basic linear algebra and first-order optimization.

B.1 TENSOR NOTATION AND UNITARY MODE-3 TRANSFORMS

For $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{m\times n\times K}$, its kth frontal slice is $\boldsymbol{X}_k = \boldsymbol{\mathcal{X}}(:,:,k)$. A mode-3 transform by $\boldsymbol{Q} \in \mathbb{C}^{K\times K}$ is

$$\widehat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}} \times_3 \boldsymbol{Q}, \qquad \widehat{\boldsymbol{X}}_j = \sum_{k=1}^{K} \boldsymbol{Q}_{jk}\boldsymbol{X}_k. \tag{8}$$

The Frobenius inner product and norm are

$$\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle_{\mathrm{F}} := \sum_{k=1}^{K} \mathrm{tr}(\boldsymbol{X}_k^{\mathsf{H}}\boldsymbol{Y}_k), \qquad \|\boldsymbol{\mathcal{X}}\|_{\mathrm{F}}^2 := \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle_{\mathrm{F}}. \tag{9}$$

Lemma 1 (Parseval identities for a unitary layer transform)**.** *If $\boldsymbol{Q}^{\mathsf{H}}\boldsymbol{Q} = \boldsymbol{I}$, then*

$$\langle \boldsymbol{\mathcal{X}} \times_3 \boldsymbol{Q}, \boldsymbol{\mathcal{Y}} \times_3 \boldsymbol{Q} \rangle_{\mathrm{F}} = \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle_{\mathrm{F}}, \qquad \|\boldsymbol{\mathcal{X}} \times_3 \boldsymbol{Q}\|_{\mathrm{F}} = \|\boldsymbol{\mathcal{X}}\|_{\mathrm{F}}. \tag{10}$$

Proof. We prove the inner-product preservation by expanding the Frobenius inner product entry-wise. The norm preservation then follows as a special case.

First, we express the tensor in terms of its tubes. For each position (a, b) in the $m \times n$ matrix, define the tube vector

$$\boldsymbol{x}_{ab} := \begin{bmatrix} \boldsymbol{X}_1(a,b) \\ \boldsymbol{X}_2(a,b) \\ \vdots \\ \boldsymbol{X}_K(a,b) \end{bmatrix} \in \mathbb{C}^K, \qquad \boldsymbol{y}_{ab} := \begin{bmatrix} \boldsymbol{Y}_1(a,b) \\ \boldsymbol{Y}_2(a,b) \\ \vdots \\ \boldsymbol{Y}_K(a,b) \end{bmatrix} \in \mathbb{C}^K. \tag{11}$$

These vectors collect the values along the third mode (the layer dimension) for a fixed matrix entry (a, b). The Frobenius inner product of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ can then be written as

$$\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle_{\mathrm{F}} = \sum_{a=1}^{m} \sum_{b=1}^{n} \boldsymbol{x}_{ab}^{\mathsf{H}} \boldsymbol{y}_{ab}. \tag{12}$$

Indeed, $\boldsymbol{x}_{ab}^{\mathsf{H}} \boldsymbol{y}_{ab} = \sum_{k=1}^{K} \overline{\boldsymbol{X}_k(a,b)} \boldsymbol{Y}_k(a,b)$, and summing over all (a, b) recovers the standard Frobenius inner product.

Second, we express the transformed tensor in terms of the same tubes. Applying $\boldsymbol{Q}$ along the third mode transforms each tube vector independently:

$$\widehat{\boldsymbol{X}}_j(a,b) = \sum_{k=1}^{K} \boldsymbol{Q}_{jk} \boldsymbol{X}_k(a,b), \tag{13}$$

which means that the transformed tube at position (a, b) is

$$\widehat{\boldsymbol{x}}_{ab} := \begin{bmatrix} \widehat{\boldsymbol{X}}_1(a,b) \\ \widehat{\boldsymbol{X}}_2(a,b) \\ \vdots \\ \widehat{\boldsymbol{X}}_K(a,b) \end{bmatrix} = \boldsymbol{Q} \boldsymbol{x}_{ab}. \tag{14}$$

Similarly, $\widehat{\boldsymbol{y}}_{ab} = \boldsymbol{Q} \boldsymbol{y}_{ab}$.

Then we compute the inner product of the transformed tensors. Using the same tube-sum identity as in Step 1, the inner product of the transformed tensors is

$$\langle \boldsymbol{\mathcal{X}} \times_3 \boldsymbol{Q}, \boldsymbol{\mathcal{Y}} \times_3 \boldsymbol{Q} \rangle_{\mathrm{F}} = \sum_{a=1}^{m} \sum_{b=1}^{n} \widehat{\boldsymbol{x}}_{ab}^{\mathsf{H}} \widehat{\boldsymbol{y}}_{ab}. \tag{15}$$

Substituting $\widehat{\boldsymbol{x}}_{ab} = \boldsymbol{Q} \boldsymbol{x}_{ab}$ and $\widehat{\boldsymbol{y}}_{ab} = \boldsymbol{Q} \boldsymbol{y}_{ab}$ gives

$$\widehat{\boldsymbol{x}}_{ab}^{\mathsf{H}} \widehat{\boldsymbol{y}}_{ab} = (\boldsymbol{Q} \boldsymbol{x}_{ab})^{\mathsf{H}} (\boldsymbol{Q} \boldsymbol{y}_{ab}) = \boldsymbol{x}_{ab}^{\mathsf{H}} \boldsymbol{Q}^{\mathsf{H}} \boldsymbol{Q} \boldsymbol{y}_{ab}. \tag{16}$$

Since $\boldsymbol{Q}$ is unitary, $\boldsymbol{Q}^{\mathsf{H}} \boldsymbol{Q} = \boldsymbol{I}$, so

$$\widehat{\boldsymbol{x}}_{ab}^{\mathsf{H}} \widehat{\boldsymbol{y}}_{ab} = \boldsymbol{x}_{ab}^{\mathsf{H}} \boldsymbol{y}_{ab}. \tag{17}$$

At last, we sum over all entries. Summing the identity $\widehat{\boldsymbol{x}}_{ab}^{\mathsf{H}} \widehat{\boldsymbol{y}}_{ab} = \boldsymbol{x}_{ab}^{\mathsf{H}} \boldsymbol{y}_{ab}$ over all $m \times n$ entries (a, b) yields

$$\langle \boldsymbol{\mathcal{X}} \times_3 \boldsymbol{Q}, \boldsymbol{\mathcal{Y}} \times_3 \boldsymbol{Q} \rangle_{\mathrm{F}} = \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle_{\mathrm{F}}. \tag{18}$$

This proves the inner-product preservation.

Moreover, we can have the norm preservation as a corollary. Setting $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}}$ in the inner-product preservation gives

$$\| \boldsymbol{\mathcal{X}} \times_3 \boldsymbol{Q} \|_{\mathrm{F}}^2 = \| \boldsymbol{\mathcal{X}} \|_{\mathrm{F}}^2. \tag{19}$$

Taking square roots (both sides are nonnegative) yields the norm-preservation identity. □

B.2 SVD, MATRIX NORMS, AND THE POLAR FACTOR

Let $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ have compact singular value decomposition

$$\boldsymbol{A} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^{\mathsf{H}}, \qquad \boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_r), \qquad r = \mathrm{rank}(\boldsymbol{A}). \tag{20}$$

The spectral, nuclear, and Frobenius norms are

$$\|\boldsymbol{A}\|_{\mathrm{op}} = \sigma_1, \qquad \|\boldsymbol{A}\|_* = \sum_{i=1}^{r} \sigma_i, \qquad \|\boldsymbol{A}\|_{\mathrm{F}}^2 = \sum_{i=1}^{r} \sigma_i^2. \tag{21}$$

The canonical polar factor of a nonzero matrix $\boldsymbol{A}$ is defined using its compact singular value decomposition $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{H}}$. Specifically, we set

$$\mathrm{polar}(\boldsymbol{A}) := \boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}. \tag{22}$$

This factor acts as the unitary part of the polar decomposition, i.e. it retains the singular vectors of $\boldsymbol{A}$ while replacing all nonzero singular values by one. For a zero matrix, we define $\mathrm{polar}(\mathbf{0}) := \mathbf{0}$.

Lemma 2 (Elementary polar identities)**.** *For $\boldsymbol{A} \neq 0$ with rank r,*

$$\|\,\mathrm{polar}(\boldsymbol{A})\|_{\mathrm{op}} = 1, \qquad \|\,\mathrm{polar}(\boldsymbol{A})\|_{\mathrm{F}}^2 = r, \qquad \langle \boldsymbol{A}, \mathrm{polar}(\boldsymbol{A})\rangle_{\mathrm{F}} = \|\boldsymbol{A}\|_*. \tag{23}$$

Proof. The nonzero singular values of $\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}$ are all one. Therefore its largest singular value is one and its squared Frobenius norm is the sum of r ones. For the inner product,

$$\begin{aligned} \langle \boldsymbol{A}, \boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\rangle_{\mathrm{F}} &= \mathrm{tr}\big(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\big) && (24)\\ &= \mathrm{tr}\big(\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}\big) && (25)\\ &= \mathrm{tr}(\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{H}}) = \mathrm{tr}(\boldsymbol{\Sigma}) = \sum_{i=1}^{r} \sigma_i = \|\boldsymbol{A}\|_*. && (26) \end{aligned}$$

The quantity is real and nonnegative, so taking its real part does not change it. □

Lemma 3 (Operator–nuclear Hölder inequality)**.** *For matrices of the same shape,*

$$|\langle \boldsymbol{A}, \boldsymbol{B}\rangle_{\mathrm{F}}| \le \|\boldsymbol{A}\|_* \|\boldsymbol{B}\|_{\mathrm{op}}. \tag{27}$$

Equality in the maximization over $\|\boldsymbol{B}\|_{\mathrm{op}} \le 1$ is attained by $\boldsymbol{B} = \mathrm{polar}(\boldsymbol{A})$.

Proof. Write $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{H}}$. Cyclicity of trace gives

$$\langle \boldsymbol{A}, \boldsymbol{B}\rangle_{\mathrm{F}} = \mathrm{tr}(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{\Sigma}\boldsymbol{U}^{\mathsf{H}}\boldsymbol{B}\boldsymbol{V}). \tag{28}$$

Let $\boldsymbol{C} = \boldsymbol{U}^{\mathsf{H}}\boldsymbol{B}\boldsymbol{V}$. Multiplication by matrices with orthonormal columns cannot increase the spectral norm, so $\|\boldsymbol{C}\|_{\mathrm{op}} \le \|\boldsymbol{B}\|_{\mathrm{op}}$. Since every diagonal entry satisfies $|\boldsymbol{C}_{ii}| \le \|\boldsymbol{C}\|_{\mathrm{op}}$,

$$\begin{aligned} |\,\mathrm{tr}(\boldsymbol{\Sigma}\boldsymbol{C})| &= \left|\sum_i \sigma_i \boldsymbol{C}_{ii}\right| \le \sum_i \sigma_i |\boldsymbol{C}_{ii}| && (29)\\ &\le \left(\sum_i \sigma_i\right) \|\boldsymbol{C}\|_{\mathrm{op}} \le \|\boldsymbol{A}\|_* \|\boldsymbol{B}\|_{\mathrm{op}}. && (30) \end{aligned}$$

If $\|\boldsymbol{B}\|_{\mathrm{op}} \le 1$, Lemma 2 shows that $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}$ is feasible and gives inner product $\|\boldsymbol{A}\|_*$. Hence

$$\sup_{\|\boldsymbol{B}\|_{\mathrm{op}} \le 1} \langle \boldsymbol{A}, \boldsymbol{B}\rangle_{\mathrm{F}} = \|\boldsymbol{A}\|_*. \tag{31}$$

□

Lemma 4 (Rank-based norm bounds)**.** *If $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ and $r_0 := \min(m, n)$, then*

$$\|\boldsymbol{A}\|_* \le r_0 \|\boldsymbol{A}\|_{\mathrm{op}}, \qquad \|\boldsymbol{A}\|_{\mathrm{F}}^2 \le r_0 \|\boldsymbol{A}\|_{\mathrm{op}}^2. \tag{32}$$

Proof. There are at most r_0 nonzero singular values and each is at most $\sigma_1 = \|\boldsymbol{A}\|_{\mathrm{op}}$. Therefore

$$\sum_i \sigma_i \le r_0 \sigma_1, \qquad \sum_i \sigma_i^2 \le r_0 \sigma_1^2. \tag{33}$$

□

B.3 Dual norms, norm ordering, and linear minimization oracles (LMO)

We begin by recalling the fundamental duality between measuring the size of a direction and measuring the size of a gradient. Given a norm $\|\cdot\|$ that defines the geometry of our trust region, its dual norm is defined as

$$\|\boldsymbol{g}\|_* := \sup_{\|\boldsymbol{d}\|\le 1} \langle \boldsymbol{g}, \boldsymbol{d}\rangle. \tag{34}$$

This quantity measures the maximum possible alignment between a fixed vector $\boldsymbol{g}$ (typically a gradient) and any direction $\boldsymbol{d}$ lying within the unit ball of the primal norm. Intuitively, the dual norm tells us how "large" the gradient appears when measured against the geometry induced by $\|\cdot\|$.

Given a trust-region radius $\rho > 0$, the linear minimization oracle (LMO) seeks the direction within the scaled ball $\|\boldsymbol{d}\| \le \rho$ that minimizes the first-order approximation of the objective:

$$\arg\min_{\|\boldsymbol{d}\|\le\rho} \langle \boldsymbol{g}, \boldsymbol{d}\rangle. \tag{35}$$

Because the objective $\langle \boldsymbol{g}, \boldsymbol{d}\rangle$ is linear in $\boldsymbol{d}$, the minimum over a symmetric convex ball is always attained on the boundary, pointing in the direction opposite to $\boldsymbol{g}$. More precisely, by the definition of the dual norm in equation 34, the optimal value is

$$\min_{\|\boldsymbol{d}\|\le\rho} \langle \boldsymbol{g}, \boldsymbol{d}\rangle = -\rho\|\boldsymbol{g}\|_*. \tag{36}$$

The direction that achieves this minimum is called the *normalized steepest-descent direction* with respect to the norm $\|\cdot\|$. It points in the direction of the negative dual certificate and has norm exactly ρ.

Lemma 5 (Order reversal for dual norms)**.** *Suppose two norms satisfy*

$$a\|\boldsymbol{x}\|_A \le \|\boldsymbol{x}\|_B \le b\|\boldsymbol{x}\|_A \qquad \textit{for all } \boldsymbol{x}, \tag{37}$$

where $a, b > 0$. Then their duals satisfy

$$\frac{1}{b}\|\boldsymbol{g}\|_{A,*} \le \|\boldsymbol{g}\|_{B,*} \le \frac{1}{a}\|\boldsymbol{g}\|_{A,*}. \tag{38}$$

Proof. We prove the two inequalities in equation 38 separately.

First, for the lower bound: $\|\boldsymbol{g}\|_{B,*} \ge \frac{1}{b}\|\boldsymbol{g}\|_{A,*}$.

The condition $\|\boldsymbol{x}\|_B \le b\|\boldsymbol{x}\|_A$ tells us that the B-ball is larger than the A-ball scaled by b. More precisely, any vector with $\|\boldsymbol{x}\|_A \le 1/b$ must satisfy $\|\boldsymbol{x}\|_B \le 1$. Hence

$$\{\boldsymbol{x} : \|\boldsymbol{x}\|_A \le 1/b\} \;\subseteq\; \{\boldsymbol{x} : \|\boldsymbol{x}\|_B \le 1\}. \tag{39}$$

Taking the supremum of the linear functional $\langle \boldsymbol{g}, \boldsymbol{x}\rangle$ over a larger set can only increase the result. Therefore

$$\|\boldsymbol{g}\|_{B,*} = \sup_{\|\boldsymbol{x}\|_B\le 1} \langle \boldsymbol{g}, \boldsymbol{x}\rangle \ge \sup_{\|\boldsymbol{x}\|_A\le 1/b} \langle \boldsymbol{g}, \boldsymbol{x}\rangle = \frac{1}{b}\sup_{\|\boldsymbol{y}\|_A\le 1} \langle \boldsymbol{g}, \boldsymbol{y}\rangle = \frac{1}{b}\|\boldsymbol{g}\|_{A,*}. \tag{40}$$

The third equality uses the change of variables $\boldsymbol{x} = \boldsymbol{y}/b$.

Second, for the upper bound: $\|\boldsymbol{g}\|_{B,*} \le \frac{1}{a}\|\boldsymbol{g}\|_{A,*}$.

Similarly, the condition $a\|\boldsymbol{x}\|_A \le \|\boldsymbol{x}\|_B$ implies

$$\{\boldsymbol{x} : \|\boldsymbol{x}\|_B \le 1\} \;\subseteq\; \{\boldsymbol{x} : \|\boldsymbol{x}\|_A \le 1/a\}. \tag{41}$$

Taking the supremum of $\langle \boldsymbol{g}, \boldsymbol{x}\rangle$ over the smaller set (the B-ball) gives a value no larger than taking it over the larger set (the A-ball):

$$\|\boldsymbol{g}\|_{B,*} = \sup_{\|\boldsymbol{x}\|_B\le 1} \langle \boldsymbol{g}, \boldsymbol{x}\rangle \le \sup_{\|\boldsymbol{x}\|_A\le 1/a} \langle \boldsymbol{g}, \boldsymbol{x}\rangle = \frac{1}{a}\|\boldsymbol{g}\|_{A,*}. \tag{42}$$

Again, the last equality follows from $\boldsymbol{x} = \boldsymbol{y}/a$. □

B.4 THE Q-TRANSFORM SPECTRAL QUANTITY IS A NORM

Lemma 6 (tensor spectral norm under $\boldsymbol{Q}$-transform axioms)**.** *If $\boldsymbol{Q}$ is invertible, then*

$$\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} := \max_j \|(\boldsymbol{\mathcal{X}} \times_3 \boldsymbol{Q})_j\|_{\mathrm{op}} \tag{43}$$

is a norm.

Proof. We verify the three norm axioms.

Positive definiteness. Every spectral norm is nonnegative, hence the maximum is nonnegative. If $\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} = 0$, then every transformed slice is zero, so $\boldsymbol{\mathcal{X}} \times_3 \boldsymbol{Q} = 0$. Since $\boldsymbol{Q}$ is invertible, applying $\boldsymbol{Q}^{-1}$ gives $\boldsymbol{\mathcal{X}} = 0$. Conversely, $\boldsymbol{\mathcal{X}} = 0$ clearly gives norm zero.

Absolute homogeneity. For any scalar c,

$$\|c\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} = \max_j \|c\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}} = |c| \max_j \|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}} = |c|\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}}. \tag{44}$$

Triangle inequality. Linearity of the transform and the matrix operator-norm triangle inequality give

$$\|\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{Y}}\|_{\boldsymbol{Q},\mathrm{Ts}} = \max_j \|\widehat{\boldsymbol{X}}_j + \widehat{\boldsymbol{Y}}_j\|_{\mathrm{op}} \tag{45}$$

$$\leq \max_j \big(\|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}} + \|\widehat{\boldsymbol{Y}}_j\|_{\mathrm{op}}\big) \tag{46}$$

$$\leq \max_j \|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}} + \max_j \|\widehat{\boldsymbol{Y}}_j\|_{\mathrm{op}}. \tag{47}$$

□

B.5 T-PRODUCT AND T-SVD TENSOR ALGEBRA

We begin by introducing the foundational algebraic framework for third-order tensors under invertible linear transforms. Throughout this section, we let $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ denote a third-order tensor, and we fix an orthogonal linear transform $\boldsymbol{Q} \in \mathbb{R}^{I_3 \times I_3}$ satisfying $\boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{I}$. The application of $\boldsymbol{Q}$ to a tensor is understood mode-wise along the third dimension, i.e., $\boldsymbol{Q}(\boldsymbol{\mathcal{A}}) := \boldsymbol{\mathcal{A}} \times_3 \boldsymbol{Q}$.

Definition 2 (t-product under orthogonal transform (Kernfeld et al., 2015))**.** *For tensors $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times r \times I_3}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{r \times I_2 \times I_3}$, their t-product, denoted by $\boldsymbol{\mathcal{A}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{B}}$, is the tensor in $\mathbb{R}^{I_1 \times I_2 \times I_3}$ defined as*

$$\boldsymbol{\mathcal{A}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{B}} \;:=\; \boldsymbol{Q}^{-1}\big(\boldsymbol{Q}(\boldsymbol{\mathcal{A}}) \odot \boldsymbol{Q}(\boldsymbol{\mathcal{B}})\big), \tag{48}$$

where $\odot$ denotes the slice-wise matrix product: for each $i_3 = 1, \ldots, I_3$,

$$\big[\boldsymbol{Q}(\boldsymbol{\mathcal{A}}) \odot \boldsymbol{Q}(\boldsymbol{\mathcal{B}})\big]_{:,:,i_3} \;=\; \big[\boldsymbol{Q}(\boldsymbol{\mathcal{A}})\big]_{:,:,i_3} \cdot \big[\boldsymbol{Q}(\boldsymbol{\mathcal{B}})\big]_{:,:,i_3}.$$

Definition 3 (f-diagonal tensor (Kilmer & Martin, 2011))**.** *A tensor $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is said to be f-*diagonal *if every frontal slice $\boldsymbol{\mathcal{S}}_{:,:,i_3}$ is a diagonal matrix for all $i_3 \in \{1, \ldots, I_3\}$.*

Definition 4 (Identity tensor under $\boldsymbol{Q}$ (Kernfeld et al., 2015))**.** *The tensor $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{I_1 \times I_1 \times I_3}$ is called the* identity tensor *(with respect to $\boldsymbol{Q}$) if its transformed version $\boldsymbol{Q}(\boldsymbol{\mathcal{I}})$ has each frontal slice equal to the identity matrix $\boldsymbol{I}_{I_1} \in \mathbb{R}^{I_1 \times I_1}$.*

Definition 5 (Tensor conjugate transpose (Kernfeld et al., 2015))**.** *For a tensor $\boldsymbol{\mathcal{O}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, its* conjugate transpose *$\boldsymbol{\mathcal{O}}^\top \in \mathbb{R}^{I_2 \times I_1 \times I_3}$ is defined slice-wise in the transformed domain:*

$$\big[\boldsymbol{Q}(\boldsymbol{\mathcal{O}}^\top)\big]_{:,:,i_3} \;=\; \big(\big[\boldsymbol{Q}(\boldsymbol{\mathcal{O}})\big]_{:,:,i_3}\big)^\top, \qquad i_3 = 1, \ldots, I_3.$$

Definition 6 (Orthogonal tensor (Kernfeld et al., 2015))**.** *A tensor $\boldsymbol{\mathcal{O}} \in \mathbb{R}^{I_1 \times I_1 \times I_3}$ is called* orthogonal *(under the t-product $*_{\boldsymbol{Q}}$) if it satisfies*

$$\boldsymbol{\mathcal{O}}^\top *_{\boldsymbol{Q}} \boldsymbol{\mathcal{O}} \;=\; \boldsymbol{\mathcal{O}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{O}}^\top \;=\; \boldsymbol{\mathcal{I}},$$

where $\boldsymbol{\mathcal{I}}$ is the identity tensor from Definition 4.

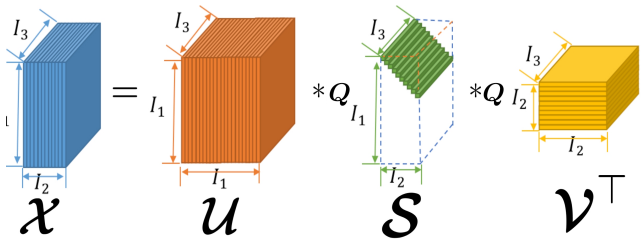


Figure 3: Illustrations of T-SVD

Theorem 6. *(T-SVD) (Kernfeld et al., 2015) Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. Then it can be factorized as $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{U}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{S}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{V}}^\top$, where $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{I_1 \times I_1 \times I_3}$, $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{I_2 \times I_2 \times I_3}$ are orthogonal, and $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is a f-diagonal tensor.*

The T-SVD in Theorem 6 is illustrated in figure 3.

The slice decoupling is all that is needed in the tensor-polar proof.

B.6 Smoothness and the descent lemma

Let $\|\cdot\|$ be a norm and $\|\cdot\|_*$ its dual. A differentiable function is L-smooth Beck (2017) in this norm pair if

$$\|\nabla F(\boldsymbol{X}) - \nabla F(\boldsymbol{Y})\|_* \leq L\|\boldsymbol{X} - \boldsymbol{Y}\|. \tag{49}$$

Lemma 7 (Descent lemma in a general norm)**.** *Let $F : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable function, and let $\|\cdot\|$ be a norm on $\mathbb{R}^d$ with dual norm $\|\cdot\|_*$. Under equation 49, i.e., its gradient is L-Lipschitz under the dual norm, then, for any $\boldsymbol{X}, \boldsymbol{D} \in \mathbb{R}^d$,*

$$F(\boldsymbol{X} + \boldsymbol{D}) \leq F(\boldsymbol{X}) + \langle \nabla F(\boldsymbol{X}), \boldsymbol{D} \rangle + \frac{L}{2}\|\boldsymbol{D}\|^2. \tag{50}$$

Proof. We prove the descent lemma by integrating the smoothness condition along the line segment from $\boldsymbol{X}$ to $\boldsymbol{X} + \boldsymbol{D}$.

First, we give the exact integral representation. Define the scalar function $\phi(s) := F(\boldsymbol{X} + s\boldsymbol{D})$ for $s \in [0, 1]$. By the fundamental theorem of calculus,

$$F(\boldsymbol{X} + \boldsymbol{D}) - F(\boldsymbol{X}) = \phi(1) - \phi(0) = \int_0^1 \phi'(s)\, ds. \tag{51}$$

Applying the chain rule gives $\phi'(s) = \langle \nabla F(\boldsymbol{X} + s\boldsymbol{D}), \boldsymbol{D} \rangle$, hence

$$F(\boldsymbol{X} + \boldsymbol{D}) - F(\boldsymbol{X}) = \int_0^1 \langle \nabla F(\boldsymbol{X} + s\boldsymbol{D}), \boldsymbol{D} \rangle\, ds. \tag{52}$$

To isolate the first-order term, we add and subtract $\langle \nabla F(\boldsymbol{X}), \boldsymbol{D} \rangle$ inside the integral:

$$F(\boldsymbol{X} + \boldsymbol{D}) - F(\boldsymbol{X}) = \int_0^1 \langle \nabla F(\boldsymbol{X}), \boldsymbol{D} \rangle\, ds + \int_0^1 \langle \nabla F(\boldsymbol{X} + s\boldsymbol{D}) - \nabla F(\boldsymbol{X}), \boldsymbol{D} \rangle\, ds \tag{53}$$

$$= \langle \nabla F(\boldsymbol{X}), \boldsymbol{D} \rangle + \int_0^1 \langle \nabla F(\boldsymbol{X} + s\boldsymbol{D}) - \nabla F(\boldsymbol{X}), \boldsymbol{D} \rangle\, ds. \tag{54}$$

The first term is exactly the linear approximation at $\boldsymbol{X}$; the integral captures the accumulated deviation of the gradient along the path.

Then we bound the integral using smoothness. For each $s \in [0, 1]$, the dual Hölder inequality gives

$$|\langle \nabla F(\boldsymbol{X} + s\boldsymbol{D}) - \nabla F(\boldsymbol{X}), \boldsymbol{D} \rangle| \leq \|\nabla F(\boldsymbol{X} + s\boldsymbol{D}) - \nabla F(\boldsymbol{X})\|_* \, \|\boldsymbol{D}\|. \tag{55}$$

By the L-smoothness assumption in equation 49, the gradient deviation is bounded by

$$\|\nabla F(\boldsymbol{X} + s\boldsymbol{D}) - \nabla F(\boldsymbol{X})\|_* \leq L\|s\boldsymbol{D}\| = Ls\|\boldsymbol{D}\|. \tag{56}$$

Combining these two bounds yields

$$|\langle \nabla F(\boldsymbol{X} + s\boldsymbol{D}) - \nabla F(\boldsymbol{X}), \boldsymbol{D} \rangle| \leq Ls\|\boldsymbol{D}\|^2. \tag{57}$$

Applying the bound in equation 57 to the integral in equation 54, we obtain

$$F(\boldsymbol{X} + \boldsymbol{D}) - F(\boldsymbol{X}) \leq \langle \nabla F(\boldsymbol{X}), \boldsymbol{D} \rangle + \int_0^1 Ls\|\boldsymbol{D}\|^2 \, ds \tag{58}$$

$$= \langle \nabla F(\boldsymbol{X}), \boldsymbol{D} \rangle + \frac{L}{2}\|\boldsymbol{D}\|^2. \tag{59}$$

This proves the claimed descent inequality. □

C Related work

Optimizers can be categorized by the structural granularity they exploit.

Vector-aware optimizers. Vector-aware optimizers treat parameters as vectors, updating each coordinate independently based on per-coordinate gradient statistics. Adagrad (Duchi et al., 2011) introduced per-parameter scaling but suffers from aggressive learning-rate decay due to accumulating squared gradients. RMSprop (Hinton et al., 2012) addressed this by employing exponential moving averages, providing more stable updates. Adam (Kingma & Ba, 2015), combining momentum with adaptive scaling, has become the default optimizer in most deep learning pipelines, with AdamW (Loshchilov & Hutter, 2017) further improving generalization by decoupling weight decay from gradient updates.

Recent variants enhance efficiency and convergence through various strategies. Adan (Xie et al., 2024) and Win (Zhou et al., 2024; 2023) strengthen Adam with Nesterov-style acceleration for faster convergence. Lion (Chen et al., 2023) removes second-moment tracking entirely, trading some adaptability for significant memory savings. Sophia (Liu et al., 2023) leverages approximate second-order information (Hessian diagonal) to accelerate pre-training. Adam-mini (Zhang et al., 2024) reduces memory footprint via block-wise learning rates, partitioning parameters into blocks that share adaptive scales. Despite their practical success, all these methods fundamentally treat parameters as flattened vectors, updating each coordinate independently and disregarding the structural correlations inherent in matrix-valued or tensor-valued parameters such as weight matrices and convolutional kernels.

Matrix-aware optimizers. Recognizing the limitations of vector-wise updates, a growing body of work exploits matrix structure to improve optimization. K-FAC (Martens & Grosse, 2015) and Shampoo (Gupta et al., 2018) use Kronecker-factored approximations of the Fisher or curvature matrix, enabling efficient second-order updates that respect matrix geometry. Adafactor (Shazeer & Stern, 2018) reduces memory through factored second-moment estimates, while LAMB (You et al., 2019) introduces layer-wise normalization for stable large-batch training. More recently, GaLore (Zhao et al., 2024) projects gradients into low-rank subspaces to substantially reduce optimizer memory usage during pre-training. SOAP (Vyas et al., 2025) interprets Shampoo in the eigenbasis of its preconditioner and performs Adam-style updates in that basis, achieving improved performance over Adam at the cost of additional computational overhead. AdaDiag (Nguyen et al., 2025) leverages SVD to transform gradients into a near-diagonal preconditioning space, accelerating convergence through better-conditioned updates.

Among matrix-aware methods, Muon (Jordan et al., 2024) takes a distinctly different approach. Instead of constructing a preconditioner, Muon applies the polar factor—the unitary part of the polar decomposition—to each layer-wise momentum matrix via efficient Newton–Schulz iterations. This operation replaces all nonzero singular values with one, effectively regularizing the spectral distribution of update matrices and preventing rank collapse. The result is improved isotropy and training stability, particularly in large-scale settings where gradient rank collapse is a known issue. Subsequent work has scaled Muon to larger architectures (Liu et al., 2025a). Other recent advances further explore structural efficiency and scalability (Wang et al., 2024; Chen et al., 2024; Pethick et al., 2025; Liu et al., 2025b; An et al., 2025; Xie et al., 2025; Yu et al., 2024). Collectively, these matrix-aware methods demonstrate that structural awareness—particularly spectral operations at the

matrix level—can substantially enhance training efficiency, naturally motivating further exploration into higher-order tensor structures.

Tensor-aware optimizers. Recent advances in optimization for deep neural networks have sought to exploit the structured nature of parameters beyond the vectorized view. Tensorion Bogachev et al. (2026) extends Muon to higher-order parameter tensors, such as convolutional kernels, by replacing the matrix spectral norm with a designed tensor norm that balances tightness to the tensor spectral norm. Concretely, Tensorion defines a relaxed dual norm as the maximum of nuclear norms over a set of tensor unfoldings, and its primal norm is obtained via duality. The optimizer then selects a matricization mode that yields the most balanced unfolding, thereby maximizing a high-probability upper bound on the nuclear norm. Then it applies Newton–Schulz orthogonalization to the resulting matrix before folding it back.

Tensorion is primarily designed for convolutional neural networks and other vision-oriented architectures where parameters naturally appear as 4D tensors. Empirically, it demonstrates improved convergence and test accuracy over both SGD-based and Adam-based baselines on CIFAR and Tiny ImageNet datasets, as well as consistent gains when applied to the convolutional patch-embedding module in Vision Transformers. However, its reliance on a fixed unfolding per layer, while computationally efficient, may not fully adapt to dynamically changing gradient structures during training, and its validation is largely confined to computer vision tasks, with limited exploration in large-scale language modeling or sequential domains.

In contrast, TEON Zhang et al. (2026a) generalizes Muon across layers rather than within a single tensor. Motivated by the observation that layer-wise orthogonalization neglects cross-layer gradient correlations, TEON stacks gradient matrices of the Query, Key, and Value projections of Transformer blocks into a third-order tensor and applies mode-1 matricization followed by matrix orthogonalization. This choice is grounded in a theoretical analysis under the non-Euclidean trust-region framework, which shows that TEON can achieve up to a $\sqrt{K}$-fold improvement over Muon in convergence bounds when the stacked gradients exhibit aligned top right singular vectors. TEON is thus tailored for large language model pre-training, where parameter blocks are predominantly matrices but exhibit strong inter-layer and inter-component correlations. Experimental results on GPT-style and LLaMA-style models, ranging from 60M to 1B parameters, demonstrate consistent perplexity improvements over Muon and AdamW under various approximate SVD schemes. Nevertheless, $\sqrt{K}$-fold convergence improvement is theoretically appealing but practically limited, as its benefits are realized only at $K = 2$. Larger stacks fail to maintain the required singular-vector alignment and yield diminishing performance. More fundamentally, TEON applies a single global orthogonalization to the entire concatenated unfolding. The flat shape of a single matrix unfolded by all stacked layers may cause information loss. It inevitably suppresses weaker but potentially informative layer-specific signals when K grows or when layer-wise directions are not perfectly aligned.

The limitations of Tensorion and TEON reveal a common structural tension: Tensorion operates *intra-tensor* by selecting a single unfolding of a native higher-order parameter, while TEON operates *inter-tensor* by globally matricizing stacked layers, but both rely on a single, fixed matricization that collapses cross-mode or cross-layer information into a monolithic matrix before orthogonalization. This raises a natural question: can we design a tensor-aware optimizer that captures cross-layer structure without sacrificing per-layer expressivity, while avoiding the NP-hardness of the full tensor spectral norm?

We answer this affirmatively with **TUON**, a novel optimizer built upon transform-based tensor algebra. Instead of globally matricizing stacked layers, TUON applies a unitary transform $\boldsymbol{Q}$ along the layer mode, effectively rotating cross-layer information into a transform domain where each frontal slice becomes an independent $m \times n$ matrix representing a transformed cross-layer mode. Critically, TUON then applies the Muon polar map *independently to each transformed slice* before inverting the transform. This "mix-then-separate" strategy captures cross-layer correlations through the unitary mixing while preserving per-layer diversity through slice-wise polar maps, preventing the dominance of a single global direction that plagues TEON's global unfolding.

TUON is algebraically distinct from both predecessors. Unlike TEON, whose global polar map is invariant to unitary layer transforms, TUON's slice-wise polar map does *not* commute with the layer transform, making the choice of $\boldsymbol{Q}$ genuinely consequential and enabling the optimizer to adapt to the data-dependent structure of layer-wise gradients. Unlike Tensorion, TUON operates

in an exact LMO sense under a well-defined $\boldsymbol{Q}$ spectral norm, with dual norm equal to the sum of transformed-slice nuclear norms, eliminating the need for unfolding-mode search or spectral-norm relaxation. Theoretically, TUON enjoys a geometry-matched convergence guarantee and achieves up to a $\sqrt{K}$-fold improvement over TEON in both radius-matched and energy-matched oracle alignments, with the gain governed by a measurable alignment factor. Practically, the method adds only a lightweight unitary transform and retains the same Newton–Schulz iterations as Muon, making it scalable and easy to integrate into existing training pipelines. By combining the theoretical rigor of exact LMO geometry with the practical efficiency of transform-based tensor algebra, TUON offers a principled and generalizable solution to tensor-aware optimization that bridges the gap between intra-tensor structure and inter-layer correlations.

Tensor algebra. In the past years, two classical frameworks have dominated tensor decomposition related tensor algebra. CP decomposition factorizes a tensor as a sum of rank-one tensors; while conceptually straightforward, it is computationally demanding and its rank is NP-hard to compute (Kolda & Bader, 2009). Tucker decomposition, also known as Higher-Order Singular Value Decomposition (HOSVD), generalizes matrix SVD by decomposing a tensor into a core tensor multiplied by factor matrices along each mode (Tucker, 1966; De Lathauwer et al., 2000). These methods are grounded in a **matricization-based tensor algebra**, where tensors are first unfolded into matrices, and all subsequent algebraic operations are carried out at the matrix level. Although HOSVD successfully produces orthogonal bases per mode, the resulting core tensor is generally not f-diagonal (definitions can be found in Appendix B.5), losing the neat spectral interpretation of diagonal singular values in the matrix case. To remedy this, Kilmer & Martin (2011) introduced the t-product, a novel multiplication based on circular convolution and the Fast Fourier Transform (FFT). The t-product induces the tensor SVD (T-SVD):

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{U}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{S}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{V}}^{\top},$$

where $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{V}}$ are orthogonal, and $\boldsymbol{\mathcal{S}}$ is f-diagonal, thereby mirroring the matrix SVD structure. Computation proceeds by applying FFT along the third dimension, performing matrix SVD on each frontal slice, and inverting the transform. A major generalization is made by Kernfeld et al. (2015), who show that the t-product only depends on the chosen invertible linear transform and need not be limited to the Fourier transform. This leads to tensor-tensor products with arbitrary invertible transforms, such as the Discrete Cosine Transform (DCT), wavelets, or even data-dependent transforms, greatly expanding flexibility and applicability (Kernfeld et al., 2015; Lu et al., 2019). This yields a family of products and factorizations with algebraic structures (Kernfeld et al., 2015). These methods rely on the t-product, a tensor algebra defined natively on tensors rather than through unfolding. Matrix operations appear only as slice-wise operations in the transform domain. We refer to this paradigm as **transform-based tensor algebra**.

Subsequent works systematically investigate relations between T-SVD and HOSVD and established algebraic properties such as orthogonal invariance. Meanwhile, the tensor nuclear norm (TNN) and the tubal rank were defined based on T-SVD, providing powerful convex optimization tools for low-rank tensor recovery, including tensor completion and robust principal component analysis (Lu et al., 2018; 2019; Zheng et al., 2020). Algorithmically, the original T-SVD requires computing many matrix SVDs. Lu et al. (2018) reduce the cost by leveraging conjugate symmetry in the Fourier domain, requiring only about half the SVD computations. More recently, randomized T-SVD algorithms have been proposed for large-scale data via tensor random projections to obtain fast low-rank approximations. These tools have been successfully applied to tensor completion, robust principal component analysis, and multilinear systems (Lu et al., 2019; Zheng et al., 2020; Lu et al., 2018). Recent work extends these ideas to higher-order tensors via separable transforms along multiple modes (Zheng et al., 2020), and to data-dependent transforms that adapt the basis to the data at hand (Kong et al., 2021; Zheng et al., 2026). General tensor algebra has also been considered in low-rank optimization (Yang et al., 2026b;a; 2025). We specialize these tools to a full-parameter optimizer and develop comparative, convergence, and inexact-oracle analyses.

The definitions and theorems related to t-product and T-SVD can be found in the Preliminaries B.5 in Appendix.

D MUON AND TEON GEOMETRIES

This appendix fixes the baseline notation used in the main comparisons. The discussion concerns ideal exact polar maps; finite-iteration implementations may approximate them.

D.1 MUON

For a stack $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{m\times n\times K}$ with frontal slices $\boldsymbol{X}_k \in \mathbb{C}^{m\times n}$ for $k = 1, \dots, K$, we define the Muon norm pair as follows:

$$\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}} := \max_{1\le k\le K} \|\boldsymbol{X}_k\|_{\mathrm{op}}, \qquad \|\boldsymbol{\mathcal{G}}\|_{\mathrm{Muon},*} := \sum_{k=1}^{K} \|\boldsymbol{G}_k\|_*. \tag{60}$$

Here, $\boldsymbol{\mathcal{X}}$ denotes a generic tensor direction (used as the argument of the primal norm), while $\boldsymbol{\mathcal{G}}$ denotes a gradient tensor (used as the argument of the dual norm). The constraint $\|\boldsymbol{\mathcal{D}}\|_{\mathrm{Muon}} \le \rho$ means that each layer's update direction $\boldsymbol{D}_k$ has spectral norm at most ρ, independently of all other layers.

Proposition 4 (Ideal Muon as an exact LMO)**.** *Suppose each gradient slice has a compact singular value decomposition*

$$\boldsymbol{G}_k = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^{\mathsf{H}}, \qquad k = 1, \dots, K, \tag{61}$$

where $\boldsymbol{U}_k \in \mathbb{C}^{m\times r_k}$ and $\boldsymbol{V}_k \in \mathbb{C}^{n\times r_k}$ have orthonormal columns, and $\boldsymbol{\Sigma}_k = \mathrm{diag}(\sigma_{k,1}, \dots, \sigma_{k,r_k}) \in \mathbb{R}^{r_k\times r_k}$ is diagonal with positive singular values. Then an optimal solution of the linear minimization problem

$$\min_{\|\boldsymbol{\mathcal{D}}\|_{\mathrm{Muon}}\le\rho} \langle \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{D}}\rangle_{\mathrm{F}} \tag{62}$$

is given by the slices

$$\boldsymbol{D}_k^* = -\rho\, \boldsymbol{U}_k \boldsymbol{V}_k^{\mathsf{H}}, \qquad k = 1, \dots, K, \tag{63}$$

and the corresponding optimal value is

$$\min_{\|\boldsymbol{\mathcal{D}}\|_{\mathrm{Muon}}\le\rho} \langle \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{D}}\rangle_{\mathrm{F}} = -\rho \sum_{k=1}^{K} \|\boldsymbol{G}_k\|_*. \tag{64}$$

Proof. The trust-region constraint $\|\boldsymbol{\mathcal{D}}\|_{\mathrm{Muon}} \le \rho$ is, by definition, equivalent to the K independent constraints

$$\|\boldsymbol{D}_k\|_{\mathrm{op}} \le \rho \qquad \text{for all } k = 1, \dots, K. \tag{65}$$

The objective also decouples:

$$\langle \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{D}}\rangle_{\mathrm{F}} = \sum_{k=1}^{K} \langle \boldsymbol{G}_k, \boldsymbol{D}_k\rangle_{\mathrm{F}}. \tag{66}$$

Thus the overall optimization separates into K independent matrix subproblems, each of the form

$$\min_{\|\boldsymbol{D}_k\|_{\mathrm{op}}\le\rho} \langle \boldsymbol{G}_k, \boldsymbol{D}_k\rangle_{\mathrm{F}}. \tag{67}$$

By Lemma 3, the optimal solution to each subproblem is

$$\boldsymbol{D}_k^* = -\rho\ \mathrm{polar}(\boldsymbol{G}_k) = -\rho\, \boldsymbol{U}_k \boldsymbol{V}_k^{\mathsf{H}}, \tag{68}$$

with optimal value $-\rho\|\boldsymbol{G}_k\|_*$. Summing the K independent optimal values proves the claim. □

D.2 TEON

Without loss of generality, for analysis, let $\mathscr{U}(\boldsymbol{\mathcal{X}}) = [\boldsymbol{X}_1, \dots, \boldsymbol{X}_K]$. Define

$$\|\boldsymbol{\mathcal{X}}\|_{\mathrm{TEON}} := \|\mathscr{U}(\boldsymbol{\mathcal{X}})\|_{\mathrm{op}}, \qquad \|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*} := \|\mathscr{U}(\boldsymbol{\mathcal{G}})\|_*. \tag{69}$$

Unlike Muon, TEON couples all slices through one concatenated matrix constraint.

Proposition 5 (Ideal TEON as an exact LMO). *Let* $\mathscr{U}(\mathcal{G}) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{H}}$. *Then*

$$\mathcal{D}^*_{\mathrm{TEON}} = -\rho\mathscr{U}^{-1}(\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}) \tag{70}$$

solves

$$\min_{\|\mathcal{D}\|_{\mathrm{TEON}}\le\rho} \langle \mathcal{G}, \mathcal{D}\rangle_{\mathrm{F}}, \tag{71}$$

with optimal value $-\rho\|\mathscr{U}(\mathcal{G})\|_*$.

Proof. Matricization preserves the Frobenius inner product, so the tensor problem is exactly

$$\min_{\|\boldsymbol{D}_{(1)}\|_{\mathrm{op}}\le\rho} \langle \mathscr{U}(\mathcal{G}), \boldsymbol{D}_{(1)}\rangle_{\mathrm{F}}. \tag{72}$$

Lemma 3 gives $\boldsymbol{D}^*_{(1)} = -\rho\boldsymbol{U}\boldsymbol{V}^{\mathsf{H}}$. Folding this matrix back to a tensor gives the stated direction. □

D.3 How a layer transform acts on the TEON unfolding

We establish a key algebraic identity that underpins the comparison between TEON and our proposed method.

Let $\mathcal{X} \in \mathbb{C}^{m\times n\times K}$ be a third-order tensor with frontal slices $\boldsymbol{X}_1, \dots, \boldsymbol{X}_K$, and let $\mathscr{U}(\mathcal{X}) := [\boldsymbol{X}_1, \dots, \boldsymbol{X}_K] \in \mathbb{C}^{m\times nK}$ denote its horizontal unfolding. Under the mode-3 transform convention, applying a unitary matrix $\boldsymbol{Q} \in \mathbb{C}^{K\times K}$ along the layer dimension and then unfolding is equivalent to first unfolding and then multiplying on the right by a Kronecker product:

$$\mathscr{U}(\mathcal{X} \times_3 \boldsymbol{Q}) = \mathscr{U}(\mathcal{X})(\boldsymbol{Q}^{\mathsf{T}} \otimes \boldsymbol{I}_n). \tag{73}$$

The Kronecker product $\boldsymbol{R}_{\boldsymbol{Q}} := \boldsymbol{Q}^{\mathsf{T}} \otimes \boldsymbol{I}_n$ is unitary whenever $\boldsymbol{Q}$ is:

$$\boldsymbol{R}_{\boldsymbol{Q}}^{\mathsf{H}}\boldsymbol{R}_{\boldsymbol{Q}} = (\overline{\boldsymbol{Q}} \otimes \boldsymbol{I}_n)(\boldsymbol{Q}^{\mathsf{T}} \otimes \boldsymbol{I}_n) = (\overline{\boldsymbol{Q}}\boldsymbol{Q}^{\mathsf{T}}) \otimes \boldsymbol{I}_n = \boldsymbol{I}_{nK}. \tag{74}$$

This identity reveals a crucial distinction between TEON and TUON. The TEON norm is defined as the operator norm of the horizontal unfolding:

$$\|\mathcal{X}\|_{\mathrm{TEON}} := \|\mathscr{U}(\mathcal{X})\|_{\mathrm{op}}. \tag{75}$$

Since $\boldsymbol{R}_{\boldsymbol{Q}}$ is unitary, right-multiplication by $\boldsymbol{R}_{\boldsymbol{Q}}$ leaves the operator norm invariant:

$$\|\mathscr{U}(\mathcal{X} \times_3 \boldsymbol{Q})\|_{\mathrm{op}} = \|\mathscr{U}(\mathcal{X})\boldsymbol{R}_{\boldsymbol{Q}}\|_{\mathrm{op}} = \|\mathscr{U}(\mathcal{X})\|_{\mathrm{op}}. \tag{76}$$

Consequently, the TEON norm is invariant to the choice of the unitary layer transform $\boldsymbol{Q}$: the transform is entirely "absorbed" by the unitary invariance of the spectral norm.

This stands in sharp contrast to TUON, where the layer transform is applied *before* the nonlinear slice-wise polar map. Because the polar map does not commute with unitary transformations (as shown in Theorem 2), the TUON geometry depends nontrivially on $\boldsymbol{Q}$. The identity in equation 73 is therefore instrumental in the TEON comparison (Section 3.2) and the transform-cancellation proof (Theorem 2), where it formalizes why TEON cannot exploit cross-layer structure through a unitary layer basis.

E Selection of Q

E.1 Momentum tensor and layer transform

For one compatible group, stack the gradients and momenta as $\mathcal{G}_t(:,:,k) = \boldsymbol{G}_t^{(k)}$ and $\mathcal{M}_t = \beta\mathcal{M}_{t-1} + (1-\beta)\mathcal{G}_t$ where $\mathcal{M}_t \in \mathbb{R}^{m\times n\times K}$. Given a unitary $\boldsymbol{Q} \in \mathbb{C}^{K\times K}$, $\boldsymbol{Q}^{\mathsf{H}}\boldsymbol{Q} = I$, our tensor-product-transform treats every tube as a length-K column,

$$\widehat{\mathcal{M}}_t = \mathcal{M}_t \times_3 \boldsymbol{Q}, \quad \text{where the } j\text{-th slice of } \widehat{\mathcal{M}}_t \quad \widehat{\boldsymbol{M}}_{t,j} = \widehat{\mathcal{M}}_t(:,:,j) = \sum_{k=1}^{K} \boldsymbol{Q}_{jk}\boldsymbol{M}_t^{(k)}. \tag{77}$$

Before we study the choice of $\boldsymbol{Q}$, we establish a rigorous equivalence between tensor slice representation and matrix multiplication.

We use the sans-serif bold symbol M exclusively for the mode-3 unfolding matrix, reserving the italic bold $\boldsymbol{M}$ for individual slice matrices (e.g., $\boldsymbol{M}_t^{(k)}$).

First, we establish a rigorous equivalence between the slice-wise linear combination

$$\widehat{\boldsymbol{M}}_{t,j} = \sum_{k=1}^{K} Q_{jk}\, \boldsymbol{M}_t^{(k)}, \qquad j = 1, \ldots, K, \tag{78}$$

and the mode-3 matrix multiplication

$$\widehat{\mathsf{M}}_t = \mathbf{Q}\, \mathsf{M}_t, \tag{79}$$

where $\boldsymbol{M}_t^{(k)} \in \mathbb{R}^{m\times n}$ are the original slices, $Q_{jk} \in \mathbb{C}$ are the entries of $\mathbf{Q} \in \mathbb{C}^{K\times K}$, and $\widehat{\boldsymbol{M}}_{t,j} \in \mathbb{R}^{m\times n}$ are the resulting slices.

It can be proved as follows. Define the **vectorization** operator $\mathrm{vec} : \mathbb{R}^{m\times n} \to \mathbb{R}^{mn}$ that stacks the columns of a matrix into a single column vector. We construct the mode-3 unfolding matrix $\mathsf{M}_t \in \mathbb{R}^{K\times mn}$ by stacking the transposed vectorizations of each slice as rows:

$$\mathsf{M}_t = \begin{bmatrix} \mathrm{vec}(\boldsymbol{M}_t^{(1)})^\mathsf{T} \\ \mathrm{vec}(\boldsymbol{M}_t^{(2)})^\mathsf{T} \\ \vdots \\ \mathrm{vec}(\boldsymbol{M}_t^{(K)})^\mathsf{T} \end{bmatrix}. \tag{80}$$

Equivalently, the k-th row of M_t satisfies

$$(\mathsf{M}_t)_{k,:} = \mathrm{vec}(\boldsymbol{M}_t^{(k)})^\mathsf{T}. \tag{81}$$

Then we consider the matrix multiplication. Let $\mathbf{Q} \in \mathbb{C}^{K\times K}$. Compute the matrix product

$$\widehat{\mathsf{M}}_t = \mathbf{Q}\, \mathsf{M}_t \in \mathbb{C}^{K\times mn}. \tag{82}$$

By the definition of matrix multiplication, the j-th row of $\widehat{\mathsf{M}}_t$ is

$$(\widehat{\mathsf{M}}_t)_{j,:} = \sum_{k=1}^{K} Q_{jk}\, (\mathsf{M}_t)_{k,:}. \tag{83}$$

Next, we apply the inverse vectorization operator $\mathrm{vec}^{-1} : \mathbb{R}^{mn} \to \mathbb{R}^{m\times n}$ to each row of $\widehat{\mathsf{M}}_t$. By linearity of vec^{-1},

$$\mathrm{vec}^{-1}\Big((\widehat{\mathsf{M}}_t)_{j,:}^\mathsf{T}\Big) = \mathrm{vec}^{-1}\left(\sum_{k=1}^{K} Q_{jk}\, (\mathsf{M}_t)_{k,:}^\mathsf{T}\right) = \sum_{k=1}^{K} Q_{jk}\ \mathrm{vec}^{-1}\big((\mathsf{M}_t)_{k,:}^\mathsf{T}\big). \tag{84}$$

By construction of M_t in equation 80,

$$\mathrm{vec}^{-1}\big((\mathsf{M}_t)_{k,:}^\mathsf{T}\big) = \boldsymbol{M}_t^{(k)}. \tag{85}$$

Therefore, defining

$$\widehat{\boldsymbol{M}}_{t,j} := \mathrm{vec}^{-1}\Big((\widehat{\mathsf{M}}_t)_{j,:}^\mathsf{T}\Big), \tag{86}$$

we obtain

$$\widehat{\boldsymbol{M}}_{t,j} = \sum_{k=1}^{K} Q_{jk}\, \boldsymbol{M}_t^{(k)}. \tag{87}$$

As a conclusion, equation 87 is identical to equation 78. Hence,

$$\widehat{\boldsymbol{M}}_{t,j} = \sum_{k=1}^{K} Q_{jk} \boldsymbol{M}_t^{(k)} \iff \widehat{\mathsf{M}}_t = \mathbf{Q}\mathsf{M}_t \tag{88}$$

where the equivalence holds up to the bijective mapping between slices and rows established by vec and vec^{-1}. Consequently, the slice-wise linear combination is precisely the row-wise linear combination effected by left-multiplication by $\mathbf{Q}$ on the mode-3 unfolding.

The equivalence above justifies interpreting $\times_3$ as left-multiplication on the mode-3 unfolding. Under above formulations, we give several concrete choices of $\boldsymbol{Q}$ and analyze the default DecorQ method we choose .

DCT. For the orthonormal DCT-II,

$$\boldsymbol{Q}_{jk} = \alpha_j \cos\left[\frac{\pi}{K}\left(k+\frac{1}{2}\right)j\right], \quad \alpha_0 = K^{-1/2}, \quad \alpha_j = (2/K)^{1/2}\ (j>0),$$

with zero-based indices. It is real and expresses smooth and oscillatory depth modes without imposing a periodic endpoint.

The main practical transform is as follow. For $K = 2$,

$$\widehat{\boldsymbol{M}}_{t,1} = \frac{\boldsymbol{M}_t^{(1)} + \boldsymbol{M}_t^{(2)}}{\sqrt{2}}, \qquad \widehat{\boldsymbol{M}}_{t,2} = \frac{\boldsymbol{M}_t^{(1)} - \boldsymbol{M}_t^{(2)}}{\sqrt{2}}. \tag{89}$$

The first slice is a shared adjacent-layer mode and the second is a residual mode.

Haar transform. The Haar transform is the simplest orthogonal wavelet transform. For a vector of length $N = 2^n$, it recursively decomposes adjacent pairs of entries into their sum and difference, producing a multi-scale representation: the leading coefficients encode global averages, while the remaining coefficients encode local differences at increasingly finer scales.

Formally, the Haar matrix H_N is defined recursively as

$$H_1 = [1], \qquad H_{2n} = \frac{1}{\sqrt{2}}\begin{bmatrix} H_n \otimes [1,\ 1] \\ I_n \otimes [1,\ -1] \end{bmatrix}, \tag{90}$$

where $\otimes$ denotes the Kronecker product and I_n is the $n \times n$ identity matrix. For example,

$$H_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad H_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}. \tag{91}$$

The Haar matrix is orthogonal, i.e., $H_N H_N^\top = H_N^\top H_N = I$, and can be applied in $O(N)$ time via a butterfly scheme. Unlike the DCT, whose basis functions are global cosine waves, each Haar basis function is supported on a small number of adjacent coordinates. It is therefore well suited to capturing local, piecewise, or multi-scale correlation structures, while requiring only additions, subtractions, and powers of $\sqrt{2}$, with no trigonometric evaluations.

DFT/FFT. The unitary DFT uses $\boldsymbol{Q}_{jk} = K^{-1/2} e^{-2\pi i jk/K}$. It is implemented with an orthonormal FFT. Frontal slices and NS products are complex-valued, with Hermitian rather than ordinary transposes.

DecorQ. Let $\mathsf{M}_t \in \mathbb{R}^{K \times mn}$ be the mode-3 unfolding of the momentum tensor $\boldsymbol{\mathcal{M}}_t$, whose k-th row is $\mathrm{vec}(\boldsymbol{M}_t^{(k)})^\mathsf{T}$. We set $\boldsymbol{Q} = U_3^\mathsf{T}$, where U_3 is the left singular matrix from the SVD $\mathsf{M}_t = U_3 \Sigma_3 V_3^\mathsf{T}$. This choice is not arbitrary: it diagonalizes the layer-wise Gram matrix, inducing a geometrically decoupled coordinate system in which subsequent operations act independently on each slice.

To see this, consider the Gram matrix of the original slices:

$$\mathsf{C}_t := \mathsf{M}_t \mathsf{M}_t^\mathsf{T} \in \mathbb{R}^{K \times K}, \qquad (\mathsf{C}_t)_{ij} = \langle \boldsymbol{M}_t^{(i)}, \boldsymbol{M}_t^{(j)} \rangle_{\mathrm{F}}, \tag{92}$$

whose (i, j)-th entry measures the Frobenius inner product between the i-th and j-th momentum slices. This matrix encodes the **pairwise coupling** among layers, that is large off-diagonal entries indicate strongly correlated update directions, while near-zero entries indicate decoupled evolution.

The goal of $\boldsymbol{Q}$ is to **diagonalize** this coupling structure via a unitary rotation, so that each transformed slice evolves independently in its own orthogonal subspace.

Under the transform $\widehat{\mathsf{M}}_t = \boldsymbol{Q}\mathsf{M}_t$, the Gram matrix of the transformed slices becomes

$$\widehat{\mathsf{C}}_t = \widehat{\mathsf{M}}_t\widehat{\mathsf{M}}_t^{\mathsf{T}} = \boldsymbol{Q}\mathsf{M}_t\mathsf{M}_t^{\mathsf{T}}\boldsymbol{Q}^{\mathsf{T}} = U_3^{\mathsf{T}}\big(U_3\Sigma_3^2U_3^{\mathsf{T}}\big)U_3 = \Sigma_3^2, \tag{93}$$

which is diagonal. Hence, for any $i \neq j$,

$$\langle \widehat{\boldsymbol{M}}_{t,i}, \widehat{\boldsymbol{M}}_{t,j}\rangle_{\mathrm{F}} = (\widehat{\mathsf{C}}_t)_{ij} = 0. \tag{94}$$

The transformed slices $\widehat{\boldsymbol{M}}_{t,1}, \ldots, \widehat{\boldsymbol{M}}_{t,K}$ are therefore mutually orthogonal in the Frobenius inner product sense.

This orthogonality has a fundamental algorithmic consequence. The polarization step computes $\widehat{\boldsymbol{P}}_{t,j} = \mathrm{polar}(\widehat{\boldsymbol{M}}_{t,j})$, which preserves the column and row subspaces of each slice while discarding its magnitude. Since the slices are mutually orthogonal, their polar factors remain confined to their respective orthogonal subspaces:

$$\langle \widehat{\boldsymbol{P}}_{t,i}, \widehat{\boldsymbol{P}}_{t,j}\rangle_{\mathrm{F}} = 0, \quad i \neq j. \tag{95}$$

Because the inverse transform $\boldsymbol{\mathcal{P}}_t = \widehat{\boldsymbol{\mathcal{P}}}_t \times_3 \boldsymbol{Q}^{\mathsf{H}}$ is a unitary rotation (and thus preserves inner products), this orthogonality carries over to the original tensor domain:

$$\langle \boldsymbol{\mathcal{P}}_t(:,:,i), \boldsymbol{\mathcal{P}}_t(:,:,j)\rangle_{\mathrm{F}} = 0, \quad i \neq j. \tag{96}$$

Consequently, the update direction for slice i resides entirely in a subspace orthogonal to that of slice j. Any modification to slice i's update—whether scaling, clipping, or otherwise adjusting its magnitude—produces no projection onto slice j's subspace. In other words, adjusting slice i does not affect slice j. This geometric decoupling effectively transforms the original coupled multi-layer optimization problem into K independent single-channel subproblems, each evolving in its own orthogonal subspace without mutual interference.

F Polarization via Newton–Schulz iteration

We now describe the practical polarization step that replaces the expensive singular value decomposition. Instead of computing the exact polar factor via SVD, we use a fast iterative approximation based on the Newton–Schulz polynomial.

For each transformed slice $\widehat{\boldsymbol{M}}_{t,j} \in \mathbb{R}^{m\times n}$, we first ensure that the matrix is in a shape suitable for efficient iteration. If $m > n$, we transpose the slice so that the number of rows does not exceed the number of columns; this keeps the Gram matrix $\boldsymbol{X}\boldsymbol{X}^{\mathsf{T}}$ as small as possible and reduces computational cost. After optionally transposing, we initialize

$$\boldsymbol{X}_{0,j} = \frac{\widehat{\boldsymbol{M}}_{t,j}}{\|\widehat{\boldsymbol{M}}_{t,j}\|_{\mathrm{F}} + \epsilon}, \tag{97}$$

where $\epsilon > 0$ is a small constant to prevent division by zero. This initialization scales the slice to unit Frobenius norm, which is a stable starting point for the iteration.

We then apply J steps of the quintic Newton–Schulz iteration. At each step $\ell = 0, 1, \ldots, J-1$, we compute

$$\boldsymbol{A}_{\ell,j} = \boldsymbol{X}_{\ell,j}\boldsymbol{X}_{\ell,j}^{\mathsf{H}}, \tag{98}$$

and update

$$\boldsymbol{X}_{\ell+1,j} = a\boldsymbol{X}_{\ell,j} + \big(b\boldsymbol{A}_{\ell,j} + c\boldsymbol{A}_{\ell,j}^2\big)\,\boldsymbol{X}_{\ell,j}, \tag{99}$$

where $(a, b, c) = (3.4445, -4.7750, 2.0315)$ are the precomputed coefficients from the PolarExpress scheme (Amsel et al., 2026). These coefficients are designed to flatten the singular-value spectrum rapidly: the iteration drives all nonzero singular values toward one within a small number of steps. The output $\boldsymbol{X}_{J,j}$ is an approximation to the exact polar factor $\mathrm{polar}(\widehat{\boldsymbol{M}}_{t,j})$, though it is not exact due to the finite iteration count and floating-point rounding (Kim & Oh, 2026). In practice, $J = 4$ or 5 suffices for most applications.

G STACK SCALING AND WEIGHT UPDATE

After obtaining the approximate polar factors $\boldsymbol{\mathcal{P}}_t$ in the original tensor domain via inverse transformation, we apply a single shared scaling factor to the entire stack. This step controls the overall step magnitude while preserving the direction of the update.

Let $\mathrm{RMS}(\boldsymbol{\mathcal{P}}_t)$ denote the root-mean-square value of all entries in the tensor $\boldsymbol{\mathcal{P}}_t$:

$$\mathrm{RMS}(\boldsymbol{\mathcal{P}}_t) = \sqrt{\frac{1}{mnK} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{K} |\boldsymbol{\mathcal{P}}_t(i,j,k)|^2}. \tag{100}$$

We define the scaling factor

$$\gamma_t = \frac{s}{\mathrm{RMS}(\boldsymbol{\mathcal{P}}_t) + \epsilon}, \tag{101}$$

where $s > 0$ is a target RMS value (typically set to a constant such as $1/\sqrt{\max(m,n)}$ as suggested by the rank–energy identity in Proposition 3), and ϵ is a small constant to avoid division by zero. The scaled update is then

$$\widetilde{\boldsymbol{\mathcal{P}}}_t = \gamma_t \, \boldsymbol{\mathcal{P}}_t. \tag{102}$$

Because γ_t is shared across all layers, this scaling changes the effective trust-region radius but does not alter the relative direction of the update across slices. This is a crucial design choice: it preserves the geometric structure of the polar direction while allowing the overall step size to be tuned by a single global parameter.

Finally, we apply decoupled weight decay and the scaled update to obtain the new parameters:

$$\boldsymbol{\mathcal{W}}_{t+1} = (1 - \eta_t \lambda) \, \boldsymbol{\mathcal{W}}_t - \eta_t \, \widetilde{\boldsymbol{\mathcal{P}}}_t, \tag{103}$$

where η_t is the learning rate and λ is the weight-decay coefficient. This update rule combines the momentum-based polar direction with standard L_2 regularization.

We note that one could also apply per-slice scaling factors, for example to preserve the Frobenius energy of each individual slice. Such variants can be useful in certain engineering contexts, but they correspond to a different weighted geometry and are not part of the main algorithm presented here. We therefore treat them as separate ablations and focus on the shared-scaling scheme throughout this work.

G.1 COMPUTATIONAL COMPLEXITY ANALYSIS

We analyze the per-iteration computational cost of the proposed optimizer for a single tensor group consisting of K matrices of size $m \times n$. The main computational burden lies in the slice-wise Newton–Schulz polar iterations. For each of the K slices, each iteration involves matrix multiplications of an $m \times n$ matrix with its transpose, incurring $O(mn \cdot \min(m,n))$ operations. With a fixed number of iterations J (typically $J = 5$ in our implementation), the total cost of polarization is $O(JKmn \cdot \min(m,n))$, which is $O(Kmn \cdot \min(m,n))$ since J is a constant. In addition to polarization, the tensor transform $\boldsymbol{\mathcal{M}}_t \times_3 \boldsymbol{Q}$ and its inverse each require $O(K^2 mn)$ operations. When $\boldsymbol{Q}$ is updated adaptively from the momentum tensor, an additional SVD on the $K \times mn$ unfolding matrix is performed every I (e.g., 100) steps, costing $O(K^2 mn)$ per update (assuming $K \ll mn$), which amortizes to $O(K^2 mn/I)$ per iteration. Thus the total complexity per iteration is approximately $K\,mn \cdot \min(m,n) + K^2 mn$. Since $K \ll m$ and $K \ll n$, the polarization step therefore dominates the overall cost, and the total complexity per iteration is approximately $O(K\,mn \cdot \min(m,n))$.

Muon incurs $O(Kmn \cdot \min(m,n))$ cost, while TEON requires $O(m(nK) \cdot \min(m, nK))$. Our method has a cost of $O(Kmn \cdot \min(m,n))$. In the typical regime where $K \ll \min(m,n)$, these complexities are comparable. Moreover, this overhead remains negligible relative to the overall pretraining cost. Thus, TUON achieves its cross-layer structural benefits at a modest computational premium, scaling gracefully with the group size K. Table 1 compares the resulting complexities.

H Detailed proofs for Section 3

H.1 Proof of Proposition 1: duality of the Q-block norms

Before solving a norm-constrained steepest-descent problem, we must know the dual norm. It measures the largest possible first-order alignment with a direction in the unit trust region. Proposition 1 identifies the measure naturally associated with TUON.

Proposition 1[Dual norm] *For unitary $\boldsymbol{Q}$, $\|\cdot\|_{\boldsymbol{Q},\mathrm{Tn}}$ is the dual norm of $\|\cdot\|_{\boldsymbol{Q},\mathrm{Ts}}$:*

$$\sup_{\|\boldsymbol{\mathcal{D}}\|_{\boldsymbol{Q},\mathrm{Ts}}\le 1} \langle \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{D}}\rangle_{\mathrm{F}} = \|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}}. \tag{104}$$

Proof. We prove the duality by establishing both the upper bound and its attainability. The argument proceeds in three steps.

First, we move the inner product to the transform domain. By the Parseval identity for unitary mode-$\boldsymbol{Q}$ transforms (Lemma 1), the Frobenius inner product is preserved under the layer-mode transformation:

$$\langle \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{D}}\rangle_{\mathrm{F}} = \langle \widehat{\boldsymbol{\mathcal{G}}}, \widehat{\boldsymbol{\mathcal{D}}}\rangle_{\mathrm{F}} = \sum_{j=1}^{K} \langle \widehat{\boldsymbol{G}}_j, \widehat{\boldsymbol{D}}_j\rangle_{\mathrm{F}}, \tag{105}$$

where $\widehat{\boldsymbol{\mathcal{G}}} = \boldsymbol{\mathcal{G}} \times_3 \boldsymbol{Q}$ and $\widehat{\boldsymbol{\mathcal{D}}} = \boldsymbol{\mathcal{D}} \times_3 \boldsymbol{Q}$ denote the transformed tensors, and $\widehat{\boldsymbol{G}}_j, \widehat{\boldsymbol{D}}_j \in \mathbb{C}^{m\times n}$ are their respective j-th frontal slices. Thus the tensor inner product decouples into a sum of K independent matrix inner products.

Second, we have the universal upper bound. For any feasible direction satisfying $\|\boldsymbol{\mathcal{D}}\|_{\boldsymbol{Q},\mathrm{Ts}} \le 1$, the definition of the tensor spectral norm under $\boldsymbol{Q}$-transform implies

$$\|\widehat{\boldsymbol{D}}_j\|_{\mathrm{op}} \le 1 \qquad \text{for all } j = 1, \dots, K. \tag{106}$$

Applying the matrix Hölder inequality (Lemma 3) slice-wise yields

$$\langle \widehat{\boldsymbol{G}}_j, \widehat{\boldsymbol{D}}_j\rangle_{\mathrm{F}} \le \|\widehat{\boldsymbol{G}}_j\|_* \|\widehat{\boldsymbol{D}}_j\|_{\mathrm{op}} \le \|\widehat{\boldsymbol{G}}_j\|_*. \tag{107}$$

Summing this inequality over all slices gives

$$\langle \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{D}}\rangle_{\mathrm{F}} \le \sum_{j=1}^{K} \|\widehat{\boldsymbol{G}}_j\|_* = \|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}}. \tag{108}$$

Since this bound holds for every feasible $\boldsymbol{\mathcal{D}}$, we conclude that the supremum is at most $\|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}}$.

At last, we construct the equality case to show that the upper bound is tight. For each nonzero transformed slice, take its compact singular value decomposition

$$\widehat{\boldsymbol{G}}_j = \boldsymbol{U}_j \boldsymbol{\Sigma}_j \boldsymbol{V}_j^{\mathsf{H}}, \tag{109}$$

and define its canonical polar factor

$$\widehat{\boldsymbol{D}}_j^* = \boldsymbol{U}_j \boldsymbol{V}_j^{\mathsf{H}} = \mathrm{polar}(\widehat{\boldsymbol{G}}_j). \tag{110}$$

For zero slices, set $\widehat{\boldsymbol{D}}_j^* = \boldsymbol{0}$. Lemma 2 guarantees that each polar factor satisfies $\|\widehat{\boldsymbol{D}}_j^*\|_{\mathrm{op}} \le 1$, so the assembled tensor $\widehat{\boldsymbol{\mathcal{D}}}^*$ is feasible under the tensor spectral norm under $\boldsymbol{Q}$-transform. Moreover, the polar alignment identity gives

$$\langle \widehat{\boldsymbol{G}}_j, \widehat{\boldsymbol{D}}_j^*\rangle_{\mathrm{F}} = \|\widehat{\boldsymbol{G}}_j\|_*. \tag{111}$$

Summing over j and applying the inverse unitary transform (which preserves the inner product) yields equality in equation 108. Therefore the upper bound is attained, and the supremum is exactly $\|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}}$. □

By applying the transform $\boldsymbol{Q}$, the original tensor norm decouples into independent matrix norms across the K slices. Therefore, the natural measure of the gradient's descent potential is no longer its Frobenius norm, but the sum of nuclear norms of the transformed slices. This decoupling serves as the basic tool behind the exact oracle, the comparison with Muon and TEON, and the convergence.

H.2 Proof of Theorem 1: exact normalized steepest descent

Theorem 1 [Exact LMO and normalized steepest descent] *For every radius $\rho > 0$,*

$$-\rho \boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}}) \in \underset{\|\boldsymbol{\mathcal{D}}\|_{\boldsymbol{Q},\mathrm{Ts}} \le \rho}{\arg\min} \langle \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{D}} \rangle_{\mathrm{F}}, \tag{112}$$

and the optimal value is $-\rho\|\boldsymbol{\mathcal{M}}\|_{\boldsymbol{Q},\mathrm{Tn}} = -\rho \sum_{j=1}^{K} \|\widehat{\boldsymbol{M}}_j\|_*$.

Proof. For any $\boldsymbol{\mathcal{D}}$ with $\|\boldsymbol{\mathcal{D}}\|_{\boldsymbol{Q},\mathrm{Ts}} \le \rho$, the dual Hölder inequality from Proposition 1 gives

$$|\langle \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{D}} \rangle_{\mathrm{F}}| \le \|\boldsymbol{\mathcal{M}}\|_{\boldsymbol{Q},\mathrm{Tn}} \|\boldsymbol{\mathcal{D}}\|_{\boldsymbol{Q},\mathrm{Ts}} \le \rho \|\boldsymbol{\mathcal{M}}\|_{\boldsymbol{Q},\mathrm{Tn}}. \tag{113}$$

Hence

$$\langle \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{D}} \rangle_{\mathrm{F}} \ge -\rho \|\boldsymbol{\mathcal{M}}\|_{\boldsymbol{Q},\mathrm{Tn}}. \tag{114}$$

No feasible direction can have a smaller linearized objective value.

In transformed coordinates, set

$$\widehat{\boldsymbol{D}}_j^* := -\rho \,\mathrm{polar}(\widehat{\boldsymbol{M}}_j). \tag{115}$$

Each nonzero polar factor has spectral norm one, so

$$\|\boldsymbol{\mathcal{D}}^*\|_{\boldsymbol{Q},\mathrm{Ts}} = \max_j \|\widehat{\boldsymbol{D}}_j^*\|_{\mathrm{op}} \le \rho. \tag{116}$$

Inverse transforming the slices gives exactly

$$\boldsymbol{\mathcal{D}}^* = -\rho \boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}}). \tag{117}$$

Using Parseval and Lemma 2,

$$\langle \boldsymbol{\mathcal{M}}, \boldsymbol{\mathcal{D}}^* \rangle_{\mathrm{F}} = -\rho \sum_{j=1}^{K} \langle \widehat{\boldsymbol{M}}_j, \mathrm{polar}(\widehat{\boldsymbol{M}}_j) \rangle_{\mathrm{F}} \tag{118}$$

$$= -\rho \sum_{j=1}^{K} \|\widehat{\boldsymbol{M}}_j\|_* \tag{119}$$

$$= -\rho \|\boldsymbol{\mathcal{M}}\|_{\boldsymbol{Q},\mathrm{Tn}}. \tag{120}$$

This equals the universal lower bound in equation 114, so the proposed direction is globally optimal. □

H.3 Proof of Proposition 2: tensor-polar equivalence

The update of our method also admits a natural interpretation in the language of tensor algebra. Let

$$\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{U}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{S}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{V}}^\top \tag{121}$$

be the T-SVD (Kernfeld et al., 2015; Kilmer & Martin, 2011; Zheng et al., 2026) of $\boldsymbol{\mathcal{M}}$ (see Appendix B.5 for definitions), where $\boldsymbol{\mathcal{S}}$ is an $\boldsymbol{f}$-diagonal tensor whose diagonal entries in the transform domain are the nonzero singular values of the corresponding transformed slices. Define $\boldsymbol{\mathcal{J}}_{\mathrm{supp}}$ to be the $\boldsymbol{f}$-diagonal tensor obtained from $\boldsymbol{\mathcal{S}}$ by replacing every nonzero diagonal entry with 1, while leaving zeros unchanged. In other words, $\boldsymbol{\mathcal{J}}_{\mathrm{supp}}$ is the indicator of the support of the singular values in the transform domain. The polar factor of a matrix $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is $\boldsymbol{U}\boldsymbol{V}^\top$, which replaces the singular values $\boldsymbol{\Sigma}$ by the identity on the support. Analogously, the generalized tensor polar factor of $\boldsymbol{\mathcal{M}}$ is obtained by replacing the singular-value tensor $\boldsymbol{\mathcal{S}}$ with its support indicator $\boldsymbol{\mathcal{J}}_{\mathrm{supp}}$:

$$\text{t-polar}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}}) := \boldsymbol{\mathcal{U}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{J}}_{\mathrm{supp}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{V}}^\top. \tag{122}$$

The following proposition establishes that this tensor-algebra object is exactly the optimizer map from Theorem 1.

Proposition 2[$\boldsymbol{Q}$-tensor polar equivalence] *The optimizer map equals the partial-isometry factor of the generalized T-SVD:*

$$\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}}) = \mathrm{tpolar}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}}) = \boldsymbol{\mathcal{U}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{J}}_{\mathrm{supp}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{V}}^\top. \tag{123}$$

For $K = 1$, this reduces to matrix Muon. For $\boldsymbol{Q} = \boldsymbol{I}$, it reduces to independent layer-wise polar maps.

Proof. We prove the equivalence by working in the transform domain, where the Tensor SVD decouples into independent matrix SVDs.

First, we make slice-wise decomposition in the transform domain. By the definition of the generalized T-SVD in Appendix B.5, equation 121 is equivalent to a collection of K independent matrix singular value decompositions applied to each transformed slice:

$$\widehat{\boldsymbol{M}}_j = \widehat{\boldsymbol{U}}_j \widehat{\boldsymbol{S}}_j \widehat{\boldsymbol{V}}_j^{\mathsf{H}}, \qquad j = 1, \dots, K, \tag{124}$$

where $\widehat{\boldsymbol{S}}_j = \operatorname{diag}(\sigma_{j,1}, \dots, \sigma_{j,r_j}) \in \mathbb{R}^{r_j \times r_j}$ is a diagonal matrix containing the nonzero singular values of the j-th transformed slice, and $r_j = \operatorname{rank}(\widehat{\boldsymbol{M}}_j)$.

Then we replace singular values by their support indicators. For each j, define $\widehat{\boldsymbol{J}}_{\mathrm{supp},j} \in \mathbb{R}^{r_j \times r_j}$ to be the diagonal matrix

$$\widehat{\boldsymbol{J}}_{\mathrm{supp},j} := \operatorname{diag}(\mathbf{1}_{\{\sigma_{j,1}>0\}}, \dots, \mathbf{1}_{\{\sigma_{j,r_j}>0\}}) = \boldsymbol{I}_{r_j}, \tag{125}$$

which is simply the identity matrix of size r_j, since all diagonal entries of $\widehat{\boldsymbol{S}}_j$ are positive by construction. In the full $m \times n$ matrix, this corresponds to a diagonal matrix with ones on the support of the nonzero singular values and zeros elsewhere. Then

$$\widehat{\boldsymbol{U}}_j \widehat{\boldsymbol{J}}_{\mathrm{supp},j} \widehat{\boldsymbol{V}}_j^{\mathsf{H}} = \widehat{\boldsymbol{U}}_j \widehat{\boldsymbol{V}}_j^{\mathsf{H}} = \operatorname{polar}(\widehat{\boldsymbol{M}}_j). \tag{126}$$

This is precisely the canonical polar factor of the j-th transformed slice: it retains the left and right singular vectors while discarding the singular values.

Thus, by reassembling the tensor, the transformed frontal slices of the tensor

$$\boldsymbol{\mathcal{U}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{J}}_{\mathrm{supp}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{V}}^{\top} \tag{127}$$

are exactly the slice-wise polar factors constructed in Step 2. Applying the inverse transform $\boldsymbol{Q}^{\mathsf{H}}$ along the third mode reconstructs the original tensor domain. By the definition of $\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}})$ in Section 3.1, this reconstructed tensor is precisely $\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}})$. Therefore,

$$\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}}) = \boldsymbol{\mathcal{U}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{J}}_{\mathrm{supp}} *_{\boldsymbol{Q}} \boldsymbol{\mathcal{V}}^{\top}, \tag{128}$$

which proves the proposition. □

H.4 Proof of Theorem 2: cancellation and non-cancellation

Theorem 2[unfolding cancellation versus slice-wise non-cancellation] *Let $\boldsymbol{M}_{(1)} = \mathscr{U}(\boldsymbol{\mathcal{M}})$ and $\boldsymbol{R}_{\boldsymbol{Q}} = \boldsymbol{Q}^{\mathsf{T}} \otimes \boldsymbol{I}_n$. For the canonical matrix polar map,*

$$\operatorname{polar}(\boldsymbol{M}_{(1)} \boldsymbol{R}_{\boldsymbol{Q}}) \boldsymbol{R}_{\boldsymbol{Q}}^{\mathsf{H}} = \operatorname{polar}(\boldsymbol{M}_{(1)}). \tag{129}$$

Hence a unitary layer transform is algebraically redundant around a single global unfolding polar. In contrast,

$$\big(\operatorname{polar}_{\mathrm{slice}}(\boldsymbol{\mathcal{M}} \times_3 \boldsymbol{Q})\big) \times_3 \boldsymbol{Q}^{\mathsf{H}} \neq \operatorname{polar}_{\mathrm{slice}}(\boldsymbol{\mathcal{M}}) \tag{130}$$

in general.

Proof. We prove the two claims separately. First, we will show that unfolding polar cancels the transform. Let

$$\boldsymbol{M}_{(1)} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^{\mathsf{H}} \tag{131}$$

be a compact singular value decomposition of the unfolded matrix. Because $\boldsymbol{R}_{\boldsymbol{Q}}$ is unitary, we have

$$\boldsymbol{M}_{(1)} \boldsymbol{R}_{\boldsymbol{Q}} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^{\mathsf{H}} \boldsymbol{R}_{\boldsymbol{Q}} = \boldsymbol{U} \boldsymbol{\Sigma} (\boldsymbol{R}_{\boldsymbol{Q}}^{\mathsf{H}} \boldsymbol{V})^{\mathsf{H}}. \tag{132}$$

The columns of $\boldsymbol{R}_{\boldsymbol{Q}}^{\mathsf{H}} \boldsymbol{V}$ remain orthonormal, so this is another compact SVD with the same left singular vectors $\boldsymbol{U}$ and the same singular values $\boldsymbol{\Sigma}$. Therefore,

$$\operatorname{polar}(\boldsymbol{M}_{(1)} \boldsymbol{R}_{\boldsymbol{Q}}) = \boldsymbol{U} (\boldsymbol{R}_{\boldsymbol{Q}}^{\mathsf{H}} \boldsymbol{V})^{\mathsf{H}} = \boldsymbol{U} \boldsymbol{V}^{\mathsf{H}} \boldsymbol{R}_{\boldsymbol{Q}} = \operatorname{polar}(\boldsymbol{M}_{(1)}) \boldsymbol{R}_{\boldsymbol{Q}}. \tag{133}$$

Multiplying both sides on the right by $\boldsymbol{R}_{\boldsymbol{Q}}^{\mathsf{H}}$ proves equation 129.

Second, we give a concrete slice-wise counterexample. It suffices to construct this single instance where equality fails. Consider two scalar slices

$$\boldsymbol{M}_1 = 2, \qquad \boldsymbol{M}_2 = 1, \tag{134}$$

and the two-point orthonormal DCT/Hadamard matrix

$$\boldsymbol{Q} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{135}$$

If we apply scalar polar maps directly to the original slices, we obtain $(1, 1)^\mathsf{T}$, since the polar factor of any positive scalar is 1.

In the transformed path, we first mix the slices:

$$\boldsymbol{Q} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 3 \\ 1 \end{bmatrix}. \tag{136}$$

Both entries are positive, so their scalar polar factors are again $(1, 1)^\mathsf{T}$. Applying the inverse transform gives

$$\boldsymbol{Q}^\mathsf{T} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix}, \tag{137}$$

which differs from the original $(1, 1)^\mathsf{T}$. Therefore, slice-wise polarization and layer mixing do not commute in general. □

H.5 Proof of Proposition 3: exact update energy

Proposition 3[Rank–energy identity] *Let* $r_j := \operatorname{rank}(\widehat{\boldsymbol{M}}_j)$. *Then*

$$\|\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}})\|_{\mathrm{F}}^2 = \sum_{j=1}^{K} r_j, \qquad \mathrm{RMS}(\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}})) = \sqrt{\frac{\sum_j r_j}{mnK}}. \tag{138}$$

If every transformed slice is full rank, then

$$\mathrm{RMS}(\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}) = \frac{1}{\sqrt{\max(m, n)}}, \tag{139}$$

independently of K.

Proof. We prove the rank–energy identity by analyzing the Frobenius norm of the polar factor in the transform domain and then transferring the result back to the original domain via unitary invariance.

For each transformed slice $\widehat{\boldsymbol{M}}_j \in \mathbb{C}^{m\times n}$, Lemma 2 states that the canonical polar factor $\operatorname{polar}(\widehat{\boldsymbol{M}}_j)$ has all nonzero singular values equal to one. Consequently, its squared Frobenius norm equals the number of nonzero singular values, which is precisely its rank:

$$\|\operatorname{polar}(\widehat{\boldsymbol{M}}_j)\|_{\mathrm{F}}^2 = \operatorname{rank}(\widehat{\boldsymbol{M}}_j) =: r_j. \tag{140}$$

This holds for each slice independently.

The transformed polar tensor $\widehat{\boldsymbol{\mathcal{P}}} := \boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}}) \times_3 \boldsymbol{Q}$ has frontal slices $\widehat{\boldsymbol{P}}_j = \operatorname{polar}(\widehat{\boldsymbol{M}}_j)$. Since the Frobenius norm of a tensor is the square root of the sum of the squared Frobenius norms of its frontal slices, we have

$$\|\widehat{\boldsymbol{\mathcal{P}}}\|_{\mathrm{F}}^2 = \sum_{j=1}^{K} \|\widehat{\boldsymbol{P}}_j\|_{\mathrm{F}}^2 = \sum_{j=1}^{K} r_j. \tag{141}$$

By definition, $\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}}) = \widehat{\boldsymbol{\mathcal{P}}} \times_3 \boldsymbol{Q}^\mathsf{H}$. Since $\boldsymbol{Q}$ is unitary, Lemma 1 guarantees that the mode-3 unitary transform preserves the Frobenius norm:

$$\|\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}})\|_{\mathrm{F}} = \|\widehat{\boldsymbol{\mathcal{P}}}\|_{\mathrm{F}}. \tag{142}$$

Combining this with equation 141 gives

$$\|\mathcal{P}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}})\|_{\mathrm{F}}^2 = \sum_{j=1}^{K} r_j. \tag{143}$$

This establishes the first part of the proposition.

The root-mean-square (RMS) value of a tensor is defined as the Frobenius norm divided by the square root of the total number of entries. Since $\mathcal{P}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}}) \in \mathbb{C}^{m\times n\times K}$ has mnK entries, we have

$$\mathrm{RMS}\big(\mathcal{P}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}})\big)^2 = \frac{1}{mnK}\|\mathcal{P}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}})\|_{\mathrm{F}}^2 = \frac{1}{mnK}\sum_{j=1}^{K} r_j. \tag{144}$$

Taking square roots yields the RMS formula in the proposition.

If every transformed slice is full rank, then each slice has rank $r_j = r := \min(m, n)$, since a matrix in $\mathbb{C}^{m\times n}$ can have rank at most $\min(m, n)$. Substituting $r_j = r$ into the RMS formula gives

$$\mathrm{RMS}\big(\mathcal{P}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}})\big)^2 = \frac{Kr}{mnK} = \frac{1}{\max(m, n)}. \tag{145}$$

Here we used the identity $mn/r = \max(m, n)$, which follows from $r = \min(m, n)$. Therefore,

$$\mathrm{RMS}\big(\mathcal{P}_{\boldsymbol{Q}}(\boldsymbol{\mathcal{M}})\big) = \frac{1}{\sqrt{\max(m, n)}}, \tag{146}$$

which is independent of K. This completes the proof. □

H.6 Non-unitary transforms

Then we show the necessity of the invertibility of transform $\boldsymbol{Q}$. Non-unitary coordinates do not preserve the Frobenius pairing, so it cannot simply use the proposed pipeline.

Theorem 7. *For a non-unitary transform $\boldsymbol{T}$. Let $\mathcal{L}_{\boldsymbol{T}}(\boldsymbol{\mathcal{D}}) := \boldsymbol{\mathcal{D}} \times_3 \boldsymbol{T}$ and $\|\boldsymbol{\mathcal{D}}\|_{\boldsymbol{T},\mathrm{Ts}} := \max_j \|\big(\mathcal{L}_{\boldsymbol{T}}\boldsymbol{\mathcal{D}}\big)_j\|_{\mathrm{op}}$. An exact radius-$\rho$ LMO is $\boldsymbol{\mathcal{D}}^* = -\rho\,\mathcal{L}_{\boldsymbol{T}}^{-1}\big[\mathrm{polar}_{\mathrm{slice}}\big(\mathcal{L}_{\boldsymbol{T}}^{-*}\boldsymbol{\mathcal{G}}\big)\big]$, where $\mathcal{L}_{\boldsymbol{T}}^{-*}\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{G}} \times_3 \boldsymbol{T}^{-\mathsf{H}}$. For unitary $\boldsymbol{T} = \boldsymbol{Q}$, $\boldsymbol{T}^{-\mathsf{H}} = \boldsymbol{Q}$ and the practical TUON rule is recovered.*

This Theorem shows that a non-orthogonal transform requires the inverse-adjoint correction above.

Proof. We derive the correct linear minimization oracle for a non-unitary transform by working in the transformed coordinate system.

First, we change variables to the transformed direction space. Let the transformed direction be

$$\widehat{\boldsymbol{\mathcal{D}}} = \mathcal{L}_{\boldsymbol{T}}\boldsymbol{\mathcal{D}}, \qquad \text{so that} \qquad \boldsymbol{\mathcal{D}} = \mathcal{L}_{\boldsymbol{T}}^{-1}\widehat{\boldsymbol{\mathcal{D}}}. \tag{147}$$

Under this change of variables, the trust-region constraint becomes

$$\max_{1\le j\le K} \|\widehat{\boldsymbol{D}}_j\|_{\mathrm{op}} \le \rho, \tag{148}$$

where $\widehat{\boldsymbol{D}}_j$ denotes the j-th frontal slice of $\widehat{\boldsymbol{\mathcal{D}}}$. This constraint is now separable across slices.

Second, we transform the linear objective correctly. A naive approach would transform the gradient by the same operator $\mathcal{L}_{\boldsymbol{T}}$, yielding $\mathcal{L}_{\boldsymbol{T}}\boldsymbol{\mathcal{G}}$. However, because $\mathcal{L}_{\boldsymbol{T}}$ is not unitary, this would not preserve the Frobenius pairing. The correct transformed gradient is obtained via the adjoint.

Recall that the adjoint of a linear map is defined by the identity

$$\langle \boldsymbol{\mathcal{A}}, \mathcal{L}_{\boldsymbol{T}}\boldsymbol{\mathcal{B}}\rangle_{\mathrm{F}} = \langle \mathcal{L}_{\boldsymbol{T}}^{*}\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}\rangle_{\mathrm{F}} \quad \text{for all } \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}. \tag{149}$$

Applying this to the objective, we have

$$\begin{aligned}
\langle \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{D}}\rangle_{\mathrm{F}} &= \langle \boldsymbol{\mathcal{G}}, \mathcal{L}_{\boldsymbol{T}}^{-1}\widehat{\boldsymbol{\mathcal{D}}}\rangle_{\mathrm{F}} && (150)\\
&= \langle (\mathcal{L}_{\boldsymbol{T}}^{-1})^{*}\boldsymbol{\mathcal{G}}, \widehat{\boldsymbol{\mathcal{D}}}\rangle_{\mathrm{F}} && (151)\\
&= \langle \mathcal{L}_{\boldsymbol{T}}^{-*}\boldsymbol{\mathcal{G}}, \widehat{\boldsymbol{\mathcal{D}}}\rangle_{\mathrm{F}}, && (152)
\end{aligned}$$

where $\mathcal{L}_{\boldsymbol{T}}^{-*} := (\mathcal{L}_{\boldsymbol{T}}^{-1})^*$ denotes the adjoint of the inverse operator. Thus, the correct gradient in the transformed coordinates is $\mathcal{L}_{\boldsymbol{T}}^{-*}\boldsymbol{\mathcal{G}}$, not the naive $\mathcal{L}_{\boldsymbol{T}}\boldsymbol{\mathcal{G}}$.

Then we solve the transformed block problem. With the constraint separable across slices and the objective expressed in terms of $\mathcal{L}_{\boldsymbol{T}}^{-*}\boldsymbol{\mathcal{G}}$, the optimization problem decouples into K independent matrix subproblems of the form

$$\min_{\|\widehat{\boldsymbol{D}}_j\|_{\mathrm{op}}\le\rho} \langle (\mathcal{L}_{\boldsymbol{T}}^{-*}\boldsymbol{\mathcal{G}})_j, \widehat{\boldsymbol{D}}_j\rangle_{\mathrm{F}}. \tag{153}$$

By Lemma 3, the optimal solution for each slice is the negative polar factor scaled by ρ:

$$\widehat{\boldsymbol{D}}_j^* = -\rho\,\mathrm{polar}\big((\mathcal{L}_{\boldsymbol{T}}^{-*}\boldsymbol{\mathcal{G}})_j\big). \tag{154}$$

Assembling the slices gives the transformed optimal direction

$$\widehat{\boldsymbol{\mathcal{D}}}^* = -\rho\,\mathrm{polar}_{\mathrm{slice}}\big(\mathcal{L}_{\boldsymbol{T}}^{-*}\boldsymbol{\mathcal{G}}\big)\,. \tag{155}$$

Mapping back to the original coordinates via $\mathcal{L}_{\boldsymbol{T}}^{-1}$ yields

$$\boldsymbol{\mathcal{D}}^* = -\rho\,\mathcal{L}_{\boldsymbol{T}}^{-1}\big[\mathrm{polar}_{\mathrm{slice}}\big(\mathcal{L}_{\boldsymbol{T}}^{-*}\boldsymbol{\mathcal{G}}\big)\big]\,, \tag{156}$$

which is the claimed inverse-adjoint LMO formula.

Finally, for the specific case of mode-3 multiplication, the inverse operator is $\mathcal{L}_{\boldsymbol{T}}^{-1}\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}} \times_3 \boldsymbol{T}^{-1}$. Its adjoint is therefore

$$(\mathcal{L}_{\boldsymbol{T}}^{-1})^*\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}} \times_3 (\boldsymbol{T}^{-1})^{\mathsf{H}} = \boldsymbol{\mathcal{X}} \times_3 \boldsymbol{T}^{-\mathsf{H}}. \tag{157}$$

Hence

$$\mathcal{L}_{\boldsymbol{T}}^{-*}\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{G}} \times_3 \boldsymbol{T}^{-\mathsf{H}}. \tag{158}$$

If $\boldsymbol{T} = \boldsymbol{Q}$ is unitary, then $\boldsymbol{Q}^{-\mathsf{H}} = \boldsymbol{Q}$, so the formula reduces to the familiar TUON rule:

$$\boldsymbol{\mathcal{D}}^* = -\rho\,\mathcal{L}_{\boldsymbol{Q}}^{-1}[\mathrm{polar}_{\mathrm{slice}}(\mathcal{L}_{\boldsymbol{Q}}\boldsymbol{\mathcal{G}})]\,, \tag{159}$$

which recovers the practical implementation from Section 2. □

H.7 A TIME-VARYING UNITARY BASIS

Moreover, in our algorithm, a data-dependent LSM basis may change during training. We therefore show whether the convergence argument survives when each step uses a different unitary geometry.

Theorem 8 (Time-varying unitary transforms)**.** *Let $\boldsymbol{Q}_t$ be a unitary transform selected from information available before the current update, and suppose the descent lemma at iteration t holds with constant L_t under $\|\cdot\|_{\boldsymbol{Q}_t,\mathrm{Ts}}$. For $\boldsymbol{\mathcal{W}}_{t+1} = \boldsymbol{\mathcal{W}}_t - \eta_t\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}_t}(\nabla F(\boldsymbol{\mathcal{W}}_t))$, we have*

$$\sum_{t=0}^{T-1}\eta_t\|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q}_t,\mathrm{Tn}} \le F(\boldsymbol{\mathcal{W}}_0) - F_\star + \frac{1}{2}\sum_{t=0}^{T-1}L_t\eta_t^2. \tag{160}$$

No explicit basis-drift term appears when momentum is stored in the original tensor coordinates and $\boldsymbol{Q}_t$ only defines the current oracle geometry. If momentum is stored in transformed coordinates, it must be transported whenever the basis changes. This distinction gives a concrete implementation rule for adaptive DecorQ.

Proof. We prove that time-varying unitary transforms preserve the convergence guarantee, provided the update direction is the exact LMO for the geometry defined at each iteration.

First we apply the descent lemma at iteration t. Let $\boldsymbol{Q}_t$ be the unitary transform used at iteration t, and let L_t be the smoothness constant of F with respect to the $\boldsymbol{Q}_t$ norm pair. At this iteration, the exact TUON update is

$$\boldsymbol{\mathcal{W}}_{t+1} = \boldsymbol{\mathcal{W}}_t - \eta_t\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}_t}(\nabla F(\boldsymbol{\mathcal{W}}_t)). \tag{161}$$

By Theorem 1, the polar direction $\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}_t}(\nabla F(\boldsymbol{\mathcal{W}}_t))$ is the exact LMO for the $\boldsymbol{Q}_t$-block geometry, so the descent lemma (Lemma 7) gives

$$F(\boldsymbol{\mathcal{W}}_{t+1}) \le F(\boldsymbol{\mathcal{W}}_t) - \eta_t\|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q}_t,\mathrm{Tn}} + \frac{L_t\eta_t^2}{2}. \tag{162}$$

This inequality holds for each t independently, without any reference to previous transforms $\boldsymbol{Q}_{t-1}, \boldsymbol{Q}_{t-2}, \ldots$.

Then, we sum equation 162 from $t = 0$ to $T - 1$ yields

$$\sum_{t=0}^{T-1} \eta_t \|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q}_t,\mathrm{Tn}} \leq F(\boldsymbol{\mathcal{W}}_0) - F(\boldsymbol{\mathcal{W}}_T) + \frac{1}{2}\sum_{t=0}^{T-1} L_t \eta_t^2. \tag{163}$$

Since F is bounded below by $F_\star$, we have $F(\boldsymbol{\mathcal{W}}_T) \geq F_\star$, so

$$\sum_{t=0}^{T-1} \eta_t \|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q}_t,\mathrm{Tn}} \leq F(\boldsymbol{\mathcal{W}}_0) - F_\star + \frac{1}{2}\sum_{t=0}^{T-1} L_t \eta_t^2. \tag{164}$$

This is the claimed cumulative bound.

□

H.8 Proof of Theorem 3: norm comparability

Theorem 3[Norm comparability with Muon] *Define* $\alpha_{\boldsymbol{Q}} := \max_j \sum_{k=1}^K |\boldsymbol{Q}_{jk}|$ *and* $\beta_{\boldsymbol{Q}} := \max_k \sum_{j=1}^K |\boldsymbol{Q}_{jk}|$. *Then, for every tensor* $\boldsymbol{\mathcal{X}}$ *and for their duals* $\boldsymbol{\mathcal{G}}$,

$$\frac{1}{\beta_{\boldsymbol{Q}}}\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}} \leq \|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} \leq \alpha_{\boldsymbol{Q}}\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}}, \quad \frac{1}{\alpha_{\boldsymbol{Q}}}\|\boldsymbol{\mathcal{G}}\|_{\mathrm{Muon},*} \leq \|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}} \leq \beta_{\boldsymbol{Q}}\|\boldsymbol{\mathcal{G}}\|_{\mathrm{Muon},*}. \tag{165}$$

Since $\boldsymbol{Q}$ *is unitary,* $\alpha_{\boldsymbol{Q}}, \beta_{\boldsymbol{Q}} \leq \sqrt{K}$.

Proof. We prove the norm equivalence by establishing bounds in both directions for the primal norms; the dual inequalities then follow from standard duality arguments. The proof proceeds in four steps.

First, we get the upper bound of the $\boldsymbol{Q}$ norm by the Muon norm. For each transformed slice, we have

$$\|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}} = \left\|\sum_{k=1}^K \boldsymbol{Q}_{jk}\boldsymbol{X}_k\right\|_{\mathrm{op}} \tag{166}$$

$$\leq \sum_{k=1}^K |\boldsymbol{Q}_{jk}|\,\|\boldsymbol{X}_k\|_{\mathrm{op}} \tag{167}$$

$$\leq \left(\sum_{k=1}^K |\boldsymbol{Q}_{jk}|\right) \max_{1\leq k\leq K} \|\boldsymbol{X}_k\|_{\mathrm{op}} \tag{168}$$

$$\leq \alpha_{\boldsymbol{Q}}\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}}. \tag{169}$$

Here, the first inequality uses the triangle inequality for the operator norm, and the second uses the fact that each term is bounded by the maximum over all slices. Taking the maximum over $j = 1, \ldots, K$ on both sides yields

$$\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} \leq \alpha_{\boldsymbol{Q}}\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}}. \tag{170}$$

Second we obtain the reverse bound using the inverse transform. Since $\boldsymbol{Q}$ is unitary, we have $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^{\mathsf{H}}$, so the inverse transform expresses each original slice as a linear combination of the transformed slices:

$$\boldsymbol{X}_k = \sum_{j=1}^K \overline{\boldsymbol{Q}}_{jk}\widehat{\boldsymbol{X}}_j. \tag{171}$$

Applying the same triangle-inequality argument,

$$\|\boldsymbol{X}_k\|_{\mathrm{op}} \le \sum_{j=1}^{K} |\boldsymbol{Q}_{jk}| \, \|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}} \tag{172}$$

$$\le \left(\sum_{j=1}^{K} |\boldsymbol{Q}_{jk}| \right) \max_{1 \le j \le K} \|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}} \tag{173}$$

$$\le \beta_{\boldsymbol{Q}} \|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}}. \tag{174}$$

Taking the maximum over $k = 1, \dots, K$ gives

$$\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}} \le \beta_{\boldsymbol{Q}} \|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}}. \tag{175}$$

Rearranging yields the lower bound:

$$\frac{1}{\beta_{\boldsymbol{Q}}} \|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}} \le \|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}}. \tag{176}$$

This completes the primal norm equivalence.

Then we derive the dual inequalities via duality. Applying Lemma 5 to the two primal inequalities in equation 170 and equation 176 gives the corresponding bounds for the dual norms. The order reversal is a standard property of norm duality. A larger primal unit ball corresponds to a smaller dual unit ball, and vice versa. Specifically, from $\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} \le \alpha_{\boldsymbol{Q}} \|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}}$, we obtain $\|\boldsymbol{\mathcal{G}}\|_{\mathrm{Muon},*} \le \alpha_{\boldsymbol{Q}} \|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}}$, and from $\|\boldsymbol{\mathcal{X}}\|_{\mathrm{Muon}} \le \beta_{\boldsymbol{Q}} \|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}}$, we obtain $\|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}} \le \beta_{\boldsymbol{Q}} \|\boldsymbol{\mathcal{G}}\|_{\mathrm{Muon},*}$. Combining these yields

$$\frac{1}{\alpha_{\boldsymbol{Q}}} \|\boldsymbol{\mathcal{G}}\|_{\mathrm{Muon},*} \le \|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}} \le \beta_{\boldsymbol{Q}} \|\boldsymbol{\mathcal{G}}\|_{\mathrm{Muon},*}. \tag{177}$$

We note that here $\boldsymbol{\mathcal{G}}$ denotes an arbitrary tensor playing the role of the dual variable; it need not be a gradient for the norm inequality itself.

Finally, we can bound the constants $\alpha_{\boldsymbol{Q}}$ and $\beta_{\boldsymbol{Q}}$. It remains to show that the distortion factors are controlled by $\sqrt{K}$. Since $\boldsymbol{Q}$ is unitary, every row $\boldsymbol{q}_j$ has Euclidean norm one:

$$\sum_{k=1}^{K} |\boldsymbol{Q}_{jk}|^2 = 1. \tag{178}$$

By Cauchy–Schwarz,

$$\sum_{k=1}^{K} |\boldsymbol{Q}_{jk}| \le \sqrt{K} \left(\sum_{k=1}^{K} |\boldsymbol{Q}_{jk}|^2 \right)^{1/2} = \sqrt{K}. \tag{179}$$

The same argument applies to every column, since the columns of a unitary matrix also have Euclidean norm one. Therefore,

$$\alpha_{\boldsymbol{Q}} \le \sqrt{K}, \qquad \beta_{\boldsymbol{Q}} \le \sqrt{K}. \tag{180}$$

Substituting equation 180 into equation 170 and equation 176 yields the universal bound stated in the theorem. □

Why the $\sqrt{K}$ bound matters. The bound $\alpha_{\boldsymbol{Q}}, \beta_{\boldsymbol{Q}} \le \sqrt{K}$ is the key stability statement of the theorem. It tells us that *no unitary layer transform can distort the Muon geometry by more than a factor of $\sqrt{K}$, regardless of the data or the choice of $\boldsymbol{Q}$.* In other words, changing the layer basis from the natural layer-wise geometry to any unitary-transformed geometry can shrink or expand the norm ball by at most $\sqrt{K}$. This ensures that the geometry of TUON is never pathologically far from that of layer-wise Muon, and provides the foundation for the convergence comparisons that follow. The same $\sqrt{K}$ factor reappears naturally in the TEON comparison, where it governs the maximum possible gain of a slice-wise polar oracle over a global unfolding polar.

H.9 Proof of Corollary 1: smoothness constants

Corollary 1[Smoothness comparison] *Let $L_{\boldsymbol{Q}}$ and L_{Muon} be the smallest constants satisfying*

$$\|\nabla F(\boldsymbol{\mathcal{X}}) - \nabla F(\boldsymbol{\mathcal{Y}})\|_{\boldsymbol{Q},\mathrm{Tn}} \le L_{\boldsymbol{Q}} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Y}}\|_{\boldsymbol{Q},\mathrm{Ts}} \tag{181}$$

and the analogous inequality under the Muon norm pair. Then

$$\frac{1}{\alpha_{\boldsymbol{Q}}^2} L_{\mathrm{Muon}} \le L_{\boldsymbol{Q}} \le \beta_{\boldsymbol{Q}}^2 L_{\mathrm{Muon}}. \tag{182}$$

Proof. Define the gradient difference and the variable difference as

$$Z \coloneqq \nabla F(\boldsymbol{\mathcal{X}}) - \nabla F(\boldsymbol{\mathcal{Y}}), \qquad D \coloneqq \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Y}}. \tag{183}$$

For any $\boldsymbol{\mathcal{X}} \neq \boldsymbol{\mathcal{Y}}$, Theorem 3 gives the dual-norm comparison

$$\|Z\|_{Q,\mathrm{Tn}} \le \beta_{\boldsymbol{Q}} \|Z\|_{\mathrm{Muon},*}, \tag{184}$$

and the primal-norm comparison

$$\|D\|_{Q,\mathrm{Ts}} \ge \frac{1}{\beta_{\boldsymbol{Q}}} \|D\|_{\mathrm{Muon}}. \tag{185}$$

Dividing equation 184 by equation 185 yields

$$\frac{\|Z\|_{Q,\mathrm{Tn}}}{\|D\|_{Q,\mathrm{Ts}}} \le \beta_{\boldsymbol{Q}}^2 \frac{\|Z\|_{\mathrm{Muon},*}}{\|D\|_{\mathrm{Muon}}}. \tag{186}$$

Taking the supremum over all pairs $\boldsymbol{\mathcal{X}} \neq \boldsymbol{\mathcal{Y}}$, and using the definitions of $L_{\boldsymbol{Q}}$ and L_{Muon}, gives

$$L_{\boldsymbol{Q}} \le \beta_{\boldsymbol{Q}}^2 L_{\mathrm{Muon}}. \tag{187}$$

For the reverse direction, Theorem 3 also gives

$$\|Z\|_{\mathrm{Muon},*} \le \alpha_{\boldsymbol{Q}} \|Z\|_{Q,\mathrm{Tn}}, \qquad \|D\|_{\mathrm{Muon}} \ge \frac{1}{\alpha_{\boldsymbol{Q}}} \|D\|_{Q,\mathrm{Ts}}. \tag{188}$$

The same ratio argument yields

$$\frac{\|Z\|_{\mathrm{Muon},*}}{\|D\|_{\mathrm{Muon}}} \le \alpha_{\boldsymbol{Q}}^2 \frac{\|Z\|_{Q,\mathrm{Tn}}}{\|D\|_{Q,\mathrm{Ts}}}. \tag{189}$$

Taking the supremum over all pairs again gives

$$L_{\mathrm{Muon}} \le \alpha_{\boldsymbol{Q}}^2 L_{\boldsymbol{Q}}, \tag{190}$$

which is equivalent to

$$\frac{1}{\alpha_{\boldsymbol{Q}}^2} L_{\mathrm{Muon}} \le L_{\boldsymbol{Q}}. \tag{191}$$

Combining equation 187 and equation 191 completes the proof. □

H.10 Proof of Theorem 4: comparison with TEON

Theorem 4[LMO gain with TEON] *For every unitary $\boldsymbol{Q}$ and tensor $\boldsymbol{\mathcal{X}}$, $\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} \le \|\boldsymbol{\mathcal{X}}\|_{\mathrm{TEON}} \le \sqrt{K}\, \|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}}$. Consequently,*

$$\|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*} \le \|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}} \le \sqrt{K}\, \|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*}. \tag{192}$$

Thus, at a common radius ρ, $\rho\|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*} \le \rho\|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}} \le \sqrt{K}\,\rho\|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*}$.

Proof. We establish the norm sandwich by analyzing the relationship between the horizontal unfolding and its block structure. The proof proceeds in three parts.

First, we express the TEON norm in terms of the transformed unfolding. Let

$$\boldsymbol{A} := \mathscr{U}(\widehat{\boldsymbol{\mathcal{X}}}) = [\widehat{\boldsymbol{X}}_1, \dots, \widehat{\boldsymbol{X}}_K], \tag{193}$$

where $\widehat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}} \times_3 \boldsymbol{Q}$ and $\widehat{\boldsymbol{X}}_j$ denotes its j-th frontal slice. By the identity $\mathscr{U}(\boldsymbol{\mathcal{X}} \times_3 \boldsymbol{Q}) = \mathscr{U}(\boldsymbol{\mathcal{X}})\boldsymbol{R}_{\boldsymbol{Q}}$ with $\boldsymbol{R}_{\boldsymbol{Q}} = \boldsymbol{Q}^{\mathsf{T}} \otimes \boldsymbol{I}_n$ unitary, we have

$$\|\boldsymbol{A}\|_{\mathrm{op}} = \|\mathscr{U}(\boldsymbol{\mathcal{X}})\|_{\mathrm{op}} = \|\boldsymbol{\mathcal{X}}\|_{\mathrm{TEON}}. \tag{194}$$

This identity will be used repeatedly in the bounds below.

Second, we get the lower bound (TEON norm dominates the $\boldsymbol{Q}$ norm). Fix any block index $j \in \{1, \dots, K\}$. Consider vectors whose only nonzero block is the j-th block. On this restricted subspace, the action of $\boldsymbol{A}$ is exactly that of the matrix $\widehat{\boldsymbol{X}}_j$. Therefore, the operator norm of $\boldsymbol{A}$ is at least the operator norm of any individual block:

$$\|\boldsymbol{A}\|_{\mathrm{op}} \geq \|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}}. \tag{195}$$

Since this holds for every j, taking the maximum over j gives

$$\|\boldsymbol{A}\|_{\mathrm{op}} \geq \max_{1 \leq j \leq K} \|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}} = \|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}}. \tag{196}$$

Combining this with equation 194 yields the lower bound:

$$\|\boldsymbol{\mathcal{X}}\|_{\mathrm{TEON}} \geq \|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}}. \tag{197}$$

Intuitively, the TEON norm looks at the entire concatenated matrix and therefore cannot be smaller than any one of its constituent blocks.

We now establish the reverse inequality. Observe that

$$\boldsymbol{A}\boldsymbol{A}^{\mathsf{H}} = \sum_{j=1}^{K} \widehat{\boldsymbol{X}}_j \widehat{\boldsymbol{X}}_j^{\mathsf{H}}. \tag{198}$$

The squared operator norm of $\boldsymbol{A}$ is the largest eigenvalue of $\boldsymbol{A}\boldsymbol{A}^{\mathsf{H}}$. By the linearity of the trace and the fact that the largest eigenvalue is bounded by the sum of the largest eigenvalues of each term,

$$\|\boldsymbol{A}\|_{\mathrm{op}}^2 = \lambda_{\max}(\boldsymbol{A}\boldsymbol{A}^{\mathsf{H}}) \tag{199}$$

$$\leq \sum_{j=1}^{K} \lambda_{\max}(\widehat{\boldsymbol{X}}_j \widehat{\boldsymbol{X}}_j^{\mathsf{H}}) \tag{200}$$

$$= \sum_{j=1}^{K} \|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}}^2 \tag{201}$$

$$\leq K \max_{1 \leq j \leq K} \|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}}^2. \tag{202}$$

Taking square roots on both sides gives

$$\|\boldsymbol{A}\|_{\mathrm{op}} \leq \sqrt{K} \max_{1 \leq j \leq K} \|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}}. \tag{203}$$

Since $\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} = \max_j \|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}}$, we obtain

$$\|\boldsymbol{\mathcal{X}}\|_{\mathrm{TEON}} \leq \sqrt{K}\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}}. \tag{204}$$

The key observation here is that the TEON norm aggregates energy from all blocks, but the operator norm can at most amplify the maximum block norm by a factor of $\sqrt{K}$.

Finally, we derive the dual and oracle inequalities. Applying the dual-order-reversal lemma (Lemma 5) to the primal sandwich in equation 197 and equation 204 yields the corresponding dual-norm sandwich:

$$\|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*} \leq \|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}} \leq \sqrt{K}\|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*}. \tag{205}$$

Finally, multiplying by the common trust-region radius ρ and invoking the exact LMO values from Theorem 1 (for the $\boldsymbol{Q}$ geometry) and Proposition 5 (for TEON) gives the radius-matched oracle gain:

$$\rho\|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*} \leq \rho\|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}} \leq \sqrt{K}\rho\|\boldsymbol{\mathcal{G}}\|_{\mathrm{TEON},*}. \tag{206}$$

This completes the proof. □

H.11 Proof of Theorem 5: convergence

Theorem 5 *Suppose F is lower bounded by $F_\star$ and is $L_{\boldsymbol{Q}}$-smooth under $\|\cdot\|_{\boldsymbol{Q},\mathrm{Ts}}$. For*

$$\boldsymbol{\mathcal{W}}_{t+1} = \boldsymbol{\mathcal{W}}_t - \eta \boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\nabla F(\boldsymbol{\mathcal{W}}_t)), \tag{207}$$

we have

$$\min_{0\le t<T} \|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q},\mathrm{Tn}} \le \frac{F(\boldsymbol{\mathcal{W}}_0) - F_\star}{\eta T} + \frac{L_{\boldsymbol{Q}}\eta}{2}. \tag{208}$$

Choosing

$$\eta = \sqrt{\frac{2(F(\boldsymbol{\mathcal{W}}_0) - F_\star)}{L_{\boldsymbol{Q}} T}} \tag{209}$$

gives

$$\min_{0\le t<T} \|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\mathrm{F}} \le \min_{0\le t<T} \|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q},\mathrm{Tn}} \le \sqrt{\frac{2L_{\boldsymbol{Q}}(F(\boldsymbol{\mathcal{W}}_0) - F_\star)}{T}}. \tag{210}$$

Proof. We prove the convergence guarantee for the exact TUON update. The proof proceeds in four steps.

First, we give the one-step descent under the $\boldsymbol{Q}$ geometry. The exact TUON update displacement at iteration t is

$$\boldsymbol{\mathcal{D}}_t = -\eta \boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\nabla F(\boldsymbol{\mathcal{W}}_t)). \tag{211}$$

By Theorem 1, the polar direction $\boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\nabla F(\boldsymbol{\mathcal{W}}_t))$ has tensor spectral norm under $\boldsymbol{Q}$-transform at most one, and satisfies the polar alignment identity

$$\langle \nabla F(\boldsymbol{\mathcal{W}}_t), \boldsymbol{\mathcal{P}}_{\boldsymbol{Q}}(\nabla F(\boldsymbol{\mathcal{W}}_t))\rangle_{\mathrm{F}} = \|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q},\mathrm{Tn}}. \tag{212}$$

Applying the general descent lemma (Lemma 7) under the $\boldsymbol{Q}$ norm pair gives

$$F(\boldsymbol{\mathcal{W}}_{t+1}) \le F(\boldsymbol{\mathcal{W}}_t) - \eta \|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q},\mathrm{Tn}} + \frac{L_{\boldsymbol{Q}}\eta^2}{2}. \tag{213}$$

This inequality captures the trade-off: the linear decrease is proportional to the gradient's $\boldsymbol{Q}$ dual norm, while the penalty is controlled by the smoothness constant $L_{\boldsymbol{Q}}$.

Second, we telescope the one-step inequality. Summing equation 213 from $t = 0$ to $T - 1$ yields

$$\eta \sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q},\mathrm{Tn}} \le F(\boldsymbol{\mathcal{W}}_0) - F(\boldsymbol{\mathcal{W}}_T) + \frac{L_{\boldsymbol{Q}}\eta^2 T}{2}. \tag{214}$$

Since F is bounded below by $F_\star$, we have $F(\boldsymbol{\mathcal{W}}_T) \ge F_\star$, so

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q},\mathrm{Tn}} \le \frac{F(\boldsymbol{\mathcal{W}}_0) - F_\star}{\eta T} + \frac{L_{\boldsymbol{Q}}\eta}{2}. \tag{215}$$

Since the minimum is always no larger than the average, we obtain equation 208.

Then we optimize the step size. Let $\Delta_0 := F(\boldsymbol{\mathcal{W}}_0) - F_\star$. The right-hand side of equation 215 as a function of η is

$$b(\eta) = \frac{\Delta_0}{\eta T} + \frac{L_{\boldsymbol{Q}}\eta}{2}. \tag{216}$$

This convex function is minimized when its derivative vanishes:

$$b'(\eta) = -\frac{\Delta_0}{\eta^2 T} + \frac{L_{\boldsymbol{Q}}}{2} = 0, \tag{217}$$

which gives the optimal step size

$$\eta^* = \sqrt{\frac{2\Delta_0}{L_{\boldsymbol{Q}} T}}. \tag{218}$$

Substituting η^* into equation 215 yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q},\mathrm{Tn}} \le \sqrt{\frac{2L_{\boldsymbol{Q}}\Delta_0}{T}}. \tag{219}$$

Finally, we relate the $\boldsymbol{Q}$ dual norm to the Frobenius norm. For any tensor $\boldsymbol{\mathcal{X}}$, the tensor spectral norm under $\boldsymbol{Q}$-transform is bounded above by the Frobenius norm:

$$\|\boldsymbol{\mathcal{X}}\|_{\boldsymbol{Q},\mathrm{Ts}} = \max_{1\le j\le K}\|\widehat{\boldsymbol{X}}_j\|_{\mathrm{op}} \le \left(\sum_{j=1}^{K}\|\widehat{\boldsymbol{X}}_j\|_{\mathrm{F}}^2\right)^{1/2} = \|\boldsymbol{\mathcal{X}}\|_{\mathrm{F}}, \tag{220}$$

where the equality follows from Parseval's identity for the unitary transform $\boldsymbol{Q}$. By the standard order-reversal property of dual norms (Lemma 5), this implies

$$\|\boldsymbol{\mathcal{G}}\|_{\mathrm{F}} \le \|\boldsymbol{\mathcal{G}}\|_{\boldsymbol{Q},\mathrm{Tn}} \tag{221}$$

for any tensor $\boldsymbol{\mathcal{G}}$. Applying this to $\boldsymbol{\mathcal{G}} = \nabla F(\boldsymbol{\mathcal{W}}_t)$ gives

$$\min_{0\le t<T}\|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\mathrm{F}} \le \min_{0\le t<T}\|\nabla F(\boldsymbol{\mathcal{W}}_t)\|_{\boldsymbol{Q},\mathrm{Tn}}. \tag{222}$$

Combining this with equation 219 yields equation 210. □

I Experiments

We empirically evaluate TUON against vector-aware, matrix-aware, and tensor-aware baselines on LLM pretraining. Our experiments are designed around five questions:

1. Does TUON improve pretraining validation loss over AdamW, Muon and TEON?
2. Is the method robust to stacking depth K, from $K = 2$ to $K =$ full?
3. Is the method robust to module coverage, i.e., does it remain effective when MLP projections are tensorized alongside attention?
4. Is the method robust across model scales and architectures?
5. Is the method robust across polarization methods?

I.1 Code and reproducibility

Our implementation strictly follows the PyTorch official API, i.e. it is inherited from the base class

`torch.optim`

. The complete production-ready code is publicly available.

The code of TUON is available at https://anonymous.4open.science/r/tuon-802A/README.md.

I.2 Datasets, models and training configuration.

This appendix provides a complete description of the model families, training codebases, and datasets used in our experiments (4).

GPT-2 (nanoGPT). For GPT-2, we adopt the nanoGPT codebase, a lightweight reimplementation of the GPT-2 decoder-only transformer featuring learned absolute positional embeddings, GELU-based MLPs, pre-LayerNorm residual blocks, and tied weights between the token embedding and the language modeling head. Two standard configurations are trained: a 124M 'small' variant (12 layers, 12 attention heads, hidden width 768) and a 350M 'medium' variant (24 layers, 16 heads, width 1024). Both use a 1024-token context window and the GPT-2 BPE vocabulary (50,304 padded entries). All full-parameter runs are trained for 50k optimizer steps, with a 2000-step linear warmup followed by cosine decay to one tenth of the peak learning rate.

Qwen3 (Megatron). The Qwen3 runs are conducted with the Megatron framework, which supports tensor, pipeline, and sequence parallelism for large-scale transformers. We use the Qwen3 dense decoder architecture — RMSNorm, SwiGLU MLPs, rotary positional embeddings, grouped-query attention, and QK-normalization — at two parameter scales, 0.6B and 4B. Each run lasts 20k iterations with a 2000-step warmup. The 4B model is trained with two-stage pipeline parallelism, while the 0.6B model fits within a single pipeline stage.

OpenWebText. OpenWebText is a public reproduction of the WebText corpus used to pretrain the original GPT-2. It consists of English web pages collected from outbound Reddit links, filtered by a minimum score threshold, deduplicated, and lightly cleaned. The corpus contains roughly 8B GPT-2 BPE tokens and is used as the pretraining data for both nanoGPT scales.

C4. C4 (Colossal Clean Crawled Corpus) is a large English web corpus built from Common Crawl and subjected to aggressive heuristic filtering, including deduplication, language identification, and removal of boilerplate and offensive material. It is a widely used and well-characterized pretraining corpus, and we use it without modification for both Qwen3 scales.

Finally, all runs use sequence length 2048, global batch size 256, and seed 42. GPT-2 runs use learning rate 1×10^{-3} (Moonlight adjusted-LR); Qwen3 runs use 2×10^{-3} with TP (Tensor Parallelism)=1 and PP (Pipeline Parallelism)=1.

I.3 Baselines.

We compare against three families of optimizers.

Vector-aware. `AdamW` (Loshchilov & Hutter, 2017) is the standard coordinate-wise baseline and represents the vector-aware family. For AdamW, we sweep the peak learning rate for each model and scale, selecting the value that yields the best final validation loss. The selected peak learning rates are reported in Table 5.

Matrix-aware. `Muon` (Jordan et al., 2024) applies a polar map to each weight matrix independently. It is the strongest layer-wise matrix-aware optimizer and serves as the reference point for all reported Δ values.

Tensor-aware. We use TEON (Zhang et al., 2026a) as the tensor-aware baseline. TEON relies on depends on several hyperparameters, including the number of stacked layers, the module types to which tensorization is applied, and the unfolding mode. We therefore treat TEON as a baseline family and evaluate the following variants:

- two-layer stacking;
- stacking all compatible layers along the layer mode;
- applied only to QKV projections;
- extended to attention and MLP;
- TEON-style concat polar with mode-1 or mode-2 unfolding;

For each model scale, we select the best-performing TEON configuration and report it as the tensor-aware baseline. This ensures that the comparison is against the strongest TEON variant rather than against a single arbitrary configuration. All tensor-aware methods, including ours, are tuned under the same protocol.

TUON implementation. For each compatible parameter group, we stack the momentum tensors along the layer mode to form $\mathcal{M}_t \in \mathbb{R}^{m \times n \times K}$. We compute the mode-3 unfolding $\mathsf{M}_t = \texttt{mode3_unfold}(\boldsymbol{\mathcal{M}}_t)$, take its SVD $\mathsf{M}_t = \mathbf{U}_t \mathbf{\Sigma}_t \mathbf{V}_t^\mathsf{T}$, and set $\boldsymbol{Q}_t = \mathbf{U}_t^\mathsf{T}$. The transform is refreshed every 200 iterations. We then apply $\widehat{\boldsymbol{\mathcal{M}}}_t = \boldsymbol{\mathcal{M}}_t \times_3 \boldsymbol{Q}_t$, polarize each transformed frontal slice independently using a Newton–Schulz iteration with $J = 5$ steps, and invert the transform. The scaling factor is $s = 1/\sqrt{\max(m, n)}$ with $\epsilon = 10^{-8}$, and weight decay is decoupled. The key mechanistic difference from TEON is that TUON uses per-slice polar after a unitary mixing, whereas TEON uses concat polar with a shared singular basis.

Table 5: Peak learning rates for the AdamW baseline on GPT-2 (124M, 350M) and Qwen3 (0.6B,4B), respectively.

optimizer	GPT-2 124M	GPT-2 350M	Qwen3 0.6B	Qwen3 4B
AdamW	2.4×10^{-3}	2×10^{-3}	1×10^{-3}	3×10^{-4}
Muon	1×10^{-3}	1×10^{-3}	2×10^{-3}	1×10^{-3}
TEON	1×10^{-3}	1×10^{-3}	2×10^{-3}	1×10^{-3}
TUON	1×10^{-3}	1×10^{-3}	2×10^{-3}	1×10^{-3}

Table 6: Validation loss on GPT-2 (124M, 350M; 50k iterations) and Qwen3 (0.6B, 4B; 20k iterations) with different optimizers. $\Delta = \text{Val} - \text{Val}_{\text{AdamW}}$ within the same model and horizon; negative Δ indicates improvement over AdamW. Best per column in bold. Lower is better.

Optimizer	**124M$_{\text{GPT-2}}$**		**350M$_{\text{GPT-2}}$**		**0.6B$_{\text{Qwen3}}$**		**4B$_{\text{Qwen3}}$**	
	Val	Δ	Val	Δ	Val	Δ	Val	Δ
AdamW	**2.9230**	–	2.7307	–	2.9652	–	2.8108	–
Muon	3.0488	$+0.1258$	2.7404	$+0.0097$	2.9056	-0.0596	2.7636	-0.0472
TEON	2.9442	$+0.0212$	2.6935	-0.0372	2.8978	-0.0674	2.7604	-0.0504
TUON	2.9357	$\mathbf{+0.0127}$	**2.6925**	$\mathbf{-0.0382}$	**2.8887**	$\mathbf{-0.0765}$	**2.7598**	$\mathbf{-0.0510}$

Evaluation protocol. We use different training horizons for the two model families. GPT-2 runs are trained for 50k iterations, which is close to the standard nanoGPT/OpenWebText convergence point and allows absolute validation loss to be compared across methods. Qwen3 runs are trained for 20k iterations under a fixed compute budget. To keep the comparison internally consistent, we never compare absolute loss across families or horizons; all cross-method comparisons are made within the same family and the same horizon. For all runs we report final validation loss, intermediate validation loss every 2k iterations at the same family and horizon.

I.4 Hyperparameters

We sweep the peak learning rate for each model and scale, selecting the value that yields the best final validation loss. The selected peak learning rates are reported in Table 5.

I.5 Results on LLM pretraining

Advantages across scales and architectures. Table 6 reports validation loss on GPT-2 (124M, 350M) and Qwen3 (0.6B, 4B). TUON achieves the lowest validation loss on three of the four settings, consistently improving over Muon and TEON. On GPT-2 124M, where AdamW remains the strongest baseline, TUON still achieves the smallest gap among all tensor-aware methods.

Comparison with AdamW. Relative to AdamW, TUON reduces validation loss on GPT-2 350M and both Qwen3 scales, with improvements of 0.0382, 0.0765, and 0.0510, respectively. On GPT-2 124M, where all optimizers remain close to AdamW, TUON still outperforms Muon and TEON by 0.1131 and 0.0085, respectively.

Comparison with Muon. Muon is competitive at larger scale models, where it trails AdamW by 0.1258. In contrast, TUON remains stable across scales, outperforming Muon on all four settings. The gap is largest at 124M (0.1131) and remains consistent at 350M, 0.6B, and 4B, suggesting that the cross-layer structure exploited by TUON is particularly beneficial when the model is small and layer-wise orthogonalization alone is insufficient.

Comparison with TEON. TEON is the strongest baseline among the tensor-aware optimizers, but TUON still improves over it on every model. The gains are modest at 350M (0.0010) and 4B (0.0006), but more pronounced at 124M (0.0085) and 0.6B (0.0091). This pattern is consistent with our theoretical analysis: TUON avoids the global unfolding used by TEON, so its advantage is most visible when the number of stacked layers grows and the unfolding becomes increasingly ill-conditioned.

Table 7: Effect of transform $\boldsymbol{Q}$ on full layer stacking.

Transform	2k	4k	8k	12k	16k	18k	20k	Final
DCT	3.5411	3.3646	3.1192	2.9444	2.8411	2.8189	2.8051	2.8100
Haar	3.4437	3.1413	2.9302	2.8321	2.7655	2.7540	2.7579	2.7635
DFT	3.3981	3.1002	2.9163	2.8223	2.7599	2.7503	2.7579	2.7631
DecorQ static	3.3999	3.0975	2.9145	2.8206	2.7582	2.7482	2.7569	2.7658
Residual-DecorQ	3.4100	3.1079	2.9204	2.8255	2.7615	2.7511	2.7585	2.7640
DecorQ	3.3899	3.0916	2.9109	2.8185	2.7564	2.7461	2.7545	2.7598

Overall, TUON delivers consistent improvements over vector-, matrix-, and tensor-aware baselines across two architectures and four model scales, without requiring per-scale hyperparameter tuning. These results support the claim that a transform-based tensor algebra provides a more effective geometry for cross-layer optimization than both layer-wise orthogonalization and global unfolding.

I.6 Ablation studies

We conduct ablation studies to isolate the effect of each design choice in our method. Unless otherwise noted, all ablations are run for 20k iterations.

Choices of $\boldsymbol{Q}$. We first compare different choices of the layer transform $\boldsymbol{Q}$, including DCT, Haar Wavelet Transform, DFT and DecorQ. We also explore the variant of the DecorQ transform. In addition to the default DecorQ, we evaluate a static DecorQ basis frozen at initialization and residual-DecorQ where the transform is computed from residual statistics rather than momentum. Specifically, instead of estimating the DecorQ from momentum M_t, we consider a residual variant that uses $R_t = G_t - M_t$ or the projection residual $R_t = G_t - Q_{t-1}Q_{t-1}^\top G_t$. The basis is obtained by eigendecomposing the residual covariance and keeping the top-r eigenvectors, and is refreshed either periodically (sampled) or at every step via an EMA of the covariance (continuous). A low-rank variant retains only $r = 10$ directions. Empirically, the residual variant matches the momentum variant in both the full-layer and $K = 2$ settings, but does not surpass the mode-2 concat polar combination, indicating that the residual signal is a viable but not dominant source for basis estimation.

Table 7 reports results for full-layer stacking, where all layers within the same semantic role are stacked together.

In the original T-SVD-based tensor geometry, $\boldsymbol{Q}$ is only required to be invertible; any unitary or invertible transform yields a valid tensor algebra. However, this algebraic requirement alone does not specify which transform is appropriate for cross-layer optimization in Transformers. Our analysis in Section 2 suggests that the choice of $\boldsymbol{Q}$ should be guided by the layer statistics rather than left arbitrary. The ablation below tests this hypothesis directly.

Overall trend. All transforms outperform DCT, confirming that the choice of $\boldsymbol{Q}$ materially affects optimization. DFT, Haar, and the DecorQ family are consistently stronger than DCT across all checkpoints, with the gap widening as training progresses. This indicates that a fixed, data-independent transform such as DCT imposes a prior on cross-layer coupling that does not match the actual layer statistics in LLMs.

Data-dependent versus fixed transforms. The DecorQ achieves the best final loss (2.7598), outperforming DFT (2.7631), Haar (2.7635), and DCT (2.8100). Its advantage is visible from the earliest checkpoint (2k) and remains stable through 20k, suggesting that a data-dependent basis is consistently better aligned with the cross-layer structure than fixed transforms. Static DecorQ, which freezes the basis at initialization, reaches a competitive 2.7658 but is slightly worse than the adaptive DecorQ, indicating that refreshing the transform during training provides a small yet consistent benefit. This supports the view that $\boldsymbol{Q}$ should not be chosen arbitrarily. A transform adapted to layer statistics is preferable to any fixed choice.

Residual-DecorQ. Replacing momentum with residual statistics yields a final loss of 2.7640, which is better than static DecorQ and comparable to DFT, but does not surpass DecorQ. This suggests that the residual signal is a viable but not dominant source for basis estimation. As noted in the text, the

Table 8: Effect of stacking size K. (↑) / (↓) denote whether the full-K setting is worse / better than $K = 2$. 20k iterations.

Model	Muon	$\text{TEON}_{\text{mode-1}}$		$\text{TEON}_{\text{mode-2}}$		TUON_{DCT}		$\text{TUON}_{\text{DecorQ}}$	
		K=2	K=full	K=2	K=full	K=2	K=full	K=2	K=full
124M$_{\text{GPT-2}}$	3.0583	3.0255	3.0513(↑)	3.0332	3.0789(↑)	3.0309	3.0295(↓)	3.0299	**3.0233**(↓)
350M$_{\text{GPT-2}}$	2.8427	2.8427	2.8617(↑)	2.8643	2.8762(↑)	2.8150	2.8218(↑)	2.8183	**2.8143**(↓)
0.6B$_{\text{Qwen3}}$	2.9056	2.8994	2.9323(↑)	2.8978	2.9583(↑)	2.9203	2.9210(↑)	2.8938	**2.8887**(↓)
4B$_{\text{Qwen3}}$	2.7636	2.7627	2.7649(↑)	2.7604	2.7640(↑)	2.7606	2.8100(↑)	2.7629	**2.7598** (↓)

residual variant also does not surpass the mode-2 concat polar combination, reinforcing that DecorQ is the preferred data-dependent choice.

Overall, the ablation confirms the design principle behind TUON. Although the underlying tensor algebra only requires $\boldsymbol{Q}$ to be invertible, our data-dependent transform that reflects layer statistics is empirically superior. This validates the choice of DecorQ as the default transform and supports the broader claim that the layer mode should be treated as a structured dimension rather than an arbitrary algebraic index.

Stacking size K. Intuitively, a larger stacking size K exposes more cross-layer structure to the optimizer, and the theoretical analysis in Section 3.2 and in (Zhang et al., 2026a) suggests that increasing K raises the potential maximum performance gain. In this part, we examine whether this potential is actually realized in practice. We vary K by grouping adjacent layers, while all other settings follow the default configuration of our method and are kept fixed across all runs. For computational reasons we train all models for 20k iterations.

As shown in Table 8, TEON achieves its best performance at $K = 2$ and **consistently degrades as K increases, for both mode-1 and mode-2 unfolding**. On GPT-2 124M, moving from $K = 2$ to $K =$ full increases the validation loss from 3.0255 to 3.0513 in mode-1 and from 3.0332 to 3.0789 in mode-2; similar trends hold on the larger models. This is consistent with the observations in Zhang et al. (2026a). and reveals a fundamental gap: the theory promises more gain from larger K, yet unfolding methods consistently lose performance as K grows. It suggests two underlying causes. First, directly unfolding a tensor to handle cross-layer coupling can lose structural information. Second, even under mode-1 or mode-2 unfolding, it is difficult to align the top singular vectors across all K matrices, so the shared basis becomes increasingly mismatched as K grows.

These results break the theoretical expectation that stacking more layers yields a greater potential gain. The gap between theory and practice may come from the tensor algebra. Specifically, the fixed transform to exploit the additional structure: a larger K offers more opportunity for decorrelation, yet a static basis cannot adapt to the increasingly diverse singular subspaces of the grouped layers.

Our proposed method, TUON with the dynamic DecorQ layer transform, brings this expectation much closer to the theoretical prediction. As shown in the last two columns of Table 8, DecorQ is the only variant that improves when moving from $K = 2$ to $K =$ full on all four models: $3.0299 \to 3.0233$ on GPT-2 124M, $2.8183 \to 2.8143$ on GPT-2 350M, $2.8938 \to 2.8887$ on Qwen3 0.6B, and $2.7629 \to 2.7598$ on Qwen3 4B. In contrast, the DCT-based variant still degrades on GPT-2 350M, Qwen3 0.6B, and Qwen3 4B, and only marginally improves on GPT-2 124M. This contrast isolates the effect of the dynamic layer transform. It is not the larger K alone that helps, but the ability to adapt the shared basis to the grouped layers.

Actually, we need to admit that the gains from larger K are real but bounded. On Qwen3 4B, DecorQ improves from 2.7629 to 2.7598, a reduction of 0.0031. This aligns with the theoretical prediction that larger K provides more opportunity for decorrelation but eventually hits diminishing returns when layers become too diverse to share a common structure. The dynamic transform pushes this turning point further out, but does not remove it.

Layer type. Existing tensor-aware optimizers for LLMs are largely confined to specific module types, such as QKV-specific structures, and have not been successfully transferred to other modules such as MLPs or feed-forward projections. This restriction is not incidental. Extending the tensor

Table 9: Effect of layer type. $\text{Module}_{\text{QKV}}$ mirrors the attention-only scope of existing tensor-aware optimizers, while $\text{Module}_{\text{All}}$ extends the transform to QKVO and MLP. Lower is better; the best result for each model is highlighted in bold. 20k iterations.

Model	Muon	$\text{TEON}_{\text{mode-1}}$		$\text{TEON}_{\text{mode-2}}$		$\text{TUON}_{\text{DecorQ}}$	
		$\text{Module}_{\text{QKV}}$	$\text{Module}_{\text{All}}$	$\text{Module}_{\text{QKV}}$	$\text{Module}_{\text{All}}$	$\text{Module}_{\text{QKV}}$	$\text{Module}_{\text{All}}$
124M$_{\text{GPT-2}}$	3.0583	3.0255	3.0531	3.0332	3.1027	3.0225	**3.0233**
350M$_{\text{GPT-2}}$	2.8258	2.8427	2.8637	2.8643	2.9144	2.8291	**2.8143**
0.6B$_{\text{Qwen3}}$	2.9056	2.8994	2.9023	2.8978	2.9018	2.8897	**2.8887**
4B$_{\text{Qwen3}}$	2.7636	2.7627	2.7644	2.7604	2.7639	2.7606	**2.7598**

transform to a wider set of modules requires the shared structure to remain well aligned across heterogeneous layer types, which fixed transforms struggle to achieve.

Our method removes this restriction and applies uniformly to all layer types. In this part, we examine whether this broader applicability translates into practical gains. We compare two settings. $\text{Module}_{\text{QKV}}$ restricts the optimizer to the attention projections, mirroring the scope of existing tensor-aware optimizers for LLMs. $\text{Module}_{\text{All}}$ extends it to QKVO and MLP, covering all major weight tensors in the network. All other settings follow the default configuration of our method and are kept fixed across all runs. This comparison directly tests whether our method can escape the attention-only regime that prior tensor-aware optimizers are confined to, and whether the resulting gains extend beyond attention projections. For computational reasons we train all models for 20k iterations.

Table 9 reports the results. Extending the tensor transform from QKV to all major weight tensors has a sharply different effect across methods. TEON degrades in every configuration: under mode-1, the validation loss increases on all four models, and under mode-2 the degradation is even more severe, e.g., from 3.0332 to 3.1027 on GPT-2 124M and from 2.8643 to 2.9144 on GPT-2 350M. This confirms that a fixed transform cannot maintain alignment across heterogeneous layer types.

Our method exhibits the opposite behavior. Moving from $\text{Module}_{\text{QKV}}$ to $\text{Module}_{\text{All}}$ achieves improvements or comparable performance of validation losses on GPT-2 124M ($3.0225 \to 3.0233$), GPT-2 350M ($2.8291 \to 2.8143$), Qwen3 0.6B ($2.8897 \to 2.8887$), and Qwen3 4B ($2.7606 \to 2.7598$). More importantly, in the $\text{Module}_{\text{All}}$ setting, our method outperforms the best TEON variant by 0.0298, 0.0474, 0.0131, and 0.0041 on the four models, respectively. These results show that the benefit of our method is not specific to attention projections, but extends to MLP and output projections, where prior tensor-aware optimizers fail to transfer.

Unfolding mode. TEON first unfolds a tensor into a matrix and then applies polarization to the resulting matrix, which is equal to introduce an additional hyperparameter: whether to unfold along mode-1 or mode-2. Proposition 4.6 of (Zhang et al., 2026a) states that mode-1 orthogonalization should be preferred when the top right singular vectors v_1 have high similarity, and mode-2 orthogonalization should be preferred when the top left singular vectors u_1 have high similarity. Empirically, Section 6.1 of (Zhang et al., 2026a) reports that mode-1 orthogonalization works better in practice.

Our experiments reveal a different picture. As shown in Table 10, the better mode is *model-dependent*: GPT-2 124M and GPT-2 350M favor mode-1, while Qwen3 0.6B and Qwen3 4B favor mode-2. The gap is not negligible: on GPT-2 350M, choosing the wrong mode costs 0.0216 in validation loss. There is no single mode that dominates across all models, which is consistent with the theoretical condition in Proposition 4.6 being model-dependent itself.

This model dependence has two consequences. First, it adds a non-trivial hyperparameter to TEON: practitioners must decide which mode to use for each new model, and the wrong choice can cost a non-negligible amount of performance. Second, it shows that neither unfolding mode provides a universally correct way to align the shared basis, which is consistent with the theoretical condition in Proposition 4.6 being model-dependent itself.

Our method does not require unfolding at all, and therefore avoids this hyperparameter choice entirely. It is noticed that, in the main experiments, **we report the stronger of the two TEON modes for each model**, so that the comparison is made against the most favorable TEON configuration. As

Table 10: Effect of unfolding mode in TEON. The better mode is highlighted in bold for each model. Lower is better.

Model	Muon	$\text{TEON}_{\text{mode-1}}$	$\text{TEON}_{\text{mode-2}}$
$124\text{M}_{\text{GPT-2}}$	3.0583	**3.0255**	3.0332
$350\text{M}_{\text{GPT-2}}$	2.8258	**2.8427**	2.8643
$0.6\text{B}_{\text{Qwen3}}$	2.9056	2.8994	**2.8978**
4B_{Qwen3}	2.7636	2.7627	**2.7604**

Table 11: Comparison of standard Newton–Schulz and PolarExpress for polar approximation on Qwen3_{4B}.

Stacking layer number	**Method**	**Final val**
2	Newton–Schulz ($J = 5$)	2.7629
	PolarExpress	2.7638
Full	Newton–Schulz ($J = 5$)	2.7598
	PolarExpress	2.7569

shown in Table 6, TUON outperforms the *best* TEON variant on every model, regardless of which mode is better for that model. Even under this conservative protocol, our method remains ahead.

Polarization methods. Our optimizer relies on a polar map, which is approximated in practice by Newton–Schulz-type polynomial iterations. Exact SVD is prohibitively expensive inside every optimizer step, so a fast iterative approximation is required. The choice of this approximation is therefore a practical design question: if the optimizer's performance depends strongly on which polar iteration is used, then the method would inherit an additional hyperparameter with no principled default.

We compare the standard Newton–Schulz iteration with $J = 5$ steps against the PolarExpress approximation (Amsel et al., 2026), which designs a stage-dependent minimax polynomial sequence for rapid polar approximation and is directly applicable to the real transformed slices produced by an orthonormal DCT. Table 11 reports the results for two representative configurations on Qwen3-4B: $K = 2$ with the DecorQ transform, and full-layer stacking with the DecorQ transform.

For $K = 2$, the two methods perform comparably: Newton–Schulz reaches 2.7629 and PolarExpress 2.7638, a difference of 0.0009. For full-layer stacking, PolarExpress is slightly better: 2.7569 against 2.7598, a difference of 0.0029. Notably, the sign of the difference flips between the two settings, so neither method dominates. In both cases, the gap is an order of magnitude smaller than the gain obtained from our design choices; for reference, the gain from DCT to DecorQ is 0.047. This confirms that the polarization method is a second-order factor in our framework rather than a critical one.

This robustness has a practical implication: practitioners can freely swap the polarization method according to their computational budget without jeopardizing the main conclusions. The performance of our method is driven by the dynamic layer transform and the coupling structure, not by the specific polar iteration used to approximate the polar factor.

Notably, PolarExpress is optimized for *real* transformed slices produced by DCT (Amsel et al., 2026), a condition that holds exactly in our setting. The small residual differences at both stacking sizes suggest that its approximation error remains modest even when matrix sizes are larger and the layer structure is more complex. In practice, PolarExpress offers a favorable trade-off: it reduces the computational cost of the polar approximation while maintaining competitive performance, especially in the full-layer regime where our best configuration resides. For the main results, we retain the standard Newton–Schulz iteration for simplicity and reproducibility, while noting that PolarExpress is a viable drop-in replacement for deployment scenarios with stringent computational budgets.

Table 12: Time per iteration (ms) on four models. Lower values indicate faster training. Our method incurs only marginal overhead compared with AdamW, Muon, and TEON.

Model	Time per Iteration (ms)			
	124M$_{\text{GPT-2}}$	**350M$_{\text{GPT-2}}$**	**0.6B$_{\text{Qwen3}}$**	**4B$_{\text{Qwen3}}$**
AdamW	**152.74**	**379.95**	**3265.5**	**5466.9**
Muon	160.88	404.81	3324.3	5794.7
TEON	161.81	408.30	3385.7	6081.5
TUON	162.12	411.64	3378.3	6150.7

I.7 Training time comparison

We compare the per-iteration step time of AdamW, Muon, TEON, and our method. All experiments are conducted on identical hardware and software environments to ensure a fair comparison. For each model, we fix the global batch size, sequence length, and the number of training iterations. We report the average time per iteration (ms), which excludes the first few iterations to eliminate the influence of compilation, caching, and initialization overhead.

For each model, all optimizers share the same backbone, data pipeline, and parallelism configuration. Only the optimizer-specific computation and update are varied.

Table 12 reports the time per iteration (ms) of AdamW, Muon, TEON, and our method on the GPT-2 124M/350M and Qwen3 0.6B/4B models. The goal of this comparison is not to claim a step-time advantage, but to verify that our method does not introduce non-trivial computational overhead relative to widely used optimizers.

First, the overall trend is that, across all four models, the step time of our method is close to that of Muon and TEON. On GPT-2 124M, our method takes 162.12 ms, which is 0.2% slower than TEON (161.81 ms) and 0.8% slower than Muon (160.88 ms). On GPT-2 350M, the corresponding gaps are 0.8% and 1.7%, respectively. On the larger Qwen3 models, the differences remain small: on Qwen3 0.6B, our method is 0.2% slower than TEON and 1.6% slower than Muon. On Qwen3 4B, it is 1.1% slower than TEON and 6.1% slower than Muon.

AdamW is the fastest in absolute terms on all four models, which is expected since it only maintains first- and second-order moment estimates and performs element-wise updates. Our method, in absolute terms, is slower than AdamW by only 9.4 ms, 31.7 ms, 112.8 ms, and 683.8 ms per iteration. Given that a full training run typically involves around 20K-50K iterations, the relative overhead per iteration is modest and does not change the overall training cost qualitatively.

When it comes to the scaling behavior, an important observation is that the relative overhead of our method does not grow disproportionately with model size. Between GPT-2 124M and Qwen3 4B, the model size increases by more than an order of magnitude, yet the step time of our method remains within 12.5% of AdamW and within 1.1% of TEON on the largest model. This suggests that the additional computation in our optimizer is well amortized by the forward and backward passes, and that our method scales gracefully to larger models.

Overall, the step-time results show that our method is essentially on par with Muon and TEON, and only marginally slower than AdamW. The small per-iteration overhead is a reasonable price for the optimization benefits reported in Section 2.2, and it does not constitute a practical bottleneck for large-scale training.