STIEFEL-ADAMW: GEOMETRY-AWARE ADAMW FOR LINEAR FACTORIZATION BLOCKS

Anonymous authors
Paper under double-blind review

ABSTRACT

A pervasive structural pattern in modern deep learning is the *linear factorization block*: a submodule of the form $W = BA$ in which two parameter matrices are multiplied directly, with no intervening nonlinearity. Such blocks appear in LoRA adapters, low-rank compressed layers, query-key products of self-attention, and share a common pathology: the factorization is non-unique, which can destabilize training and limit usable learning rates. Despite this, factorization blocks are typically optimized with standard Euclidean methods that ignore the underlying geometry. We introduce Stiefel-AdamW, a near drop-in replacement for AdamW for use wherever such blocks appear. By constraining one factor on the Stiefel manifold while leaving the other Euclidean, Stiefel-AdamW relaxes the full $\mathrm{GL}(\mathbb{R}^r)$ gauge symmetry to a compact orthogonal symmetry, ruling out factor blow-up while retaining the coordinate-wise diagonal preconditioning that gives AdamW its practical strength. Moment estimation is performed in the ambient Euclidean space, with geometry entering only through a tangent-space projection and a manifold retraction. The implementation overhead over AdamW is minimal, and we show that the resulting optimizer inherits both the stability benefits of Riemannian methods and standard convergence guarantees. We validate Stiefel-AdamW on LoRA-style fine-tuning of GPT2, ViT, and Mistral 7B and on full pretraining of GPT2 on OpenWebText and Qwen2 on Wikitext-103, showing improvements over strong baselines at essentially no additional cost over AdamW.

1 INTRODUCTION

A pervasive structural pattern in modern deep-learning architectures is the presence of *linear factorization blocks*: trainable submodules of the form $W = BA$, in which two parameter matrices are multiplied directly, with no intervening nonlinearity. Such blocks arise in many guises across the modern model zoo. In LoRA-style parameter-efficient fine-tuning Hu et al. (2022); Zhang et al. (2023); Hayou et al. (2024); Zhang & Pilanci (2024); Zhao et al. (2024); Lialin et al. (2024); Schotthöfer et al. (2025b), weight correctors are parametrized as $\Delta W = BA$, with $B \in \mathbb{R}^{m\times r}$ and $A \in \mathbb{R}^{r\times n}$. In low-rank pretraining Wang et al. (2021); Khodak et al. (2021); Schotthöfer et al. (2025a) and network compression Vogels et al. (2019); Saha et al. (2023); Mo et al. (2025); Schotthöfer et al. (2022), weight matrices are similarly factorized into two trainable factors. The same structure also appears *inside* standard architectures: in self-attention Vaswani et al. (2017), the score matrix $W_Q W_K^\top$ factorizes through the rank-r key/query head dimension, and analogous matrix-matrix factorizations appear in the recurrence kernels of several structured state-space models (SSMs). In all these cases, two parameter blocks combine multiplicatively, without an intervening nonlinearity, to produce a single effective linear operator.

These blocks share a common geometric feature: the product map $\Phi(A, B) = BA$ is highly non-unique. For any $M \in \mathrm{GL}(\mathbb{R}^r)$, $\Phi(A, B) = \Phi(M^{-1}A,\, BM)$, so each effective weight W corresponds to a continuous family of parameter pairs. This gauge symmetry has direct consequences for optimization. If the loss $\mathcal{L}$, viewed as a function of the effective matrix W, has a stationary point, then $\mathcal{L} \circ \Phi$ has an entire orbit of equivalent stationary points parametrized by $\mathrm{GL}(\mathbb{R}^r)$. More worryingly, training along these blocks can be numerically unstable even without spurious minimizers: a sequence $(M_n^{-1}A_n,\, B_n M_n)$ with $M_n \to 0$ produces a bounded product W_n even though one factor diverges. These issues are well documented in practice Mishra et al. (2014); Schotthöfer et al.

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.

(2022); Zangrando et al. (2024) and become especially pronounced under unbalanced initializations, such as those typical of LoRA.

Despite the prevalence of this structure, standard practice for training linear factorization blocks is to apply Euclidean optimizers, most commonly AdamW Loshchilov & Hutter (2019), directly to the unconstrained pair (A, B), ignoring the underlying geometry entirely.

The natural geometric remedy is to optimize on the fixed-rank matrix manifold

$$\mathcal{M}_r = \mathbb{R}_*^{r\times n} \times \mathbb{R}_*^{m\times r}/\mathrm{GL}(\mathbb{R}^r),$$

viewed as a quotient Mishra et al. (2014). Here, $\mathbb{R}_*^{m\times r}$ denotes the set of full-rank matrices of size $m \times r$. While geometrically clean, this approach is at odds with adaptive optimizers such as Adam, since the full $\mathrm{GL}(\mathbb{R}^r)$ invariance forces nonlinear adaptive updates to satisfy strong equivariance constraints that, in general, cannot be reconciled with the coordinate-wise diagonal preconditioning that gives AdamW much of its practical strength. As a result, prior Riemannian adaptive methods either give up coordinate-wise adaptivity in favor of scalar preconditioners, or modify the gradient structure rather than the preconditioner itself Bécigneul & Ganea (2019); Sakai & Iiduka (2025); Schotthöfer et al. (2025a); Bian et al. (2025).

In this work, we adopt a different trade-off. Rather than enforcing full quotient invariance, we relax it to an orthogonal symmetry by restricting to the product manifold $\mathrm{St}(n, r) \times \mathbb{R}_*^{m\times r}$, on which Φ is invariant only under $\mathrm{O}(r)$, i.e., $\Phi(A, B) = \Phi(Q^\top A,\, BQ),\ \forall Q \in \mathrm{O}(r)$. This relaxation accomplishes two things at once. *Geometrically*, it makes the fibers of Φ compact, which rules out the kind of factor blow-up illustrated above and stabilizes training under aggressive learning rates. *Algorithmically*, because the constrained factor lives on an embedded submanifold and the other factor in a flat Euclidean space, we can perform all moment accumulation in the ambient space and only project onto the tangent space of the Stiefel manifold immediately before the update. This preserves AdamW's coordinate-wise diagonal preconditioning essentially unchanged.

The resulting algorithm, **Stiefel-AdamW**, is best understood as a small, structural modification of AdamW. The Euclidean factor is updated by the standard AdamW step; the Stiefel factor is updated by computing the first and second moments in the ambient space, projecting the preconditioned direction onto the tangent space, and retracting back to the manifold. The retraction step is treated as a modular component: any efficient retraction on the Stiefel manifold can be used, including the Cayley transform, QR-based retraction, polar decomposition, or Newton–Schulz iteration Kovarik (1970); Björck & Bowie (1971); Higham (2008). In our experiments, the choice of retraction has only a marginal effect on final performance, with the best option mildly model- and problem-dependent; see Appendix B.1 and Table 3 for a comparison. The overall implementation overhead over standard AdamW is minimal, yet the resulting optimizer inherits the principal benefits of geometric optimization: numerical stability, parameter invariance to orthogonal reparametrization, low memory footprint, and provable convergence under standard assumptions.

We emphasize that Stiefel-AdamW is not a method specific to LoRA or to any particular architectural family. It is a simple, near drop-in replacement for AdamW that can (and should) be used on *any* factorization block of the form $W = BA$ in which the two factors are not separated by a nonlinearity, regardless of where such a block appears in the model. This includes LoRA adapters, low-rank compressed layers, and query/key projections in attention blocks. In all these cases, applying Stiefel-AdamW rather than vanilla AdamW replaces a Euclidean parametrization with hidden gauge symmetry by one in which (most of) the symmetry has been quotiented out, and does so essentially for free.

Contributions. Our contributions are as follows.

1. We identify factorization blocks $W = BA$ ubiquitous in modern deep-learning architectures (LoRA, low-rank compressed layers, attention, SSMs) as a unifying setting in which geometric optimization can be applied transparently, and we propose Stiefel-AdamW, a near drop-in replacement for AdamW that should be used wherever such blocks appear.
2. We propose a structured relaxation of the full $\mathrm{GL}(\mathbb{R}^r)$ quotient invariance to orthogonal invariance, working on the product manifold $\mathrm{St}(n, r) \times \mathbb{R}_*^{m\times r}$. This is precisely the relaxation that allows full coordinate-wise Adam-style adaptive preconditioning to coexist with geometry-aware updates, in contrast to prior Riemannian Adam-type methods that rely on simple scalar preconditioners.

3. We design the algorithm so that all adaptive moment estimation is performed in ambient Euclidean space, with geometric corrections (tangent projection and retraction) applied only at the update step. As a consequence, the per-step cost and memory footprint are essentially those of AdamW.
4. We treat the Stiefel retraction as a modular, plug-and-play component: any efficient retraction (Cayley transform, QR, polar decomposition, Newton–Schulz) can be used, with only marginal differences in practice. Approximate retractions, when second-order accurate, remain compatible with the standard convergence theory.
5. We establish theoretical guarantees of boundedness of gradients and convergence of the regret function under standard assumptions.
6. We validate Stiefel-AdamW empirically across a range of representative tasks, from LoRA-style fine-tuning (GPT2, ViT, Mistral 7B) to full LLM pretraining (GPT2 on OpenWebText), demonstrating that the additional cost over AdamW is minimal while consistently improving stability and final performance.

2 Related Work

Riemannian Optimization. Riemannian gradient methods on embedded and quotient manifolds, including the Stiefel and Grassmann manifolds, have a long history in numerical optimization Luenberger (1972); Gabay & Luenberger (1976); Edelman et al. (1998); Yang (2007), and have since been extended to trust-region Absil et al. (2007), quasi-Newton Ring & Wirth (2012), and conjugate-gradient Sato & Iwai (2015); Sato (2016; 2022) methods. Key tools including retraction mappings Absil & Malick (2012); Absil & Oseledets (2015), efficient preconditioners Vandereycken & Vandewalle (2010); Boumal & Absil (2015), and quotient geometries Mishra et al. (2014) have been thoroughly developed; comprehensive treatments can be found in Absil et al. (2008); Sato (2021); Boumal (2023).

Adaptive Methods and their Riemannian Extensions. In deep learning, adaptive optimizers such as AdaGrad Duchi et al. (2011), RMSProp Hinton et al. (2012), Adam Kingma & Ba (2015), AMSGrad Reddi et al. (2018), and AdamW Loshchilov & Hutter (2019) are the de facto standard, combining fast convergence, robustness, and low memory overhead. Extending them to manifold-constrained settings is nontrivial, because adaptive moments and preconditioning must respect the underlying geometry. Stochastic Riemannian optimization began with Riemannian SGD Bonnabel (2013) and variance-reduced variants Zhang et al. (2016); Kasai et al. (2018). More recent work has proposed Riemannian AdaGrad and AMSGrad Bécigneul & Ganea (2019), modified AMSGrad schemes Sakai & Iiduka (2022), RASA Kasai et al. (2019), and Riemannian adaptive gradient methods with theoretical guarantees Bian et al. (2025); Sakai & Iiduka (2025). A common limitation of these approaches is that they rely on scalar or geometry-compatible preconditioners; the fully coordinate-wise diagonal preconditioning central to AdamW is generally incompatible with strict manifold invariance, which is the gap Stiefel-AdamW is designed to close.

Linear Factorization Blocks in Deep Learning. Matrix factorization blocks of the form $W = BA$, in which two parameter matrices multiply directly without an intervening nonlinearity, appear throughout modern architectures. They arise explicitly in LoRA-style parameter-efficient fine-tuning Hu et al. (2022) and its variants Hayou et al. (2024); Zhang & Pilanci (2024); Zhu et al. (2024); Wang et al. (2025); Schotthöfer et al. (2025b), in network compression Vogels et al. (2019); Saha et al. (2023); Schotthöfer et al. (2022), and implicitly inside standard architectures such as the query-key product in self-attention Vaswani et al. (2017). The non-uniqueness of such factorizations and its consequences for training stability have been studied through dynamical low-rank approximation Koch & Lubich (2007); Hnatiuk et al. (2026); Zangrando et al. (2024) and quotient-manifold optimization Mishra et al. (2014). Methods such as GeoLoRA Schotthöfer et al. (2025b) have explicitly exploited Grassmannian geometry in low-rank adaptation, and recent work has combined momentum and adaptivity with Riemannian updates Schotthöfer et al. (2025a); Sakai & Iiduka (2025).

Relation to Prior Work. The closest methods to Stiefel-AdamW are GeoLoRA Schotthöfer et al. (2025b) and RAdam Bécigneul & Ganea (2019) or its Stiefel-specific version, Cayley Adam Li et al. (2020). GeoLoRA enforces geometry on the full factorization via a quotient-manifold formulation,

Algorithm 1 Single iteration of Stiefel-AdamW.

Require: A_t with $A_t A_t^\top = I_r$, B_t, M_{t-1}^A, M_{t-1}^B, V_{t-1}^A, V_{t-1}^B, η_t, β_1, β_2, ε

$G_t^B \leftarrow \nabla_B \mathcal{L}(B_t A_t)\, A_t^\top$ ▷ Euclidean gradient w.r.t. B
$G_t^A \leftarrow B_t^\top \nabla_A \mathcal{L}(B_t A_t)$ ▷ Euclidean gradient w.r.t. A

$M_t^B \leftarrow \beta_1 M_{t-1}^B + (1-\beta_1)\, G_t^B$ ▷ First moment, B
$M_t^A \leftarrow \beta_1 M_{t-1}^A + (1-\beta_1)\, G_t^A$ ▷ First moment, A
$V_t^B \leftarrow \beta_2 V_{t-1}^B + (1-\beta_2)\, (G_t^B)^{\circ 2}$ ▷ Second moment, B
$V_t^A \leftarrow \beta_2 V_{t-1}^A + (1-\beta_2)\, (G_t^A)^{\circ 2}$ ▷ Second moment, A
$B_{t+1} \leftarrow B_t - \eta_t \Big(M_t^B / \Big(\sqrt{V_t^B + \varepsilon} \Big) + \lambda B_t \Big)$ ▷ Standard AdamW step on B
$X_t \leftarrow A_t^\top$ ▷ Column convention, $X_t \in \mathrm{St}(n, r)$
$D_t \leftarrow M_t^A / (\sqrt{V_t^A + \varepsilon})$ ▷ Preconditioned direction
$\xi_t \leftarrow -\eta_t\, \mathrm{P}_{X_t}(D_t)$ ▷ Project onto $\mathrm{T}_{X_t}\mathrm{St}(n,r)$, $\mathrm{P}_X(Z) = X\,\mathrm{skew}(X^\top Z) + (I - XX^\top)Z$
$X_{t+1} \leftarrow \mathrm{Retr}_{X_t}(\xi_t)$, $A_{t+1} \leftarrow X_{t+1}^\top$ ▷ Retract to $\mathrm{St}(n,r)$; see Section A.2.3 for details on the computational costs.

which precludes coordinate-wise adaptive preconditioning. RAdam defines a Riemannian Adam on the Stiefel manifold but uses a scalar preconditioner, departing from AdamW's diagonal adaptivity. In contrast, Stiefel-AdamW constrains only one factor to the Stiefel manifold while leaving the other in Euclidean space. This product-manifold structure is precisely what makes full coordinate-wise adaptive preconditioning tractable: moment accumulation is performed in the ambient space for both factors, and geometry enters only through one factor via a tangent-space projection followed by a retraction on the Stiefel manifold. The result is an optimizer that inherits the stability benefits of Riemannian methods and the practical performance of AdamW, at minimal additional cost.

3 The Proposed Method: Stiefel-AdamW

3.1 Problem Setup

We consider a trainable linear factorization block of the form $W = BA$, where $B \in \mathbb{R}^{m\times r}$ and $A \in \mathbb{R}^{r\times n}$. In order to avoid potential instabilities due to non-uniqueness of this representation, we impose a row-orthonormality constraint on A, namely $AA^\top = I_r$, i.e., we require $A^\top$ to lie on the Stiefel manifold $\mathrm{St}(n,r) = \{X \in \mathbb{R}^{n\times r} : X^\top X = I_r\}$, while leaving B unconstrained. This restriction reduces the invariance group from $\mathrm{GL}(\mathbb{R}^r)$ to the compact group $\mathrm{O}(r)$, making the fibers of Φ compact and ruling out the factor blow-up illustrated in Section 1; see also Section 4.2 for further details. The resulting product-manifold structure $\mathrm{St}(n,r) \times \mathbb{R}^{m\times r}$ enables an efficient fully coordinate-wise adaptive update: moment accumulation is performed in the ambient Euclidean space for both factors, with geometric corrections entering only through a tangent-space projection and a retraction onto the Stiefel manifold for the A factor.

3.2 Description of the Algorithm

In this section, we describe one iteration of Stiefel-AdamW; the pseudocode is given in Algorithm 1. For simplicity of exposition, we omit bias correction and explicit weight decay; both can be incorporated straightforwardly as in AdamW.

Let $W_t \in \mathbb{R}^{m\times n}$ be a weight matrix at iteration t, represented by the factorization $W_t = B_t A_t$, where $A_t \in \mathbb{R}^{r\times n}$ and $B_t \in \mathbb{R}^{m\times r}$. The objective function is evaluated as $\mathcal{L}(B_t A_t)$. Next, the algorithm computes the Euclidean gradients $G_t^B = \nabla_B \mathcal{L}(B_t A_t)$ and $G_t^A = \nabla_A \mathcal{L}(B_t A_t)$. As in Adam, Stiefel-AdamW then computes the first and second moments for both factors, i.e.,

$$\begin{cases} M_t^B = \beta_1 M_{t-1}^B + (1-\beta_1)\, G_t^B, \\ M_t^A = \beta_1 M_{t-1}^A + (1-\beta_1)\, G_t^A, \end{cases} \qquad \begin{cases} V_t^B = \beta_2 V_{t-1}^B + (1-\beta_2)\, (G_t^B)^{\circ 2}, \\ V_t^A = \beta_2 V_{t-1}^A + (1-\beta_2)\, (G_t^A)^{\circ 2}, \end{cases}$$

where $^{\circ 2}$ denotes elementwise squaring.

Up to this step, both factors are treated in the same way; however, the subsequent update steps do differ. Indeed, since B_t is unconstrained, the algorithm performs the usual AdamW update, i.e., $B_{t+1} = B_t - \eta_t\Big(M_t^B/(\sqrt{V_t^B + \varepsilon}) + \lambda B_t\Big)$, where η_t is the learning rate, / indicates elementwise division, the square root is also meant to be performed componentwise, and the $\varepsilon > 0$ is a small constant to avoid blowup of the metric. This factor requires no Riemannian machinery, which keeps the method simple and efficient.

For the orthonormal factor, we employ a retraction-based Riemannian update. For convenience, we switch to a column-orthonormal representation by defining $X_t \coloneqq A_t^\top \in \mathbb{R}^{n\times r}$, so that $X_t \in \mathrm{St}(n, r)$, i.e., $X_t^\top X_t = I_r$. A key feature of Stiefel-AdamW is that it performs adaptive moment estimation in the ambient Euclidean space before projecting onto the tangent space. This allows us to use coordinate-wise preconditioning, as in AdamW. In contrast, many existing Riemannian adaptive methods restrict the preconditioner to be scalar or geometry-compatible to preserve invariance, thereby limiting their practical effectiveness. More precisely, the Euclidean adaptive direction is first formed as $D_t = M_t^A/(\sqrt{V_t^A + \varepsilon})$, and then projected onto the tangent space $\mathrm{T}_{X_t}\mathrm{St}(n, r)$ to obtain the direction $\xi_t = -\eta_t \mathrm{P}_{X_t}(D_t)$, where $\mathrm{P}_{X_t}\colon \mathbb{R}^{n\times r} \to \mathrm{T}_{X_t}\mathrm{St}(n, r)$ is the orthogonal projection onto the tangent space to $\mathrm{St}(n, r)$ at X_t, $\mathrm{P}_X(Z) = X\mathrm{skew}(X^\top Z) + (I - XX^\top)\, Z$, with $\mathrm{skew}(M) = (M - M^\top)/2$. See Appendix A.1 for more details on the geometry of the Stiefel manifold.

Finally, to map a tangent vector ξ_t back onto the manifold, we need to apply a retraction mapping, $X_{t+1} = \mathrm{Retr}_{X_t}(\xi_t)$. In the right part of Table 3, we present numerical results comparing different kinds of retractions. In the remaining numerical experiments, we use the Cayley retraction as the standard choice, approximated via fixed-point iteration, because of its simplicity of implementation and good performance in the comparison tests. Moreover, our analysis shows that the framework is robust to *approximate* retractions: as established in Theorem 4.2, the introduction of a maximal Frobenius error δ in the retraction mapping merely adds a manageable linear term to the regret bound. This theoretical guarantee justifies using truncated or iterative retraction methods that can run for only a few iterations without reaching machine precision, while offering speedups. More details on computational costs of other common choices of retraction can be found in Section A.2.3.

4 Theoretical Guarantees

4.1 Regret Analysis

In this section, we present a convex regret analysis for Algorithm 1. Regret analysis is a standard tool in convex optimization that quantifies how much an optimization algorithm, when running dynamically on a family of convex objective functions $\mathcal{L}_t$, is suboptimal with respect to the optimal objective ahead of time. In particular, given a sequence of iterates $\{W_t\}_{t=1,\dots,T}$, we define the regret function as

$$R(T) \coloneqq \sum_{t=1}^{T} \mathcal{L}_t(W_t) - \min_{W} \sum_{t=1}^{T} \mathcal{L}_t(W).$$

We recall that an algorithm is said to be zero regret if $R(T)/T \to 0$ as $T \to +\infty$. Our aim is to show that Algorithm 1 indeed produces arbitrarily small regret for a small enough learning rate and retraction error. To prove this result, we will make the following assumptions:

Assumptions 4.1 (Setting of regret analysis)**.**

(H1) The family $\mathcal{L}_t\colon \mathbb{R}^{m\times r} \times \mathrm{St}(n, r) \to \mathbb{R}$ is the restriction of a family of Euclidean strictly convex functions defined on $\mathbb{R}^{r\times m} \times \mathbb{R}^{r\times n}$. By a small abuse of notation, we will also denote the extension family with $\mathcal{L}_t$, and we will denote the minimizer with (B^*, A^*).

(H2) The first momentum coefficients $\beta_{1,t} = \beta_1 b^t$ decrease geometrically in time for a constant $0 < b < 1$, and with $\beta_1 < \sqrt{\beta_2}$.

(H3) The iterates B_t stay bounded, i.e., $\sup_t \|B_t\|_{\max} \le D_\infty$.

(H4) The Euclidean gradient $\nabla\mathcal{L}_t(B_t, A_t)$ stays bounded, i.e., $\sup_t \|\nabla\mathcal{L}_t(B_t, A_t)\|_{\max} \le G_\infty$.

(H5) The second momentum update in Algorithm 1 is followed by an entrywise maximum, i.e., $V_{t+1} = \max(\beta_2 V_{t-1} + (1-\beta_2)G_t^2, V_{t-1})$, as AMSGrad (Reddi et al., 2018, Algorithm 2).

(H6) The projected direction is aligned with the globally correct direction, $\langle \xi_t, A^* - A_t \rangle \geq 0$.

We emphasize that Assumptions 4.1 are fairly standard assumptions used to study convergence of Adam-like algorithms, and they were already employed in, e.g., Reddi et al. (2018). Assumption (H6) is a hypothesis often used in Euclidean optimizers to ensure that the current local descent direction is aligned with the global direction to the minimizer.

Theorem 4.2. *(Regret bound) Under Assumptions 4.1, consider the sequence of iterates produced by Algorithm 1 with decreasing learning rates $\eta_t = \eta/\sqrt{t}$, and no weight decay. Then,*

$$R(T) \leq C_1 + C_2\sqrt{T} + C_3\sqrt{1+\log T} + C_4 \log T + C_5 T^{-1/2}, \tag{1}$$

where C_1, C_2, C_3, C_4, C_5 are constants independent of T. In particular, $\lim_{T\to+\infty} R(T)/T = 0$.

We note that, although the theoretical result requires an AMSGrad-like assumption (H5), in practice the algorithm can be used without the $\max$ update with no loss in performance. The proof of Theorem 4.2 can be found in Appendix B.

In most practical implementations, the retraction is computed only approximately via a numerical algorithm, such as the fixed-point method used in most of our experiments; see also Appendix A.2.2. The proof of Theorem 4.2 above extends straightforwardly to that case: assuming that the computed retraction has an error of $\delta > 0$, then a term $C_6\delta$ has to be added to equation 1, without significantly affecting the main result of the theorem.

4.2 Gradient Boundedness and Stability to Large Learning Rates

Working with an orthonormal factor yields a method with bounded gradients, potentially improving stability at large learning rates. Here, we make this point more concrete with an example. Let us consider the rank-r recovery problem in $\mathbb{R}^{n\times n}$, $\mathcal{L}(A,B) = \frac{1}{2}\|BA - \alpha I_n\|_{\mathrm{F}}^2$, $B \in \mathbb{R}^{n\times r}$, $A \in \mathbb{R}^{r\times n}$, trained by plain gradient descent from the canonical LoRA initialization $B_0 = 0$, A_0 arbitrary. The first GD step gives $B_1 = \eta\alpha A_0^\top$ and $A_1 = A_0$, so for the scaling $\alpha = 1/\eta$ the two factors align already after one step. Specifically, we have $B_1 = A_1^\top$ and a direct induction argument shows that the iterates preserve $B_k = A_k^\top$ thereafter. Setting $X_k := B_k = A_k^\top$, the dynamics take the form $X_{k+1} = X_k - \eta(X_k X_k^\top - \eta^{-1} I)X_k = (2I - \eta X_k X_k^\top)X_k$, thus, using the SVD $X_k = U_k \Sigma_k V_k^\top$, the dynamics decouple across singular values into the scalar recursion $\sigma_{i,k+1} = \sigma_{i,k}\big(2 - \eta\,\sigma_{i,k}^2\big)$, which diverges as soon $|2 - \eta\,\sigma_{i,k}^2| > 1$, i.e., when some $\sigma_{i,k} > \sqrt{3/\eta}$. The largest stable learning rate is therefore dictated by the largest singular value of the iterate, a property of the *parameterization*, not of the underlying optimization landscape.

If instead the constraint $A_k A_k^\top = I_r$ is enforced, the gradient descent update on B simplifies to the affine recursion $B_{k+1} = (1-\eta)\,B_k + \eta\alpha\,A_k^\top$, which is bounded for every $0 < \eta < 2$ as $\|A_k\| = 1$, regardless of singular-value scale, removing the dependence of the stable learning rate on the iterate. This phenomenon is well documented in the Riemannian optimization literature Schotthöfer et al. (2025a), and we provide a more precise result in the next Proposition 4.3; see Appendix C for the proof.

Proposition 4.3 (Gradient boundedness on fibers)**.** *Consider the maps $\Phi\colon \mathbb{R}^{m\times r} \times \mathbb{R}^{r\times n} \to \mathbb{R}^{m\times n}$ given by $\Phi(B,A) = BA$ and the map $\widetilde{\Phi} = \Phi|_{\mathrm{St}(n,r)\times\mathbb{R}^{m\times r}}$. Let $W \in \mathbb{R}^{m\times n}$ be a fixed matrix with $\mathrm{rank}(W) \leq r$ such that $\nabla\mathcal{L}(W) \neq 0$. Let $\mathcal{F} := \Phi^{-1}(W)$ and $\widetilde{\mathcal{F}} := \widetilde{\Phi}^{-1}(W)$. Then,*

$$\|\nabla(\mathcal{L}\circ\Phi)\|_{L^\infty(\mathcal{F})} = +\infty, \quad \|\nabla(\mathcal{L}\circ\widetilde{\Phi})\|_{L^\infty(\widetilde{\mathcal{F}})} < +\infty.$$

In particular, it is known that the boundedness of $\nabla(\mathcal{L}\circ\Phi)$ is closely related to the range of stable learning rates. This suggests that optimization algorithms on the parameterization $\widetilde{\Phi}$ are more stable than the ones on the more redundant representation Φ.

5 NUMERICAL EXPERIMENTS

In this section, to show the effectiveness and scalability of Stiefel-AdamW, we present several numerical experiments for both fine-tuning pretrained models with LoRA adapters (Hu et al., 2022) and LLM pretraining. We compare against six baselines: standard **AdamW** (Loshchilov & Hutter, 2019); **Scaled AdamW** (Zhang & Pilanci, 2024), which introduces a coupled preconditioner accounting for the product structure; **GeoLoRA** (Schotthöfer et al., 2025b), which uses two Stiefel representations; **LoRA-RITE** (Yen et al., 2025) and **LoRA-Pro** (Wang et al., 2025), which modify the gradient structure to reduce sensitivity to the non-uniqueness of the factorization; and **Cayley Adam** (Li et al., 2020), which uses the Riemannian Adam variant proposed in (Bécigneul & Ganea, 2019) on the Stiefel manifold. We emphasize that the latter method uses a scalar preconditioner, whereas Stiefel-AdamW uses a diagonal preconditioner.

5.1 LORA FINE-TUNING

GPT2 In this experiment, we tested Stiefel-AdamW for fine-tuning GPT2 on the E2E Natural Language Generation challenge (Novikova et al., 2017) with LoRA of rank 4. We report the results in Table 1. In all experiments, we trained the models for 5 epochs with a batch size of 8. For details on the hyperparameter settings, see Table 4. As shown in Table 3, Stiefel-AdamW outperforms all baselines across all tasks except ROUGE-L. Interestingly, we observe that, consistently with the findings of Zhang & Pilanci (2024) for Scaled AdamW, Stiefel-AdamW achieves better performance when using more aggressive moving average parameters β_1, β_2. This suggests that part of AdamW's update may be spent along invariant directions, while Riemannian-informed approaches avoid this and allow greater emphasis to be placed on the current gradient direction. All numerical experiments were performed on a single NVIDIA A100 80GB, except for GPT2 pretraining, which was performed on two NVIDIA H100 80GB via Modal.

Table 1: Fine-tuning performance with low-rank adapters. Best results highlighted in bold. **Left:** GPT2 on the E2E Natural Language Generation challenge, with rank = 4. AdamW and Scaled AdamW are reported from Zhang & Pilanci (2024). **Right:** ViT-Base on CIFAR-10, for three different choices of adapter rank. We report with $\pm\sigma$ the standard deviation over 5 random initializations.

Method	BLEU	NIST	MET	ROUGE-L	CIDEr
AdamW Loshchilov & Hutter (2019)	68.41 ± 0.4950	8.65 ± 0.04	46.38 ± 0.12	71.13 ± 0.17	2.51 ± 0.001
Scaled AdamW Zhang & Pilanci (2024)	**69.17** ± 0.43	8.72 ± 0.058	46.44 ± 0.16	**71.57** ± 0.235	2.51 ± 0.005
Stiefel-AdamW	69.20 ± 0.964	**8.74** ± 0.102	**46.48** ± 0.259	71.40 ± 0.44	**2.51** ± 0.02
GeoLoRA Schotthöfer et al. (2025a)	68.11 ± 0.271	8.60 ± 0.07	45.82 ± 0.24	70.47 ± 0.388	2.41 ± 0.02
LoRA-RITE Yen et al. (2025)	68.98 ± 1.03	8.69 ± 0.11	46.35 ± 0.219	71.17 ± 0.410	2.48 ± 0.05
LoRA-Pro Wang et al. (2025)	68.12 ± 0.25	8.61 ± 0.06	45.82 ± 0.250	70.46 ± 0.382	2.42 ± 0.02
Cayley Adam Li et al. (2020)	68.97 ± 1.04	8.69 ± 0.11	46.36 ± 0.215	71.16 ± 0.415	2.48 ± 0.05

	Rank		
	32	64	128
	95.6 ±0.2	95.55 ±0.15	95.82 ±0.29
	92.25 ±0.28	94.83 ±0.31	95.30 ±0.12
	95.91 ±0.19	**96.04** ±0.15	**96.41** ±0.10
	95.53 ±0.44	95.12 ±0.37	95.27 ±0.23
	94.55 ±0.32	94.98 ±0.10	94.89 ±0.24
	90.17 ±2.30	94.37 ±0.26	94.32 ±0.21
	95.44 ±0.92	95.71 ±0.17	96.12 ±0.11

Vision Transformers In this experiment, we fine-tuned the base Vision Transformer from Dosovitskiy et al. (2021) on CIFAR-10 Krizhevsky & Hinton (2009), with results shown in the right part of Table 1. In Figure 1, we compare loss descent and time per iteration against the best loss achieved. All models have been trained for 50 epochs with a batch size of 64, LoRA alpha 32, learning rate 10^{-3}, and no scheduler. For all optimizers, we used weight decay of 10^{-5} on all adapters, applied to the key-query attention matrices, attention projection, and the last two fully connected layers. We did not optimize biases and left them as in the pretrained model. As we can observe from the results in the right panel of Table 3 and Figure 1, Stiefel-AdamW is able to outperform all baselines in terms of performance, with convergence speed comparable to that of Scaled AdamW (Zhang & Pilanci, 2024) and AdamW (Loshchilov & Hutter, 2019).

Mistral 7B In this experiment, we tested the effectiveness of Stiefel-AdamW for fine-tuning Mistral 7B (Jiang et al., 2023) on the GLUE benchmark (Wang et al., 2019) for natural language understanding, following the implementation in (Zhang & Pilanci, 2024). LoRA adapters of rank 16 have been applied to all query, key, value projection, and gate matrices of multihead attention. We did not train biases, and for all optimizers, we set the LoRA alpha learning-rate scaling parameter to 16. We used mixed precision for all optimizers: the base model was loaded in its 4-bit quantized version, optimizer states were in float32, and operations were performed in mixed-precision bfloat16. We trained all models using the codebase of Zhang & Pilanci (2024), with a dropout of 0.1 and

Figure 1: **Left and center:** Loss function descent for ViT Base on CIFAR-10 LoRA fine-tuning (LoRA rank 64 and learning rate 10^{-3}). **Right:** Learning rate versus best training loss for different optimizers when fine-tuning ViT Base on CIFAR-10. In this experiment, we fixed the LoRA rank at 32 and trained all models for 50 epochs.

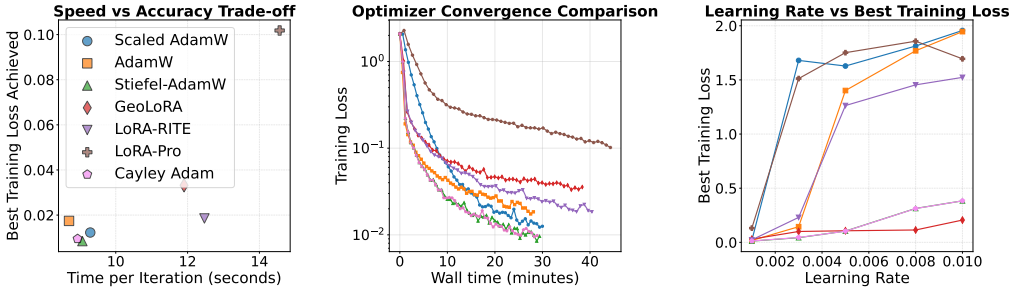

Table 2: Scores for rank 16 LoRA fine-tuning of the 4-bit quantized Mistral 7B model on the GLUE benchmark for Natural Language Understanding (NLU) challenges with different optimizers (best results highlighted in bold). In parentheses, we report, for each task, the percentage deviation from the best performer. SGD, Scaled GD, and Scaled AdamW results reported from (Zhang & Pilanci, 2024, Table 2).

Method	**GLUE**								
	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B	WNLI	Avg.
SGD	88.15 (-4.14%)	96.10 (-1.18%)	70.10 (-22.07%)	55.89 (-22.23%)	94.22 (-1.21%)	88.59 (-3.94%)	50.90 (-44.27%)	47.64 (-48.37%)	49.30 (-43.54%)	71.21
Scaled GD Zhang & Pilanci (2024)	90.21 (-1.89%)	96.90 (-0.36%)	81.62 (-9.26%)	68.17 (-5.14%)	94.40 (-1.01%)	91.15 (-1.16%)	54.15 (-40.72%)	90.31 (-2.13%)	56.34 (-35.48%)	80.36
AdamW	91.64 (-0.34%)	**97.25**	87.01 (-3.27%)	**71.87**	94.79 (-0.61%)	91.81 (-0.44%)	90.25 (-1.19%)	90.51 (-1.92%)	85.91 (-1.61%)	89.00
Scaled AdamW Zhang & Pilanci (2024)	90.68 (-1.38%)	**97.25**	89.46 (-0.55%)	71.30 (-0.79%)	94.67 (-0.73%)	**92.22**	**91.34**	91.10 (-1.28%)	83.10 (-4.83%)	89.01
Stiefel-AdamW	**91.95**	96.79 (-0.47%)	**89.95**	70.61 (-1.75%)	94.78 (-0.62%)	91.83 (-0.42%)	90.61 (-0.80%)	**92.28**	**87.32**	**89.57**
GeoLoRA Schotthöfer et al. (2025b)	91.30 (-0.71%)	94.61 (-2.71%)	87.26 (-2.99%)	69.78 (-2.91%)	**95.37**	90.80 (-1.54%)	88.81 (-2.77%)	91.45 (-0.90%)	**87.32**	88.52

a batch size of 8 across all models. For AdamW, Scaled AdamW, and GeoLoRA, we used the optimal hyperparameters (learning rate and β_1, β_2) from Zhang & Pilanci (2024); for Stiefel-AdamW, we used $\beta_1 = \beta_2 = 0.95$. For each GLUE task, we report the value of the standard test metric (either accuracy or correlation) and the percentage deviation from the best performer among all optimizers. As shown in Table 2, Stiefel-AdamW outperforms all baselines in terms of average score. On single tasks, Stiefel-AdamW outperforms all baselines on MNLI, MRPC, STS-B, and WNLI, while maintaining a competitive performance on all other tasks.

5.2 GPT2 Pretraining

As the proposed method works for all problems in which the manifold of rank-r matrices appears, it also applies directly to pretraining transformer-based architectures. In particular, self-attention naturally respects this structure, as the image of the map $(W_Q, W_K) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{n\times r} \mapsto W_Q W_K^\top$ is exactly the set $\mathcal{M}_r$. Despite this, the parametrization map is highly non-injective, and therefore the problem could be restated equivalently by minimizing on $\mathcal{M}_r \cong \mathrm{St}(n,r) \times \mathbb{R}^{n\times r}_*/\mathrm{O}(r)$ instead of $\mathbb{R}^{n\times r} \times \mathbb{R}^{n\times r}$. In contrast, we optimize all other parameters, such as biases or non-structured matrices, with the standard AdamW step. In Table 3 we present the results for pretraining GPT2 (Radford et al., 2019) on OpenWebText (Gokaslan & Cohen, 2019) using Karpathy's reproduction[1]. We reproduced the pretraining for Stiefel-AdamW using exactly the same AdamW hyperparameters from the repository. We trained both AdamW and Stiefel-AdamW for 7000 iterations. As shown, Stiefel-AdamW produces results comparable to standard AdamW (Loshchilov & Hutter, 2019) in both performance and peak GPU memory usage. We performed no hyperparameter tuning, and Stiefel-AdamW uses the same hyperparameters as AdamW, as shown in the reproduced repository.

[1] https://github.com/karpathy/nanoGPT

Table 3: **Top Left**: Full GPT2 pretraining on OpenWeb Text for 6000 iterations. **Bottom Left**: Qwen2 pretraining on Wikitext-103, standard deviation reported over 5 random seeds. **Right:** Ablation over different retractions on ViT Base.

Method	**Test Loss** $\pm\sigma$	**Peak Memory** (GB)
AdamW	3.260 ± 0.05	13.8
Stiefel-AdamW	$\mathbf{3.236} \pm 0.07$	13.8

Method	**Loss** $\pm\sigma$
AdamW	2.66 ± 0.05
Stiefel-AdamW	$\mathbf{2.59} \pm 0.004$

Retraction	**Test acc.** ($r = 32$)	**Test acc.** ($r = 64$)	**Test acc.** ($r = 128$)
Cayley FP	96.19	96.25	96.45
Cayley SMW	96.11	96.09	96.52
Cayley direct	96.11	96.22	96.61
QR	95.41	94.83	93.71
Polar	96.11	96.22	96.61
Newton–Schulz	96.11	96.20	96.52

This experiment highlights that Stiefel-AdamW is not limited to fine-tuning scenarios such as LoRA, but can be applied directly in pretraining settings where low-rank structure arises naturally.

5.3 QWEN2 PRETRAINING

To showcase compatibility of our proposed approach with the presence of positional embeddings such as RoPE Su et al. (2024), in Table 3 we also present numerical results for Qwen2 Yang et al. (2024) LoRA pretraining on WikiText-103 Merity et al. (2016). The model with the adapters has 100M parameters; it was trained for 20K steps with a batch size of 16 and two steps of gradient accumulation. As shown in Table 3, Stiefel-AdamW outperforms the Euclidean version of AdamW in both mean and variance.

5.4 STEPSIZE STABILITY

Motivated by Proposition 4.3, we show the stability of Stiefel-AdamW with respect to the learning rate, numerically demonstrating that methods with compact fibers are more stable with respect to learning-rate size.

We fine-tuned multiple vision transformers (ViT Base) on CIFAR-10 for different learning rates, keeping all other hyperparameters fixed as in Section 5.1. In the right panel of Figure 1, we plot the learning rate against the best loss obtained during training. As expected, the results in Figure 1 show that "pure" Riemannian methods such as GeoLoRA Schotthöfer et al. (2025b) and the proposed Stiefel-AdamW are more stable with respect to learning-rate magnitude.

In particular, Riemannian methods in which the fiber of Φ is not compact (such as AdamW and Scaled AdamW, which are defined in $\mathbb{R}^{r\times n} \times \mathbb{R}^{m\times r}$), appear to be less stable with respect to larger learning rates, despite the preconditioning (i.e., different metric) employed in Scaled AdamW Zhang & Pilanci (2024).

6 CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

In this work, we presented Stiefel-AdamW, a stochastic Riemannian variant of AdamW that naturally provides convergence guarantees. The compactness of the space in which one factor lives helps avoid potential numerical instabilities that can arise from unbalanced initializations. One key advantage of the proposed method is its simplicity of implementation, which requires only minimal machinery from Riemannian optimization theory while maintaining guarantees.

We demonstrated the method's effectiveness and scalability across a range of problems, from fine-tuning to pretraining LLMs. One limitation of the current method is that it still retains a set of orthogonal invariances, which stems from the entrywise nature of the Adam algorithm. While this can be solved by imposing a gauge condition and working on horizontal spaces on the quotient space $\mathcal{M}_r$, it requires additional computational effort, as the gradient needs to be computed using joint information on the pair (B, A), $(\nabla_B \mathcal{L}, \nabla_A \mathcal{L})$, and therefore does not allow for full parallelization. Future research could extend the current method to different preconditioners and propose a similarly simple version that is fully invariant on the manifold of fixed-rank matrices and can partially mitigate this extra computational effort.

AI USE STATEMENT

In this work, we used generative AI tools to polish writing. We have not used generative AI tools for producing theoretical results or to aid for numerical implementation, and the rest of the required disclosure tasks are not applicable to this work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

P.-A. Absil and J. Malick. Projection-like Retractions on Matrix Manifolds. *SIAM J. Optim.*, 22(1): 135–158, 2012. doi: 10.1137/100802529.

P.-A. Absil and I. V. Oseledets. Low-rank retractions: a survey and new results. *Computational Optimization and Applications*, 62(1):5–29, Sep 2015. ISSN 1573-2894. doi: 10.1007/s10589-014-9714-4. URL https://doi.org/10.1007/s10589-014-9714-4.

P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Found. of Comput. Math.*, 7:303–330, 2007. ISSN 1572-9036. doi: 10.1007/s10208-005-0179-9.

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. ISBN 978-0-691-13298-3.

Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *International Conference on Learning Representations (ICLR 2019)*, volume 9, pp. 6384–6399, 2019.

Fengmiao Bian, Jinyang Zheng, Ziyun Liu, Jianzhou Luo, and Jian-Feng Cai. Finding Low-Rank Matrix Weights in DNNs via Riemannian Optimization: RAdaGrad and RAdamW. In *The 39th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=tiGFiCrmKm.

Å. Björck and C. Bowie. An Iterative Algorithm for Computing the Best Estimate of an Orthogonal Matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971. ISSN 00361429. URL http://www.jstor.org/stable/2949484.

Silvère Bonnabel. Stochastic Gradient Descent on Riemannian Manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013. doi: 10.1109/TAC.2013.2254619.

Nicolas Boumal. *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, 2023. doi: 10.1017/9781009166164.

Nicolas Boumal and P.-A. Absil. Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. *Linear Algebra and its Applications*, 475:200–239, 2015. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2015.02.027. URL https://www.sciencedirect.com/science/article/pii/S0024379515001342.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1998. doi: 10.1137/S0895479895290954. URL https://doi.org/10.1137/S0895479895290954.

Daniel Gabay and David G. Luenberger. Efficiently Converging Minimization Methods Based on the Reduced Gradient. *SIAM Journal on Control and Optimization*, 14(1):42–61, 1976. doi: 10.1137/0314004. URL https://doi.org/10.1137/0314004.

Aaron Gokaslan and Vanya Cohen. OpenWebText Corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 17783–17806, 2024.

Nicholas J. Higham. *Functions of Matrices*. Society for Industrial and Applied Mathematics, 2008. doi: 10.1137/1.9780898717778. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898717778.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural Networks for Machine Learning Lecture 6a Overview of mini-batch gradient descent. https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf, 2012.

Arsen Hnatiuk, Jonas Kusch, Lisa Kusch, Nicolas R. Gauger, and Andrea Walther. Stochastic Dynamical Low-Rank Approximation in the Context of Machine Learning. *Journal of Optimization Theory and Applications*, 208(1):1–33, January 2026. doi: 10.1007/s10957-025-02892-1. URL https://ideas.repec.org/a/spr/joptap/v208y2026i1d10.1007_s10957-025-02892-1.html.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825.

Hiroyuki Kasai, Hiroyuki Sato, and Bamdev Mishra. Riemannian Stochastic Recursive Gradient Algorithm. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2516–2524. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kasai18a.html.

Hiroyuki Kasai, Pratik Jawanpuria, and Bamdev Mishra. Riemannian adaptive stochastic gradient algorithms on matrix manifolds. In *International conference on machine learning*, pp. 3262–3271. PMLR, 2019.

Mikhail Khodak, Neil A. Tenenholtz, Lester Mackey, and Nicolo Fusi. Initialization and Regularization of Factorized Neural Layers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KTlJT1nof6d.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6980.

Othmar Koch and Christian Lubich. Dynamical Low-Rank Approximation. *SIAM Journal on Matrix Analysis and Applications*, 29(2):434–454, 2007. doi: 10.1137/050639703. URL https://doi.org/10.1137/050639703.

Zdislav V. Kovarik. Some Iterative Methods for Improving Orthonormality. *SIAM Journal on Numerical Analysis*, 7:386–389, 1970. URL https://api.semanticscholar.org/CorpusID:121623483.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Jun Li, Fuxin Li, and Sinisa Todorovic. Efficient Riemannian Optimization on the Stiefel Manifold via the Cayley Transform. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/2002.01113.

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. ReLoRA: High-Rank Training Through Low-Rank Updates. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DLJznSp6X3.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, 2019. URL https://arxiv.org/abs/1711.05101.

David G. Luenberger. The Gradient Projection Method along Geodesics. *Manage. Sci.*, 18(11): 620–631, 1972. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/2629156.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Bamdev Mishra, Gilles Meyer, Silvère Bonnabel, and Rodolphe Sepulchre. Fixed-rank matrix factorizations and Riemannian low-rank optimization. *Computational Statistics*, 29(3):591–621, Jun 2014. ISSN 1613-9658. doi: 10.1007/s00180-013-0464-z. URL https://doi.org/10.1007/s00180-013-0464-z.

Zhanfeng Mo, Long-Kai Huang, and Sinno Jialin Pan. Parameter and Memory Efficient Pretraining via Low-rank Riemannian Optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=i0zzO7Hslk.

Jekaterina Novikova, Ondrej Dušek, and Verena Rieser. The E2E Dataset: New Challenges for End-to-End Generation. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Saarbrücken, Germany, 2017. URL https://arxiv.org/abs/1706.09254. arXiv:1706.09254.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the Convergence of Adam and Beyond. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryQu7f-RZ.

W. Ring and B. Wirth. Optimization Methods on Riemannian Manifolds and Their Application to Shape Space. *SIAM J. Optim.*, 22(2):596–627, 2012. doi: 10.1137/11082885X. URL https://doi.org/10.1137/11082885X.

Rajarshi Saha, Varun Srivastava, and Mert Pilanci. Matrix Compression via Randomized Low Rank and Low Precision Factorization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=rxsCTtkqA9.

Hiroyuki Sakai and Hideaki Iiduka. Riemannian Adaptive Optimization Algorithm and its Application to Natural Language Processing. *IEEE Transactions on Cybernetics*, 52(8):7328–7339, 2022. doi: 10.1109/TCYB.2021.3049845.

Hiroyuki Sakai and Hideaki Iiduka. A general framework of Riemannian adaptive optimization methods with a convergence analysis. *Transactions on Machine Learning Research*, pp. n/a, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=knv4lQFVoE. Reproducibility Certification.

Hiroyuki Sato. A Dai–Yuan-type Riemannian conjugate gradient method with the weak Wolfe conditions. *Comput. Optim. Appl.*, 64(1):101–118, May 2016. ISSN 1573-2894. doi: 10.1007/s10589-015-9801-1. URL https://doi.org/10.1007/s10589-015-9801-1.

Hiroyuki Sato. *Riemannian Optimization and Its Applications*. Springer International Publishing, 2021. ISBN 978-3-030-62389-0. doi: https://doi.org/10.1007/978-3-030-62391-3.

Hiroyuki Sato. Riemannian Conjugate Gradient Methods: General Framework and Specific Algorithms with Convergence Analyses. *SIAM J. Optim.*, 32(4):2690–2717, 2022. doi: 10.1137/21M1464178. URL https://doi.org/10.1137/21M1464178.

Hiroyuki Sato and Toshihiro Iwai. A new, globally convergent Riemannian conjugate gradient method. *Optimization*, 64(4):1011–1031, 2015. doi: 10.1080/02331934.2013.836650. URL https://doi.org/10.1080/02331934.2013.836650.

Steffen Schotthöfer, Emanuele Zangrando, Jonas Kusch, Gianluca Ceruti, and Francesco Tudisco. Low-rank lottery tickets: finding efficient low-rank neural networks via matrix differential equations. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pp. 20051–20063, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Steffen Schotthöfer, Timon Klein, and Jonas Kusch. A geometric framework for momentum-based optimizers for low-rank training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=cCefuzQrjK.

Steffen Schotthöfer, Emanuele Zangrando, Gianluca Ceruti, Francesco Tudisco, and Jonas Kusch. GeoLoRA: Geometric integration for parameter efficient fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=bsFWJ0Kget.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.127063. URL https://www.sciencedirect.com/science/article/pii/S0925231223011864.

Bart Vandereycken and Stefan Vandewalle. A Riemannian Optimization Approach for Computing Low-Rank Solutions of Lyapunov Equations. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2553–2579, 2010. doi: 10.1137/090764566. URL https://doi.org/10.1137/090764566.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/d9fbed9da256e344c1fa46bb46c34c5f-Paper.pdf.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding, 2019. URL https://arxiv.org/abs/1804.07461.

Hongyi Wang, Saurabh Agarwal, and Dimitris Papailiopoulos. Pufferfish: Communication-efficient Models At No Extra Cost. In A. Smola, A. Dimakis, and I. Stoica (eds.), *Proceedings of Machine Learning and Systems*, volume 3, pp. 365–386, 2021. URL https://proceedings.mlsys.org/paper_files/paper/2021/file/94cb28874a503f34b3c4a41bddcea2bd-Paper.pdf.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. LoRA-Pro: Are Low-Rank Adapters Properly Optimized? In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Representation Learning*, pp. 93787–93808, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ea184f920a0f0f8d8030aa1bd7ac9fd4-Paper-Conference.pdf.

Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1):397–434, Dec 2013. ISSN 1436-4646. doi: 10.1007/s10107-012-0584-1. URL https://doi.org/10.1007/s10107-012-0584-1.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Y. Yang. Globally Convergent Optimization Algorithms on Riemannian Manifolds: Uniform Framework for Unconstrained and Constrained Optimization. *Journal of Optimization Theory and Applications*, 132(2):245–265, Feb 2007. ISSN 1573-2878. doi: 10.1007/s10957-006-9081-0. URL https://doi.org/10.1007/s10957-006-9081-0.

Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Sai Surya Duvvuri, Inderjit S Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. LoRA Done RITE: Robust Invariant Transformation Equilibration for LoRA Optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VpWki1v2P8.

Emanuele Zangrando, Steffen Schotthöfer, Jonas Kusch, Gianluca Ceruti, and Francesco Tudisco. Geometry-aware training of factorized layers in tensor Tucker format. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Fangzhao Zhang and Mert Pilanci. Riemannian preconditioned LoRA for fine-tuning foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, pp. 59641–59669. JMLR.org, 2024.

Hongyi Zhang, Sashank J. Reddi, and Suvrit Sra. Riemannian SVRG: Fast Stochastic Optimization on Riemannian Manifolds. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/98e6f17209029f4ae6dc9d88ec8eac2c-Paper.pdf.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-Efficient LLM Training by Gradient Low-Rank Projection. In *5th Workshop on practical ML for limited/low resource settings*, 2024. URL https://openreview.net/forum?id=AzqPyO22zt.

Xiaojing Zhu and Hiroyuki Sato. Riemannian conjugate gradient methods with inverse retraction. *Computational Optimization and Applications*, 77(3):779–810, Dec 2020. ISSN 1573-2894. doi: 10.1007/s10589-020-00219-6. URL https://doi.org/10.1007/s10589-020-00219-6.

Zhenyu Zhu, Yongtao Wu, Quanquan Gu, and Volkan Cevher. Imbalance-Regularized LoRA: A Plug-and-Play Method for Improving Fine-Tuning of Foundation Models. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*, 2024. URL https://openreview.net/forum?id=hQfJ7nyDeK.

A Appendix

A.1 Geometry of the Stiefel Manifold

The (column-orthonormal) Stiefel manifold is defined as

$$\mathrm{St}(n,r) \coloneqq \{X \in \mathbb{R}^{n\times r} \colon X^\top X = I_r\}.$$

It is a smooth embedded submanifold of $\mathbb{R}^{n\times r}$ of dimension $nr - \frac{1}{2}r(r+1)$.

The tangent space at a point $X \in \mathrm{St}(n,r)$ is given by

$$\mathrm{T}_X\mathrm{St}(n,r) = \left\{\xi \in \mathbb{R}^{n\times r} \colon X^\top\xi + \xi^\top X = 0\right\}.$$

Equivalently, any tangent vector $\xi \in \mathrm{T}_X\mathrm{St}(n,r)$ can be decomposed as

$$\xi = X\Omega + X_\perp K,$$

where $\Omega \in \mathbb{R}^{r\times r}$ is skew-symmetric, $X_\perp \in \mathbb{R}^{n\times(n-r)}$ satisfies $[X\ X_\perp] \in \mathrm{O}(n)$, $\mathrm{O}(n)$ being the orthogonal group, and $K \in \mathbb{R}^{(n-r)\times r}$.

A.1.1 Orthogonal Projection onto $\mathrm{T}_X\mathrm{St}(n,r)$

For any matrix $\xi \in \mathbb{R}^{n\times r}$, its orthogonal projection onto $\mathrm{T}_X\mathrm{St}(n,r)$ with respect to the Euclidean inner product is given by

$$\mathrm{P}_X(\xi) = X\,\mathrm{skew}(X^\top\xi) + (I_n - XX^\top)\,\xi, \tag{2}$$

where $\mathrm{skew}(M) \coloneqq \frac{1}{2}(M - M^\top)$ denotes the skew-symmetric part of a square matrix.

Equation equation 2 admits the equivalent and more compact expression

$$\mathrm{P}_X(\xi) = \xi - X\,\mathrm{sym}(X^\top\xi),$$

where $\mathrm{sym}(M) \coloneqq \frac{1}{2}(M + M^\top)$ denotes the symmetric part. This formula is commonly used in Riemannian optimization on the Stiefel manifold; see, e.g., (Absil et al., 2008, Prop. 3.6.1).

A.1.2 Row-Orthonormal Stiefel Manifold

In this work, we enforce a *row-orthonormal* constraint on the factor $A \in \mathbb{R}^{r\times n}$, namely,

$$AA^\top = I_r.$$

This corresponds to working with the transpose variable $X \coloneqq A^\top \in \mathrm{St}(n,r)$. All Riemannian operations (projection, retraction, and gradient computation) are therefore performed on X, and the updated factor is recovered as $A = X^\top$.

A.2 Cayley Transform and Retraction

A commonly used retraction on the Stiefel manifold $\mathrm{St}(n,r) = \{X \in \mathbb{R}^{n\times r} \colon X^\top X = I_r\}$ is based on the Cayley transform. The Cayley transform generates a smooth curve on the Stiefel manifold by exponentiating a skew-symmetric matrix in a rational form, thereby avoiding explicit matrix exponentials.

The closed-form Cayley retraction at a point $X \in \mathrm{St}(n,r)$ is defined as

$$Y(\alpha) = \left(I_n - \tfrac{\alpha}{2}\Omega\right)^{-1}\left(I_n + \tfrac{\alpha}{2}\Omega\right)X, \tag{3}$$

where $\Omega \in \mathbb{R}^{n\times n}$ is a skew-symmetric matrix and $\alpha \geq 0$ is a step-size parameter. The curve satisfies

$$Y(0) = X, \qquad \frac{\mathrm{d}}{\mathrm{d}\alpha}Y(\alpha)\big|_{\alpha=0} = \Omega X,$$

and therefore defines a valid first-order retraction on the Stiefel manifold (Absil et al., 2008).

Computing the closed-form expression equation 3 requires solving a linear system involving an $n \times n$ matrix, which can be computationally expensive for large n. A fixed-point approximation of the Cayley transform is given by

$$Y(\alpha) = X + \frac{\alpha}{2}\,\Omega\big(X + Y(\alpha)\big). \tag{4}$$

Starting from the initialization $Y_0 = (I + \alpha\Omega)X$, this fixed-point equation can be solved with a small number of iterations, each involving only matrix multiplications. In practice, only a few iterations are sufficient to obtain an accurate approximation of the exact Cayley retraction.

A.2.1 CAYLEY RETRACTION IN OUR SETTING

As mentioned in the main text, when factorizing a weight matrix, we work with the row-orthonormal factor of size r-by-n satisfying $AA^\top = I_r$. For convenience, we switch to a column-orthonormal representation so that $X \coloneqq A^\top \in \mathrm{St}(n, r)$.

Let $\xi \in \mathrm{T}_{X_t}\mathrm{St}(n, r)$ be a tangent vector at $X_t \coloneqq A_t^\top$. Following (Li et al., 2020, Eq. (2)), the Cayley retraction used in our algorithm is defined as

$$X_{t+1} = \left(I_n - \tfrac{1}{2}\Omega\right)^{-1}\left(I_n + \tfrac{1}{2}\Omega\right)X_t, \qquad A_{t+1} \coloneqq X_{t+1}^\top,$$

where the skew-symmetric matrix Ω is constructed as

$$\widehat{\Omega} = \xi X_t^\top - \tfrac{1}{2}X_t(X_t^\top \xi X_t^\top), \qquad \Omega = \widehat{\Omega} - \widehat{\Omega}^\top.$$

In practice, we compute the Cayley retraction using the fixed-point iteration equation 4. This yields an efficient and numerically stable retraction satisfying the standard first-order retraction conditions $\mathrm{Retr}_x(0_x) = x$ and $\mathrm{DRetr}_x(0_x) = \mathrm{Id}$ required for the convergence analysis of Riemannian optimization methods; see, e.g., (Absil et al., 2008, §4.1).

A.2.2 FIXED-POINT APPROXIMATION OF THE CAYLEY RETRACTION

In this section, we analyze the fixed-point iteration used to approximate the Cayley retraction and show that it geometrically converges to the exact Cayley transform. Most importantly, we further show that the resulting approximate mapping retains the retraction properties required by the convergence theory.

Let $\{Y_i\}_{i\geq 0}$ be the sequence of fixed-point iterates defined by

$$Y_{i+1} = \mathcal{F}(Y_i), \tag{5}$$

where $\mathcal{F}$ is the fixed-point map of equation 4, namely, $\mathcal{F}(Y) \coloneqq X + \frac{\alpha}{2}\Omega(X + Y)$, and the initialization is $Y_0 = (I + \alpha\Omega)X$. We recall the contraction property of a fixed-point map.

Lemma A.1 (Contraction of the fixed-point map)**.** *Assume that $\alpha\|\Omega\|_2 < 2$ ($\alpha \geq 0$). Then $\mathcal{F}$ is a contraction mapping on $\mathbb{R}^{n\times r}$ with contraction factor*

$$\rho \coloneqq \frac{\alpha}{2}\|\Omega\|_2 < 1.$$

Proof. For any $Y_1, Y_2 \in \mathbb{R}^{n\times r}$,

$$\|\mathcal{F}(Y_1) - \mathcal{F}(Y_2)\| = \left\|\frac{\alpha}{2}\Omega Y_1 - \frac{\alpha}{2}\Omega Y_2\right\| \leq \frac{\alpha}{2}\|\Omega\|_2\,\|Y_1 - Y_2\| < \|Y_1 - Y_2\|,$$

which shows that $\mathcal{F}$ is a contraction mapping. □

Theorem A.2 (Geometric convergence to the Cayley retraction)**.** *Let $Y^\star \coloneqq \mathrm{Retr}_X(\alpha\xi)$ be the exact Cayley retraction of $\alpha\xi$ at X. Under the assumptions of Lemma A.1, the fixed-point iterates defined by equation 5 satisfy*

$$\forall i \geq 0, \qquad \|Y_i - Y^\star\| \leq \rho^i\,\|Y_0 - Y^\star\|. \tag{6}$$

Proof. Since Lemma A.1 shows that $\mathcal{F}$ is a contraction mapping, the Banach fixed-point theorem applies, and the exact retraction $Y^\star = \mathrm{Retr}_X(\alpha\xi)$ is the unique fixed point of $\mathcal{F}$. The theorem's result equation 6 follows directly. □

We are now in the position to state the properties of the approximate retraction, which we denote by

$$\widetilde{\mathrm{Retr}}_X(\alpha\xi) \coloneqq Y_s,$$

where s is the total number of fixed-point iterations performed.

Lemma A.3 (Second-order accuracy)**.** *For sufficiently small α and any fixed number of iterations s, the approximate Cayley retraction satisfies*

$$\widetilde{\mathrm{Retr}}_X(\alpha\xi) = X + \alpha\xi + \mathcal{O}(\alpha^2) + \mathcal{O}(\rho^s),$$

and converges to the exact Cayley retraction as $s \to \infty$.

In particular, for fixed s and sufficiently small step size α, the approximation error remains of higher order and does not dominate the first-order behavior of the update. This justifies using a finite number of fixed-point iterations in practice.

Proof. The exact Cayley transform equation 3 admits the expansion

$$\mathrm{Retr}_X(\alpha\xi) = X + \alpha\xi + \mathcal{O}(\alpha^2). \tag{7}$$

The initialization of the fixed-point iteration method is

$$Y_0 = (I + \alpha\Omega)X = X + \alpha\Omega X = X + \alpha\xi,$$

which is clearly a first-order approximation of $\mathrm{Retr}_X(\alpha\xi)$. Therefore

$$\|Y_0 - Y^\star\| = \mathcal{O}(\alpha^2). \tag{8}$$

By Theorem A.2, we have the contraction estimate equation 6. Inserting equation 8 into the equation 6, with $i = s$, we obtain

$$\|Y_s - Y^\star\| \le \rho^s\, \mathcal{O}(\alpha^2).$$

Since α is fixed within one update, the factor $\mathcal{O}(\alpha^2)$ can be absorbed into the constant, i.e.,

$$\|Y_s - Y^\star\| \le \mathcal{O}(\rho^s).$$

or, equivalently, $Y_s = \mathrm{Retr}_X(\alpha\xi) + \mathcal{O}(\rho^s)$. Combining this with the expansion of the exact retraction equation 7, the result of the lemma follows immediately. □

In particular, Lemma A.3 shows that the approximate Cayley mapping satisfies the first-order retraction conditions up to a controllable error. Such inexact retractions preserve the convergence guarantees of Riemannian first-order methods provided the approximation error is sufficiently small; see, e.g., standard analyses of inexact retraction schemes.

Lemma A.3 shows that for any fixed s and sufficiently small α, the approximate Cayley transform defined by the fixed-point iteration still satisfies the first-order retraction condition. The term $\mathcal{O}(\rho^s)$ is the numerical approximation error, controlled by the number of iterations s of the fixed-point approximation method. In particular, for fixed α and sufficiently large s, the approximate Cayley transform converges to the exact Cayley map.

A.2.3 Choice of Retraction

To map a tangent vector ξ_t back onto the manifold, we need to apply a retraction mapping, $X_{t+1} = \mathrm{Retr}_{X_t}(\xi_t)$. While the exponential map provides the most geometrically accurate geodesic path, it is often computationally prohibitive for large-scale problems because it requires full eigenvalue decompositions or matrix exponentials (Absil et al., 2008). In practice, a retraction is any mapping that agrees to first order with the exponential map (i.e., is centered at the point and has the differential at the origin equal to the identity map).

For the Stiefel manifold $\mathrm{St}(n, r)$, several efficient retractions exist with a computational complexity of $O(nr^2 + r^3)$, which is ideal for settings where $r \ll n$:

- **QR Decomposition:** A standard choice that performs a QR factorization of $X_t + D_t$ and extracts the orthogonal factor Q (Absil et al., 2008, (4.8)).
- **Polar Decomposition:** Maps the tangent vector to the manifold by finding the closest orthogonal matrix in the Frobenius norm, typically implemented via iterative Newton–Schulz methods (Zhu & Sato, 2020).
- **Cayley Transform:** An algebraic alternative using a skew-symmetric mapping. When implemented with the Sherman–Morrison–Woodbury (SMW) identity, it avoids large matrix inversions, reducing the cost to a $2r \times 2r$ system (Wen & Yin, 2013, §2.2). It can also be computed implicitly via the fixed-point iteration $Y_{k+1} = X + \frac{\alpha}{2}\Omega\big(X + Y_k\big)$, which converges quadratically as $o(\alpha^{2+k})$.

B PROOF OF THEOREM 4.2

Proof. Consider the sequence $\mathcal{L}_t$ of convex loss functions and define

$$R(T) \coloneqq \sum_{t=1}^{T} \mathcal{L}_t(B_t, A_t) - \mathcal{L}_t(B^*, A^*),$$

where (B_t, A_t) is the iteration in Algorithm 1, and (B^*, A^*) is a minimizer of $\sum_{t=1}^{T} \mathcal{L}_t(B, A)$. Using the convexity of $\mathcal{L}_t$, and defining $G_t^{A,B} = \nabla_{A,B}\mathcal{L}_t(B_t, A_t)$, we get

$$R(T) \leq \sum_{t=1}^{T} \langle B_t - B^*, G_t^B \rangle + \sum_{t=1}^{T} \langle A_t - A^*, G_t^A \rangle. \tag{9}$$

We now bound the first term on the right-hand side of equation 9, i.e., the Euclidean term $\langle B_t - B^*, G_t^B \rangle$. Let $H_t^B \colon \mathbb{R}^{m\times r} \to \mathbb{R}^{m\times r}$ the linear operator defined by $H_t^B(X) = (V_t^B + \varepsilon) \odot X$. We bound the norm $\|B_{t+1} - B^*\|^2_{(H_t^B)^{1/2}}$ (see Definition B.1):

$$\begin{aligned}
\|B_{t+1} - B^*\|^2_{(H_t^B)^{1/2}} =& \|B_t - \eta_t (H_t^B)^{-1/2} M_t^B - B^*\|^2_{(H_t^B)^{1/2}} = \|B_t - B^*\|^2_{(H_t^B)^{1/2}} + \\
&+ \eta_t^2 \|(H_t^B)^{-1/2} M_t^B\|^2_{(H_t^B)^{1/2}} - 2\eta_t \langle B_t - B^*, (H_t^B)^{-1/2} M_t^B \rangle_{(H_t^B)^{1/2}} \\
&\underset{\text{Lemma B.2}}{=} \|B_t - B^*\|^2_{(H_t^B)^{1/2}} + \eta_t^2 \|M_t^B\|^2_{(H_t^B)^{-1/2}} - 2\eta_t \langle B_t - B^*, M_t^B \rangle,
\end{aligned} \tag{10}$$

where the last inner product is in the Frobenius norm (for clarity, we always omit the subscript). By rearranging equation 10 (bringing the inner product on the left-hand side and the norm on the right-hand side), we get:

$$2\eta_t \langle B_t - B^*, M_t^B \rangle = \underbrace{\|B_t - B^*\|^2_{(H_t^B)^{1/2}} - \|B_{t+1} - B^*\|^2_{(H_t^B)^{1/2}}}_{=:\Delta_t^B} + \eta_t^2 \|M_t^B\|^2_{(H_t^B)^{-1/2}}. \tag{11}$$

Using the definition of $M_t^B = \beta_1 M_{t-1}^B + (1-\beta_1) G_t^B$ in equation 11, and by defining $\Delta_t^B \coloneqq \|B_t - B^*\|^2_{(H_t^B)^{1/2}} - \|B_{t+1} - B^*\|^2_{(H_t^B)^{1/2}}$, we get

$$\begin{aligned}
\langle B_t - B^*, G_t^B \rangle =& \frac{1}{2(1-\beta_1)\eta_t} \Delta_t^B + \frac{\eta_t}{2(1-\beta_1)} \|M_t^B\|^2_{(H_t^B)^{-1/2}} - \frac{\beta_1}{1-\beta_1} \langle B_t - B^*, M_{t-1}^B \rangle \\
\leq& \frac{1}{2(1-\beta_1)\eta_t} \Delta_t^B + \frac{\eta_t}{2(1-\beta_1)} \|M_t^B\|^2_{(H_t^B)^{-1/2}} + \frac{\beta_1}{1-\beta_1} \left| \langle B_t - B^*, M_{t-1}^B \rangle \right| \\
\underset{\text{Lemma B.3}}{\leq}& \frac{\Delta_t^B}{2(1-\beta_1)\eta_t} + \frac{\eta_t}{2(1-\beta_1)} \|M_t^B\|^2_{(H_t^B)^{-1/2}} + \frac{\beta_1}{2(1-\beta_1)\alpha_t^2} \|M_{t-1}^B\|^2_{(H_t^B)^{-1/2}} \\
&+ \frac{\beta_1 \alpha_t^2}{2(1-\beta_1)} \|B_t - B^*\|^2_{(H_t^B)^{1/2}}.
\end{aligned} \tag{12}$$

Apart from the retraction, the term $\langle A_t - A^*, G_t^A \rangle$ is similar to equation 12, i.e.,

$$A_{t+1} = \mathrm{Retr}_{A_t}\Big(-\eta_t \, \mathrm{P}_{A_t}\Big((H_t^A)^{-1/2} M_t^A\Big)\Big).$$

We define

$$E_t \coloneqq \mathrm{Retr}_{A_t}(-\eta_t \, \mathrm{P}_{A_t}(H_t^A)^{-1/2} M_t^A) - (A_t - \eta_t \, \mathrm{P}_{A_t}(H_t^A)^{-1/2} M_t^A),$$

and the invertible linear operator $\Gamma_t = \mathrm{P}_{A_t}(H_t^A)^{-1/2} \, \mathrm{P}_{A_t} \colon \mathrm{T}_{A_t}\mathrm{St}(n,r) \to \mathrm{T}_{A_t}\mathrm{St}(n,r)$. We define $\bar{\Gamma}_t$ as an extension of the previous map on the whole space, $\bar{\Gamma}_t = \Gamma_t + \gamma_t(I - \mathrm{P}_{A_t})$, where $\gamma_t > 0$ is a scalar. With a small abuse of notation, we will still denote by Γ_t the map $\bar{\Gamma}_t$ when there is no risk of confusion. Let $D_t \coloneqq \mathrm{P}_{A_t}(H_t^A)^{-1/2} M_t^A \in \mathrm{T}_{A_t}\mathrm{St}(n,r)$, and consider the norm

$$\begin{aligned}
\|A_{t+1} - A^*\|^2_{\Gamma_t^{-1}} &= \|A_t - \eta_t D_t + E_t - A^*\|^2_{\Gamma_t^{-1}} \\
&= \|A_t - A^*\|^2_{\Gamma_t^{-1}} - 2\eta_t \langle \Gamma_t^{-1} D_t, A_t - A^* \rangle + \eta_t^2 \|D_t\|^2_{\Gamma_t^{-1}} \\
&\quad - 2\eta_t \langle E_t, \Gamma_t^{-1} D_t \rangle + 2\langle \Gamma_t^{-1} E_t, A_t - A^* \rangle + \|E_t\|^2_{\Gamma_t^{-1}} \\
&\underset{\langle E_t, A_t - A^* \rangle_{\Gamma_t^{-1}} \leq 0}{\leq} \|A_t - A^*\|^2_{\Gamma_t^{-1}} - 2\eta_t \langle \Gamma_t^{-1} D_t, A_t - A^* \rangle + \eta_t^2 \|D_t\|^2_{\Gamma_t^{-1}} + \underbrace{-2\eta_t \langle E_t, \Gamma_t^{-1} D_t \rangle + \|E_t\|^2_{\Gamma_t^{-1}}}_{=:\delta_t}.
\end{aligned}$$

namely,

$$\|A_{t+1} - A^*\|^2_{\Gamma_t^{-1}} \leq \|A_t - A^*\|^2_{\Gamma_t^{-1}} - 2\eta_t \left\langle \Gamma_t^{-1} D_t, A_t - A^* \right\rangle + \eta_t^2 \|D_t\|^2_{\Gamma_t^{-1}} + \delta_t. \tag{13}$$

Bringing the inner product in equation 13 to the left-hand side and the norm to the right-hand side leads to

$$2\eta_t \left\langle \Gamma_t^{-1} D_t, A_t - A^* \right\rangle \leq \|A_t - A^*\|^2_{\Gamma_t^{-1}} - \|A_{t+1} - A^*\|^2_{\Gamma_t^{-1}} + \eta_t^2 \|D_t\|^2_{\Gamma_t^{-1}} + \delta_t.$$

Defining $\Delta_t^A \coloneqq \|A_t - A^*\|^2_{\Gamma_t^{-1}} - \|A_{t+1} - A^*\|^2_{\Gamma_t^{-1}}$, and dividing by $2\eta_t$, we can write

$$\langle \Gamma_t^{-1} D_t, A_t - A^* \rangle \leq \frac{1}{2\eta_t} \Delta_t^A + \frac{\eta_t}{2} \|D_t\|^2_{\Gamma_t^{-1}} + \frac{1}{2\eta_t} \delta_t.$$

We now notice that, by definition of Γ_t, we have $\Gamma_t^{-1} D_t = M_t^A$, and, by using again Young inequality (Lemma B.3), we get

$$\begin{aligned}
\langle G_t^A, A_t - A^* \rangle &\leq \frac{1}{2(1-\beta_1)\eta_t} \Delta_t^A + \frac{\eta_t}{2(1-\beta_1)} \|D_t\|^2_{\Gamma_t^{-1}} + \frac{1}{2(1-\beta_1)\eta_t} \delta_t \\
&\quad + \frac{\beta_1 \alpha_t^2}{2(1-\beta_1)} \|A_t - A^*\|_{\mathrm{F}}^2 + \frac{\beta_1}{2(1-\beta_1)\alpha_t^2} \|M_{t-1}^A\|_{\mathrm{F}}^2 \\
&= \frac{1}{2(1-\beta_1)\eta_t} \Delta_t^A + \frac{\eta_t}{2(1-\beta_1)} \left\| (H_t^A)^{-1/4} (H_t^A)^{1/4} \Gamma_t^{1/2} \Gamma_t^{-1} D_t \right\|_{\mathrm{F}}^2 + \frac{1}{2(1-\beta_1)\eta_t} \delta_t \\
&\quad + \frac{\beta_1 \alpha_t^2}{2(1-\beta_1)} \left\| (H_t^A)^{-1/4} (H_t^A)^{1/4} (A_t - A^*) \right\|_{\mathrm{F}}^2 + \frac{\beta_1}{2(1-\beta_1)\alpha_t^2} \|(H_t^A)^{-1/4} (H_t^A)^{1/4} M_{t-1}^A\|_{\mathrm{F}}^2 \\
&= \frac{1}{2(1-\beta_1)\eta_t} \Delta_t^A + \frac{\eta_t}{2(1-\beta_1)} \|(H_t^A)^{1/4} \Gamma_t^{1/2} M_t^A\|^2_{(H_t^A)^{-1/2}} + \frac{1}{2(1-\beta_1)\eta_t} \delta_t \\
&\quad + \frac{\beta_1 \alpha_t^2}{2(1-\beta_1)} \|(H_t^A)^{1/4} (A_t - A^*)\|^2_{(H_t^A)^{-1/2}} + \frac{\beta_1}{2(1-\beta_1)\alpha_t^2} \|(H_t^A)^{-1/4} M_{t-1}^A\|^2_{(H_t^A)^{1/2}} \\
&\leq \frac{\Delta_t^A}{2(1-\beta_1)\eta_t} + \frac{\eta_t \|(H_t^A)^{1/4} \Gamma_t^{1/2}\|_{\mathrm{op}}^2}{2(1-\beta_1)} \|M_t^A\|^2_{(H_t^A)^{-1/2}} + \frac{\delta_t}{2(1-\beta_1)\eta_t} \\
&\quad + \frac{\beta_1 \alpha_t^2 \|(H_t^A)^{1/4}\|_{\mathrm{op}}^2}{2(1-\beta_1)} \|A_t - A^*\|^2_{(H_t^A)^{-1/2}} + \frac{\beta_1 \|(H_t^A)^{-1/4}\|_{\mathrm{op}}^2}{2(1-\beta_1)\alpha_t^2} \|M_{t-1}^A\|^2_{(H_t^A)^{1/2}}.
\end{aligned} \tag{14}$$

The term δ_t is the only one structurally different from the ones in equation 12. Thus, equation 14 can be bounded using the definition of E_t (retraction error) with Lagrange remainder error

$$\|E_t\| \leq C \| \mathrm{D}^2 \operatorname{Retr}_{A_t}(\zeta) \| \|\eta_t D_t\|^2 \leq \tilde{C} \eta_t^2 \|D_t\|^2,$$

as

$$\begin{aligned}
\frac{\delta_t}{2(1-\beta_1)\eta_t} &\leq \frac{1}{2(1-\beta_1)\eta_t} \left[2\eta_t \|E_t\| \|M_t^A\| + \|\Gamma_t^{-1}\|_{\mathrm{op}} \|E_t\|^2 \right] \\
&\leq \frac{1}{2(1-\beta_1)} \left[2\|E_t\| \|M_t^A\| + \frac{1}{\eta_t} \|\Gamma_t^{-1}\|_{\mathrm{op}} \|E_t\|^2 \right] \\
&\leq \frac{1}{2(1-\beta_1)} \left[2\eta_t^2 \|D_t\|^2 \|M_t^A\| + \eta_t^3 \|\Gamma_t^{-1}\|_{\mathrm{op}} \|D_t\|^4 \right].
\end{aligned}$$

Using the definition

$$\|M_t^A\| \coloneqq \|(1-\beta_1) \sum_{s=1}^{t} \beta_1^{t-s} G_s^A\| \leq G_\infty,$$

and the fact that $\|D_t\| \lesssim G_\infty$, $\|\Gamma_t^{-1}\|_{\mathrm{op}}$ bounded, we get

$$\sum_{t=1}^{T} \frac{\delta_t}{2(1-\beta_1)\eta_t} \lesssim \frac{1}{2(1-\beta_1)} \left[G_\infty^3 \sum_{t=1}^{T} \eta_t^2 + G_\infty^4 \sum_{t=1}^{T} \eta_t^3 \right]. \tag{15}$$

By combining equations equation 9, equation 12, equation 14 and equation 15, we get

$$\begin{aligned}
R(T) &\leq \sum_{t=1}^{T} \left(\frac{\Delta_t^B}{2(1-\beta_1)\eta_t} + \frac{\eta_t \|M_t^B\|^2_{(H_t^B)^{-1/2}}}{2(1-\beta_1)} + \frac{\beta_1 \|M_{t-1}^B\|^2_{(H_t^B)^{-1/2}}}{2(1-\beta_1)\alpha_t^2} + \frac{\beta_1 \alpha_t^2 \|B_t - B^*\|^2_{(H_t^B)^{1/2}}}{2(1-\beta_1)} \right) \\
&\quad + \sum_{t=1}^{T} \left(\frac{\Delta_t^A}{2(1-\beta_1)\eta_t} + \frac{\eta_t K_1 \|M_t^A\|^2_{(H_t^A)^{-1/2}}}{2(1-\beta_1)} + \frac{\beta_1 K_2 \|M_{t-1}^A\|^2_{(H_t^A)^{1/2}}}{2(1-\beta_1)\alpha_t^2} + \frac{\beta_1 \alpha_t^2 K_3 \|A_t - A^*\|^2_{(H_t^A)^{-1/2}}}{2(1-\beta_1)} \right) \\
&\quad + \frac{1}{2(1-\beta_1)} \left[G_\infty^3 \sum_{t=1}^{T} \eta_t^2 + G_\infty^4 \sum_{t=1}^{T} \eta_t^3 \right],
\end{aligned} \tag{16}$$

where K_1, K_2, K_3 are three constants bounding the operator norms in equation 14. Notice that up to these constants, the terms in A and B are similar to each other; therefore, we focus on bounding the terms in B (the ones in A yield analogous bounds).

Now, we observe that, thanks to (Reddi et al., 2018, Lemma 2), for $\eta_t = \eta/\sqrt{t}$ we obtain the following bounds

$$\sum_{t=1}^{T} \eta_t \|M_t^B\|^2_{(H_t^B)^{-1/2}} \leq \frac{\eta \|G^B\|_{L^1L^2([0,T])}\sqrt{1+\log T}}{(1-\beta_1)(1-\beta_1/\sqrt{\beta_2})\sqrt{1-\beta_2}},$$

where $\|G^B\|_{L^1L^2([0,T])}$ is the $L^1 - L^2$ norm defined by

$$\|G^B\|_{L^1L^2([0,T])} \coloneqq \sum_{i,j} \left(\sum_{t=1}^{T} \left|(G_t^B)_{ij}\right|^2 \right)^{1/2}.$$

Moreover,

$$\sum_{t=1}^{T} \eta_t \|M_t^A\|^2_{(H_t^A)^{-1/2}} \leq \frac{\eta \|G^A\|_{L^1L^2([0,T])}\sqrt{1+\log T}}{(1-\beta_1)(1-\beta_1/\sqrt{\beta_2})\sqrt{1-\beta_2}},$$

where

$$\sum_{t=1}^{T} \eta_t^2 = \eta^2 \sum_{t=1}^{T} \frac{1}{t} \leq \eta^2(1+\log T), \quad \text{and} \quad \sum_{t=1}^{T} \eta_t^3 = \sum_{t=1}^{T} \frac{\eta}{t^{3/2}} \leq 1 + \int_1^T t^{-3/2}\,\mathrm{d}t = 3 - \frac{2}{\sqrt{T}}. \tag{17}$$

Similarly, we can bound the third term on the right-hand side of equation 16, i.e.,

$$\begin{aligned}
\|M_{t-1}^B\|^2_{(H_t^B)^{-1/2}} &= \left\| (H_t^B)^{-1/4}(H_{t-1}^B)^{1/4}(H_{t-1}^B)^{-1/4}(M_{t-1}^B) \right\|_{\mathrm{F}}^2 \\
&\leq \left\| (H_t^B)^{-1/4}(H_{t-1}^B)^{1/4} \right\|_{2\to 2}^2 \|M_{t-1}^B\|^2_{(H_{t-1}^B)^{-1/2}} \\
&\leq \left(\frac{1}{\beta_2}\right)^{1/4} \|M_{t-1}^B\|^2_{(H_{t-1}^B)^{-1/2}},
\end{aligned} \tag{18}$$

where the last inequality is given by the fact that

$$\begin{aligned}
\left\| (H_t^B)^{-1/4}(H_{t-1}^B)^{1/4} \right\|_{2\to 2}^4 &= \left\| \left(\frac{V_{t-1}^B}{V_t^B}\right)^{1/4} \right\|_{L^\infty}^4 = \left\| \frac{V_{t-1}^B}{\beta_2 V_{t-1}^B + (1-\beta_2)\,(G_t^B)^{\circ 2}} \right\|_{L^\infty} \\
&= \left\| 1 + \frac{(1-\beta_2)V_{t-1}^B}{\beta_2 V_{t-1}^B + (1-\beta_2)\,(G_t^B)^{\circ 2}} \right\|_{L^\infty} \leq 1 + \frac{1-\beta_2}{\beta_2}.
\end{aligned}$$

The result in equation 18 allows to apply (Reddi et al., 2018, Lemma 2) (or, equivalently, using equation 17) also on the time-shifted term $\|M_{t-1}^B\|_{(H_t^B)^{-1/2}}$, which for $1/\alpha_t^2 = \alpha/\sqrt{t}$ gives

$$\begin{aligned}
\sum_{t=1}^{T} \frac{\|M_{t-1}^B\|^2_{(H_t^B)^{-1/2}}}{\alpha_t^2} &\leq \sum_{t=1}^{T} \frac{\alpha}{\sqrt{t}} \left(\frac{1}{\beta_2}\right)^{1/4} \|M_{t-1}^B\|^2_{(H_{t-1}^B)^{-1/2}} \\
&\leq \frac{\alpha \|G^B\|_{L^1L^2([0,T])}\sqrt{1+\log T}}{\beta_2^{1/4}(1-\beta_1)(1-\beta_1/\sqrt{\beta_2})\sqrt{1-\beta_2}}.
\end{aligned} \tag{19}$$

Using assumptions (H2) that $\beta_{1,t} = \beta_1 b^t$, and (H3) that $\sup_t \|B_t - B^*\|_{\max} \leq D_\infty$ (the same quantity can be bounded by 2 for the term in A, given that Stiefel is compact and has finite diameter), together with the last bound from (Reddi et al., 2018, Theorem 4), and combining the results from

equation 17, equation 19,equation 16 (for A_t they are similar), we get

$$
\begin{aligned}
R(T) \leq & \frac{D_\infty^2 \left\|(H_T^B)^{1/4}\right\|_{2\to 2}^2}{2\eta_T(1-\beta_1)} + \frac{D_\infty^2}{2(1-\beta_1)} \sum_{t=1}^{T} \beta_{1,t}\alpha_t^2 \left\|(H_t^B)^{1/4}\right\|_{2\to 2}^2 \\
& + \frac{\operatorname{diam}_\infty(\mathrm{St}(n,r))^2 \left\|(H_T^A)^{1/4}\right\|_{2\to 2}^2}{2\eta_T(1-\beta_1)} + \frac{\operatorname{diam}_\infty(\mathrm{St}(n,r))^2}{2(1-\beta_1)} \sum_{t=1}^{T} \beta_{1,t}\alpha_t^2 \left\|(H_t^A)^{1/4}\right\|_{2\to 2}^2 \\
& + \frac{(\eta + 2\alpha\beta_1^{3/4})\sqrt{1+\log T}}{2(1-\beta_1)^2(1-\beta_1/\sqrt{\beta_2})\sqrt{1-\beta_2}} \left\|G^A\right\|_{L^1L^2([0,T])} \\
& + \frac{(\eta + 2\alpha\beta_1^{3/4})\sqrt{1+\log T}}{2(1-\beta_1)^2(1-\beta_1/\sqrt{\beta_2})\sqrt{1-\beta_2}} \left\|G^B\right\|_{L^1L^2([0,T])} \\
& + \frac{1}{2(1-\beta_1)} \left[G_\infty^3 \eta^2 (1+\log T) + G_\infty^4 \eta^3 \left(3 - \frac{2}{\sqrt{T}}\right)\right].
\end{aligned} \tag{20}
$$

To conclude, we bound the two terms $\sum_{t=1}^{T} \beta_{1,t}\alpha_t^2 \|(H_t^B)^{1/4}\|_{2\to 2}^2$ uniformly in t, for the choice $\beta_{1,t} = \beta_1 b^t$, and $\alpha_t^2 = \sqrt{t}/\alpha$, we get the upper bound

$$
\begin{aligned}
\sum_{t=1}^{T} \beta_{1,t}\alpha_t^2 \|(H_t^B)^{1/4}\|_{2\to 2}^2 &\leq \frac{\beta_1}{\alpha} \|\|(H^B)^{1/4}\|_{2\to 2}^2\|_{L_t^\infty} \sum_{t=1}^{T} \sqrt{t} b^t \\
&\underset{\substack{\text{Cauchy–Schwarz} \\ \sqrt{t}b^{t/2},\ b^{t/2}}}{\leq} \frac{\beta_1}{\alpha} \|\|(H^B)^{1/4}\|_{2\to 2}^2\|_{L_t^\infty} \left(\sum_{t=1}^{T} t b^t\right)^{1/2} \left(\sum_{t=1}^{T} b^{t/2}\right)^{1/2} \\
&\leq \frac{\beta_1}{\alpha} \|\|(H^B)^{1/4}\|_{2\to 2}^2\|_{L_t^\infty} \frac{b}{(1-b)^{3/2}}.
\end{aligned}
$$

This gives the final bound

$$
\begin{aligned}
R(T) \lesssim & \frac{D_\infty^2 \|(H_T^B)^{1/4}\|_{2\to 2}^2}{2\eta_T(1-\beta_1)} + \frac{D_\infty^2}{2(1-\beta_1)} \frac{\beta_1}{\alpha} \|\|(H^B)^{1/4}\|_{2\to 2}^2\|_{L_t^\infty} \frac{b}{(1-b)^{3/2}} \\
& + \frac{(\eta + 2\alpha\beta_1^{3/4})\sqrt{1+\log T}}{2(1-\beta_1)^2(1-\beta_1/\sqrt{\beta_2})\sqrt{1-\beta_2}} \|G^B\|_{L^1L^2([0,T])} + \frac{\operatorname{diam}_\infty(\mathrm{St}(n,r))^2 \|(H_T^A)^{1/4}\|_{2\to 2}^2}{2\eta_T(1-\beta_1)} \\
& + \frac{\operatorname{diam}_\infty(\mathrm{St}(n,r))^2}{2(1-\beta_1)} \frac{\beta_1}{\alpha} \|\|(H^A)^{1/4}\|_{2\to 2}^2\|_{L_t^\infty} \frac{b}{(1-b)^{3/2}} \\
& + \frac{(\eta + 2\alpha\beta_1^{3/4})\sqrt{1+\log T}}{2(1-\beta_1)^2(1-\beta_1/\sqrt{\beta_2})\sqrt{1-\beta_2}} \|G^A\|_{L^1L^2([0,T])} \\
& + \frac{1}{2(1-\beta_1)} \left[G_\infty^3 \eta^2 (1+\log T) + G_\infty^4 \eta^3 \left(3 - \frac{2}{\sqrt{T}}\right)\right].
\end{aligned}
$$

By using the fact that η_T^{-1}, we can collect the constant C_1 of the terms of order 1, the constant C_2 of order $\sqrt{T}$, the constant C_3 of the terms of order $\sqrt{1+\log T}$, and C_4 of the terms of order $\log T$ and with C_5 the constant of the term of order $T^{-1/2}$ to get the final bound. In particular, the order of the bound is

$$
R(T) \leq C_1\sqrt{T} + C_2 + C_3\sqrt{1+\log T} + C_4 \log T + C_5\, T^{-1/2}.
$$

□

Definition B.1. (H-norm induced by a full-rank operator) Let $H\colon V \to V$ be a self-adjoint, positive-definite operator on a finite-dimensional real Hilbert space (V, g). We define the H-weighted inner product as

$$
g_H(v,w) \coloneqq g(H^{1/2}v, H^{1/2}w),
$$

and denote the corresponding norm as

$$
\|x\|_H = \|H^{1/2}x\|_g.
$$

Lemma B.2. *(Properties of H-induced inner products) Let H and V be as in Definition B.1. Then the following holds:*

- $g_{H^\gamma}(H^\alpha v, H^\beta w) = g_{H^\gamma}(v, H^{\alpha+\beta} w) = g_{H^{\alpha+\beta+\gamma}}(v, w)$,
- $\|H^\alpha x\|_{H^\gamma} = \|H^{\alpha+\gamma/2} x\|_g = \|x\|_{H^{2\alpha+\gamma}}$.

Proof. The first point follows from the definition, self-adjointness $g(Hv, w) = g(v, Hw)$ (which holds for any power), and the fact that powers commute ($H^\alpha H^\beta = H^\beta H^\alpha$ for all α, β):

$$\begin{aligned} g_{H^\gamma}(H^\alpha v, H^\beta w) =& g(H^{\alpha+\gamma/2} v, H^{\beta+\gamma/2} w) = g(v, H^{\alpha+\beta+\gamma} w) = g(H^{(\alpha+\beta+\gamma)/2} v, H^{(\alpha+\beta+\gamma)/2} w) \\ =& g_{H^{\alpha+\beta+\gamma}}(v, w). \end{aligned}$$

The first equality is similar

$$g_{H^\gamma}(H^\alpha v, H^\beta w) = g(H^\alpha H^{\gamma/2} v, H^\beta H^{\gamma/2} w) = g(H^{\gamma/2} v, H^{\gamma/2} H^{\alpha+\beta} w) = g_{H^\gamma}(v, H^{\alpha+\beta} w).$$

The norm equality follows immediately from the definition

$$\|H^\alpha x\|_{H^\gamma}^2 = g_{H^\gamma}(H^\alpha x, H^\alpha x) = g(H^{\alpha+\gamma/2} x, H^{\alpha+\gamma/2} x) = \|H^{\alpha+\gamma/2} x\|_g^2 = \|x\|_{H^{2\alpha+\gamma}}^2.$$

□

Lemma B.3. *(Young inequality for dual H norms) Let H, V be as in Definition B.1 and let $u, v \in V$, $\zeta \neq 0$, and $\alpha \in \mathbb{R}$. Then, we have*

$$g(u, v) \leq \frac{1}{2\zeta^2} \|u\|_{H^{-\alpha}}^2 + \frac{\zeta^2}{2} \|v\|_{H^\alpha}^2.$$

Proof. Consider the expansion

$$0 \leq \|u - v\|_g^2 = g(u - v, u - v) = \|u\|_g^2 + \|v\|_g^2 - 2g(u, v),$$

which implies

$$g(u, v) \leq \frac{1}{2} \|u\|_g^2 + \frac{1}{2} \|v\|_g^2.$$

Now, since H is self-adjoint and positive definite with respect to the inner product g, we have that ζH^α is too, and therefore we get

$$g(\zeta^{-1} H^{-\alpha/2} u, \zeta H^{\alpha/2} v) \leq \frac{1}{2} \|\zeta^{-1} H^{-\alpha/2} u\|_g^2 + \frac{1}{2} \|\zeta H^{\alpha/2} v\|_g^2 = \frac{1}{2\zeta^2} \|u\|_{H^{-\alpha}}^2 + \frac{\zeta^2}{2} \|v\|_{H^\alpha}^2.$$

□

B.1 Additional Results on Different Numerical Retractions

In this section, we present numerical results comparing several possible retraction choices on the Stiefel manifold. In Figure 2, we compare matrix size against GPU wall-clock time and final error. In particular, we compare QR decomposition, polar decomposition, a direct solver for the Cayley linear system (Cayley-Direct), an iterative method for the Cayley linear system that employs the Sherman–Morrison–Woodbury formula (Cayley-SMW), the fixed-point iteration (Cayley-FP), and the Newton–Schulz iteration.

C Proof of Proposition 4.3

Let $(B, A) \in \mathcal{F}$ and consider

$$\begin{aligned} \|\nabla(\mathcal{L} \circ \Phi)(tB, t^{-1} A)\|^2 &= \|\, \mathrm{D}\, \Phi(tB, t^{-1} A)^\top \nabla \mathcal{L}(W)\|^2 \\ &= \|tB^\top \nabla \mathcal{L}(W)\|^2 + \|\nabla \mathcal{L}(W) A^\top t^{-1}\|^2 \xrightarrow[t \to 0]{} +\infty, \end{aligned}$$

and therefore $\|\nabla(\mathcal{L} \circ \Phi)\|_{L^\infty(\mathcal{F})} = +\infty$.

For the second claim, fix A and B such that $\tilde{\Phi}(B, A) = W$. Then, we have

$$\sup_{(B', A') \in \widetilde{\mathcal{F}}} \|\nabla(\mathcal{L} \circ \widetilde{\Phi})(B', A')\|^2 = \sup_{O \in \mathrm{St}(r, r)} \|\nabla(\mathcal{L} \circ \widetilde{\Phi})(BO, O^\top A)\|^2 < +\infty,$$

because of compactness of $\mathrm{St}(r, r)$ and continuity.

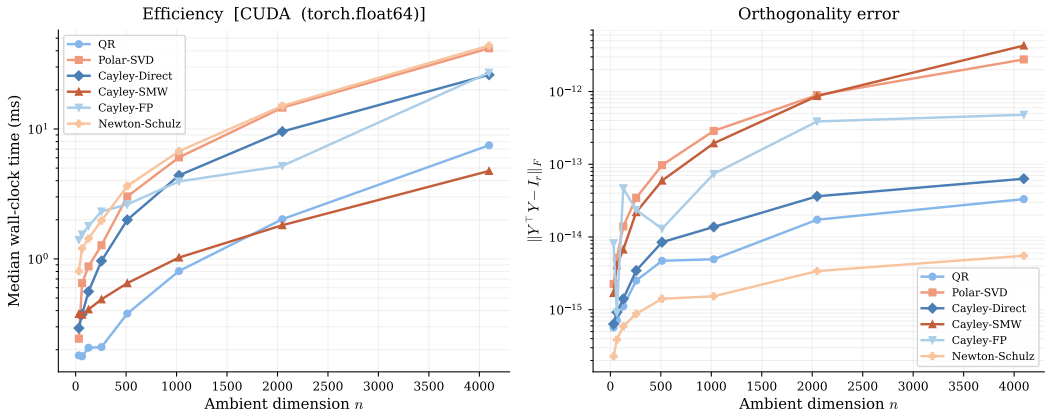

Figure 2: Comparison of different numerical retraction methods.

Configuration	Stiefel-AdamW	AdamW	Scaled AdamW	GeoLoRA
Learning rate	8×10^{-3}	8×10^{-3}	8×10^{-3}	8×10^{-3}
Weight decay	10^{-4}	10^{-2}	10^{-2}	10^{-4}
Learning rate schedule	Linear	Linear	Linear	Linear
(β_1, β_2)	$(0.98, 0.98)$	$(0.9, 0.999)$	$(0.7, 0.8)$	$(0.98, 0.98)$

Table 4: Hyperparameters for GPT2 LoRA fine-tuning on E2E.

D Additional Experimental Details

D.1 GPT2 E2E Fine-Tuning

We use hyperparameters tuned as in (Zhang & Pilanci, 2024), as reported in Table 4.