SPARQ: A SUBSPACE-PRESERVING MIXED PRECISION LMO OPTIMIZER

Anonymous authors
Paper under double-blind review

ABSTRACT

Optimizer-state memory remains a major bottleneck in large language model training. Existing low-bit state-compression methods do not explicitly account for the geometric sensitivity of linear-minimization-oracle (LMO) updates. This limitation is particularly important for Scion, whose spectral updates depend on the singular subspaces of persistent exponential-moving-average (EMA) states. We study Subspace-Preserving Adaptive Representation Quantization (SPARQ), a Scion-oriented mixed-precision state-compression method. SPARQ stores a more accurate low-rank component of each matrix-shaped EMA state and compresses the residual more aggressively, while applying the spectral LMO to matrix-shaped parameters and the root-mean-square (RMS) LMO to one-dimensional parameters. Our analysis shows that bounded reconstruction error preserves the decaying terms of the full-precision convergence bound while introducing an explicit error floor. Experiments with models up to 3B parameters show that SPARQ remains close to full-precision Scion, improves over uniform 4-bit integer (INT4) state storage, and reduces Scion optimizer-state memory by up to $6.9\times$. By reducing optimizer-state memory while preserving training quality, SPARQ can facilitate the training of larger models under memory-constrained settings.

1 INTRODUCTION

Large language model (LLM) pretraining has steadily shifted toward lower-precision arithmetic, from mixed-precision 16-bit floating-point (FP16)/bfloat16 (BF16) training (Micikevicius et al., 2018) to 8-bit floating-point (FP8) pipelines (Micikevicius et al., 2022; Peng et al., 2023; DeepSeek-AI, 2024). More recently, NVIDIA's 4-bit floating-point format (NVFP4) has achieved accuracy comparable to FP8 at trillion-token training scales (NVIDIA, 2025). These advances mainly reduce the cost of model weights, activations, and forward–backward computation. Optimizer states, however, are a separate persistent memory component and are often maintained in BF16 or 32-bit floating-point (FP32) even when the rest of the pipeline uses FP8 or 4-bit floating-point (FP4) (Peng et al., 2023; NVIDIA, 2025). Compressing them is challenging because they accumulate gradient information over many steps, allowing quantization errors to persist and compound, especially in the first- and second-moment estimates used by Adam with decoupled weight decay (AdamW) (Loshchilov & Hutter, 2017). Existing approaches use factorized second-moment estimates (Shazeer & Stern, 2018), block-wise 8-bit quantization (Dettmers et al., 2021), dynamic-range-aware 4-bit compression (Li et al., 2023), FP8 first-moment storage (Peng et al., 2023), or low-rank gradient projection before moment accumulation (Zhao et al., 2024). These methods primarily compress adaptive statistics, rather than optimizer states whose geometry determines the update direction.

A complementary line of work reduces optimizer-state memory by changing the optimizer geometry. Muon (Jordan et al., 2024) orthogonalizes momentum for hidden-layer matrices using Newton–Schulz iterations and uses AdamW for the remaining parameters. This design can achieve comparable convergence with a smaller optimizer state (Liu et al., 2025). MuonGRASP (Wu et al., 2026) subsequently introduced a subspace-preserving state-compression recipe for low-bit Muon, combining a warm-started low-rank-plus-residual decomposition with higher precision for the retained factors and more aggressive residual quantization. Other studies have explored block-wise schemes for low-precision Muon (Gupta et al., 2025). These results suggest that linear-minimization-oracle (LMO) optimizers can benefit from low-bit state storage, but they also reveal a central challenge: their updates are determined by the geometry of the optimizer state, so small quantization errors

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.

can cause large changes in the update direction. Pethick et al. (2025) generalized Muon into Scion, a family of LMO optimizers that supports different geometries across parameter structures. However, we find that Scion faces a severe error-amplification problem under low-bit state storage. Directly quantizing its exponential-moving-average (EMA) states can perturb the singular subspaces used by the spectral LMO, causing disproportionately large errors in the resulting update directions. We therefore transfer the subspace-preserving state-compression design of MuonGRASP to low-bit Scion and study the additional routing and theoretical issues that arise when the spectral LMO is applied to every matrix-shaped parameter.

Building on the above analysis, we examine how quantization errors propagate through the common spectral LMO used for matrix-shaped parameters. The spectral map can substantially amplify perturbations of the EMA buffer through its singular subspaces. This observation motivates a low-bit representation that preserves spectral subspaces across matrix-shaped parameters. Our goal is to reduce optimizer-state memory while preserving training performance as much as possible. We therefore ask:

"Can an LMO optimizer be compressed without compromising its training dynamics?"

Contributions. (1) We analyze how low-bit EMA reconstruction errors perturb the singular subspaces and update directions of spectral LMOs, explaining the sensitivity of uniform 4-bit integer (INT4) state storage. (2) We develop Subspace-Preserving Adaptive Representation Quantization (SPARQ), a subspace-preserving mixed-precision state representation for Scion. It applies the spectral LMO to all matrix-shaped parameters, uses the root-mean-square (RMS) LMO with FP32 states for one-dimensional parameters, and stores the dominant low-rank component more accurately than the residual. (3) We derive Scion-specific bounds for storage drift, convergence, and spectral-LMO sensitivity. Experiments with models up to 3B parameters show that SPARQ remains close to full-precision Scion, improves over uniform INT4 storage, and reduces optimizer-state memory by up to $6.9\times$.

2 Preliminaries

We follow the norm-constrained linear minimization oracle (LMO) framework of Pethick et al. (2025). For the unit ball $\mathcal{D} = \{x : \|x\| \le 1\}$, define

$$\mathrm{lmo}(z) \in \arg\min_{x \in \mathcal{D}} \langle z, x \rangle. \tag{1}$$

Scion maintains an exponential moving average (EMA) of stochastic gradients,

$$G_k = \alpha_k g_k + (1 - \alpha_k) G_{k-1}, \tag{2}$$

and uses it in either unconstrained or constrained dynamics:

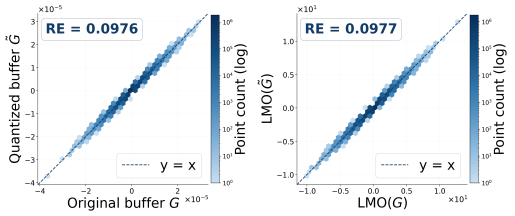
$$W_{k+1} = \begin{cases} W_k + \gamma_k \rho \,\mathrm{lmo}(G_k), & \text{unconstrained,} \\ (1 - \gamma_k) W_k + \gamma_k \rho \,\mathrm{lmo}(G_k), & \text{constrained.} \end{cases} \tag{3}$$

Here, ρ is the layerwise radius. In our implementation, every matrix-shaped parameter is assigned the scaled spectral LMO. For a matrix buffer with reduced singular value decomposition (SVD) $G = U\Sigma V^\top$, this direction is $-UV^\top$. One-dimensional parameters use $-G/\|G\|_{\mathrm{RMS}}$, where $\|G\|_{\mathrm{RMS}} = \|G\|_2/\sqrt{d}$. Fixed dimension-dependent factors follow Pethick et al. (2025) and do not change these geometries.

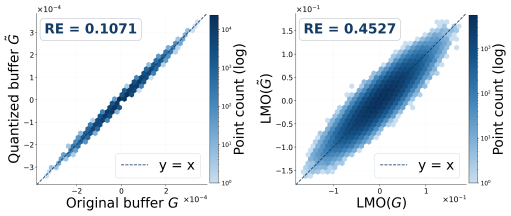
2.1 Why Naive INT4 Quantization Fails

We use two controlled references. 32-bit Scion uses the same uniform spectral routing for every matrix-shaped parameter and the RMS routing for one-dimensional parameters, and stores its EMA in FP32. 4-bit Scion keeps the same routing, radii, and update rule but replaces the EMA with uniform group-wise INT4 storage. Their comparison therefore isolates the effect of low-bit state storage. The central limitation of 4-bit Scion is that a small buffer reconstruction error need not imply a small LMO update error. We measure this discrepancy using the Frobenius-norm relative error (RE)

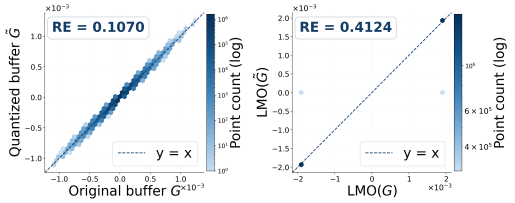
$$\mathrm{RE}(X, \widetilde{X}) := \frac{\|\widetilde{X} - X\|_F}{\|X\|_F}, \tag{4}$$



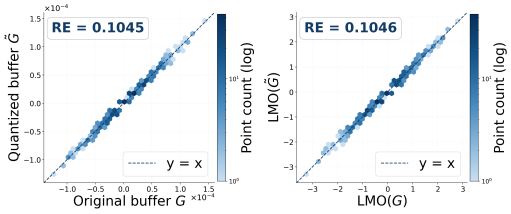
(a) Embedding matrix (spectral): $\mathrm{RE} = 0.0976 \rightarrow 0.0977$.



(b) Hidden matrix (spectral): $\mathrm{RE} = 0.1071 \rightarrow 0.4527$.



(c) Output head (spectral): $\mathrm{RE} = 0.1070 \rightarrow 0.4124$.



(d) Norm/bias (RMS): $\mathrm{RE} = 0.1045 \rightarrow 0.1046$.

Figure 1: Relative error before and after LMO routing with uniform INT4 EMA storage (seed 0, group size 128). Matrices use the spectral LMO; vector-shaped norm/bias tensors use the RMS LMO.

measured before and after applying its prescribed routing. Across parameter tensors, the buffer-level errors are similar, but the update errors depend strongly on the tensor and its LMO geometry. Figure 1 shows a representative run with seed 0 and group size 128. The embedding matrix changes from 0.0976 to 0.0977 after the spectral LMO, and the one-dimensional norm/bias parameters change from 0.1045 to 0.1046 after the RMS LMO. In contrast, the hidden matrix changes from 0.1071 to 0.4527 (a $4.23\times$ amplification), while the output head changes from 0.1070 to 0.4124 (a $3.85\times$ amplification); both matrix tensors use the same spectral LMO. Thus, uniform routing does not make the update equally insensitive across tensors: perturbations can affect the singular subspaces that determine the spectral update. Accurate buffer reconstruction alone therefore does not guarantee an accurate LMO update. This motivates the subspace-preserving representation introduced in Section 3.

3 SPARQ: Subspace-Preserving Low-Bit LMO Optimizer

The state-compression construction follows the subspace-preserving design of MuonGRASP, while its routing and analysis are adapted to the Scion family. SPARQ applies a common routing rule to all matrix-shaped parameters. Every tensor of order at least two is reshaped by retaining its leading dimension and flattening the remaining dimensions; we denote this map by view_b. The resulting matrix is assigned the appropriately scaled spectral LMO and stored in a subspace-preserving low-bit format. One-dimensional parameters retain the RMS LMO and an FP32 EMA because they constitute a negligible fraction of the optimizer state. The layerwise radii are inherited from Scion. The unconstrained and constrained updates in Eq. 3 are two dynamics of the same optimizer and use the same stored state: matrix-shaped states use the low-bit representation, whereas one-dimensional states remain in FP32. For one-dimensional parameters, view_b is the identity.

For a matrix-shaped pre-storage buffer $B_k \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$, let

$$\kappa = \max\{1, \lfloor \eta \min\{d_{\mathrm{out}}, d_{\mathrm{in}}\} \rfloor\}, \tag{5}$$

where $\eta \in (0, 1]$ is the rank ratio. SPARQ performs one warm-started subspace iteration,

$$Q_k = \mathrm{orth}(\widetilde{R}_{k-1}), \qquad P_k = \mathrm{orth}(B_k Q_k), \qquad R_k = B_k^\top P_k, \tag{6}$$

where $\mathrm{orth}(\cdot)$ returns an orthonormal basis obtained by a reduced orthogonal–triangular decomposition (QR decomposition). It then separates the captured subspace from the residual:

$$B_k = P_k R_k^\top + B_k^{\mathrm{res}}, \qquad B_k^{\mathrm{res}} := B_k - P_k R_k^\top. \tag{7}$$

Algorithm 1 SPARQ update

Require: Parameters $\{W_k^{(b)}\}_{b=1}^B$, gradients $\{g_k^{(b)}\}_{b=1}^B$, states $\{S_{k-1}^{(b)}\}_{b=1}^B$, momentum α_k, stepsize γ_k, radii $\{\rho_b\}_{b=1}^B$, rank ratio η, and mode $m \in \{Unconstrained, Constrained\}$; for constrained mode, $\|W_k^{(b)}\|_b \le \rho_b$ and $\gamma_k \in [0,1]$

for $b = 1, \dots, B$ **do**
  $X_k^{(b)} \leftarrow \mathrm{view}_b(W_k^{(b)})$; $h_k^{(b)} \leftarrow \mathrm{view}_b(g_k^{(b)})$ ▷ uniform matrix view
  if $\mathrm{order}(W_k^{(b)}) \ge 2$ **then**
    $(\widetilde{B}_{k-1}^{(b)}, \widetilde{R}_{k-1}^{(b)}) \leftarrow \mathrm{Decompress}(S_{k-1}^{(b)})$ ▷ Eq. 8
  else
    $\widetilde{B}_{k-1}^{(b)} \leftarrow S_{k-1}^{(b)}$
  end if
  $B_k^{(b)} \leftarrow \alpha_k h_k^{(b)} + (1-\alpha_k)\widetilde{B}_{k-1}^{(b)}$ ▷ Eq. 9
  if $\mathrm{order}(W_k^{(b)}) \ge 2$ **then**
    $O_k^{(b)} \leftarrow \rho_b \,\mathrm{lmo}_{\mathrm{spectral}}(B_k^{(b)})$ ▷ SPARQ routing; Eq. 1
    $S_k^{(b)} \leftarrow \mathcal{C}_\eta(B_k^{(b)}, \widetilde{R}_{k-1}^{(b)})$ ▷ SPARQ storage; Eqs. 5–7
  else
    $O_k^{(b)} \leftarrow \rho_b \,\mathrm{lmo}_{\mathrm{RMS}}(B_k^{(b)})$ ▷ Eq. 1
    $S_k^{(b)} \leftarrow B_k^{(b)}$ in FP32
  end if
  if $m = Unconstrained$ **then**
    $X_{k+1}^{(b)} \leftarrow X_k^{(b)} + \gamma_k O_k^{(b)}$ ▷ unconstrained; Eq. 3
  else
    $X_{k+1}^{(b)} \leftarrow (1-\gamma_k)X_k^{(b)} + \gamma_k O_k^{(b)}$ ▷ constrained; Eq. 3
  end if
  $W_{k+1}^{(b)} \leftarrow \mathrm{view}_b^{-1}(X_{k+1}^{(b)})$
end for
return $\{W_{k+1}^{(b)}, S_k^{(b)}\}_{b=1}^B$

The factors and residual are stored at low precision and reconstructed as

$$\widetilde{B}_k = \widetilde{P}_k \widetilde{R}_k^\top + \widetilde{B}_k^{\mathrm{res}}, \qquad e_k := \widetilde{B}_k - B_k. \tag{8}$$

Our implementation uses higher precision for the low-rank factors than for the dense residual; specifically, we use 8-bit integer (INT8) factors and an INT4 residual. The compressor includes the low-bit quantization and packing of the factors and the residual. Any analysis below is independent of these bit widths and quantizers: it depends only on the reconstruction error e_k. The reconstructed state enters the next EMA,

$$B_{k+1} = \alpha_{k+1} g_{k+1} + (1-\alpha_{k+1})\widetilde{B}_k. \tag{9}$$

Thus, the current LMO is evaluated from B_k before this buffer is compressed; quantization affects only subsequent iterations. At initialization, the EMA is zero, the right factor is random and full rank, and $\alpha_1 = 1$. The initial state S_0 is chosen so that decompression returns this zero EMA and the initialized right factor. If a buffer is zero, its LMO output is set to zero; if the warm-started subspace is rank deficient, the subspace is padded as needed before the next compression step. Algorithm 1 summarizes the complete update. Let $\mathcal{C}_\eta$ denote the subspace-preserving compressor defined by Eqs. 6–7, including the low-bit quantization and packing of the factors and residual, and let $\mathrm{Decompress}$ denote its reconstruction map. Light-gray steps highlight the SPARQ-specific implementation steps.

4 Theoretical Analysis

We establish three results. First, uniform spectral routing preserves the convergence-rate orders of full-precision Scion under the corresponding norm-dependent assumptions. Second, for both

unconstrained and constrained SPARQ with constant momentum, low-bit state storage preserves the decaying terms in the exact-storage bounds and adds an explicit reconstruction-dependent error floor. Third, we bound the sensitivity of the spectral-LMO update to buffer perturbations under a standard full-rank condition. Mathematical background and complete proofs are provided in Appendix C.

Let $x \in \mathcal{X}$ be the joint parameter, where $\mathcal{X}$ is a finite-dimensional real vector space equipped with a norm $\|\cdot\|$, dual norm $\|\cdot\|_*$, and norm ball $D := \{x : \|x\| \le \rho\}$. We write $\|\cdot\|_2$ for the Euclidean norm, which coincides with the Frobenius norm on matrix-shaped blocks, and define

$$\rho_2 := \max_{x \in D} \|x\|_2, \qquad D_2 := \max_{x,y \in D} \|x - y\|_2, \qquad \zeta := \sup_{x \neq 0} \frac{\|x\|_*}{\|x\|_2}, \qquad \mathrm{lmo}(z) \in \arg\min_{x \in D} \langle z, x \rangle. \tag{10}$$

Throughout this section, lmo denotes linear minimization over the radius-ρ ball D. For SPARQ, $\|\cdot\|$ is the product norm induced by its LMO routing.

Assumption 4.1 (Pethick et al., 2025). *(i) The objective f is bounded below by $f^\star$, and, for all $x, y \in \mathcal{X}$, its gradient satisfies*

$$\|\nabla f(x) - \nabla f(y)\|_* \le L\|x - y\|, \qquad \|\nabla f(x) - \nabla f(y)\|_2 \le L_2\|x - y\|_2. \tag{11}$$

(ii) If $\mathcal{F}_k$ denotes the history before sampling g_k, then

$$\mathbb{E}[g_k \mid \mathcal{F}_k] = \nabla f(x^k), \qquad \mathbb{E}[\|g_k - \nabla f(x^k)\|_2^2 \mid \mathcal{F}_k] \le \sigma^2. \tag{12}$$

Let B_k denote the current pre-storage EMA buffer. After the update is computed, this buffer is quantized and reconstructed as $\widetilde{B}_k := \mathrm{DEQUANT}(\mathrm{QUANT}(B_k)) = B_k + e_k$. Thus, with $B_0 = e_0 = 0$ and $\alpha_1 = 1$, the buffer evolves as

$$B_k = \alpha_k g_k + (1 - \alpha_k)(B_{k-1} + e_{k-1}), \qquad e_k := \widetilde{B}_k - B_k. \tag{13}$$

To isolate storage error, define the coupled exact-storage reference

$$G_k^\star = \alpha_k g_k + (1 - \alpha_k) G_{k-1}^\star, \qquad G_0^\star = 0, \tag{14}$$

using the same realized gradient sequence as the quantized trajectory. Let $\mathcal{Q}_k$ denote the sigma-field generated by the information available immediately before the state at step k is quantized, including B_k.

Assumption 4.2. *There exists $q \ge 0$ such that, for every k,*

$$\mathbb{E}\big[\|e_k\|_2^2 \mid \mathcal{Q}_k\big] \le q^2 \qquad \textit{almost surely}. \tag{15}$$

Following the bounded-second-moment treatment of quantization error in Alistarh et al. (2017) and Wu et al. (2018), we impose the analogous condition on the reconstruction error of the compressed EMA state. Equation 15 is used in Lemmas 4.6 and 4.7 to control the accumulated storage drift, which yields the quantization-dependent terms in Theorems 4.8 and 4.10. The following lemma records the norm-ball LMO properties used throughout the analysis.

Lemma 4.3 (Norm-ball LMO identities). *Let $\|\cdot\|$ be any norm on $\mathcal{X}$, let $\rho > 0$, and let $D = \{x : \|x\| \le \rho\}$. Then, for every $z \in \mathcal{X}$,*

$$\langle z, \mathrm{lmo}(z) \rangle = -\rho\|z\|_*, \qquad \|\mathrm{lmo}(z)\| \le \rho, \qquad \|\mathrm{lmo}(z)\|_2 \le \rho_2. \tag{16}$$

Proof of Lemma 4.3.

4.1 Convergence of SPARQ

Let $\|\cdot\|_{\mathrm{SP}}$ denote the spectral-routing product (SP) norm induced by the full-precision reference. This reference assigns the scaled spectral LMO to every matrix-shaped block, retains the RMS LMO for one-dimensional blocks, and stores all EMA buffers without compression. We apply the assumptions and algorithmic conditions of Theorems 5.4–5.7 in Pethick et al. (2025) with respect to $\|\cdot\|_{\mathrm{SP}}$. Theorems 5.4 and 5.5 cover the unconstrained dynamics with constant and vanishing momentum, respectively, while Theorems 5.6 and 5.7 cover the corresponding constrained dynamics.

Proposition 4.4. *Under the preceding conditions, spectral-LMO reassignment preserves the convergence-rate orders of Theorems 5.4–5.7 in Pethick et al. (2025). All norm-dependent constants are evaluated under $\|\cdot\|_{\mathrm{SP}}$. Proof of Proposition 4.4.*

To characterize the effect of quantized storage, define $\epsilon_k := B_k - G_k^\star$. The next three lemmas describe its evolution under the momentum schedules considered in our analysis.

Lemma 4.5. *We have $\epsilon_1 = 0$ and, for every $k \geq 2$,*

$$\epsilon_k = (1-\alpha_k)(\epsilon_{k-1} + e_{k-1}). \tag{17}$$

Proof of Lemma 4.5.

Lemma 4.6. *Let $\alpha_k = \alpha \in (0,1)$ for every $k \geq 2$, with $\alpha_1 = 1$, and suppose Assumption 4.2 holds. Then, for every $k \geq 1$,*

$$\left(\mathbb{E}\|\epsilon_k\|_2^2\right)^{1/2} \leq \frac{(1-\alpha)q}{\alpha}\left(1-(1-\alpha)^{k-1}\right) \leq \frac{(1-\alpha)q}{\alpha}. \tag{18}$$

Proof of Lemma 4.6.

Lemma 4.7. *Let $\alpha_k = 1/\sqrt{k}$ and suppose Assumption 4.2 holds. Then, for every $k \geq 1$,*

$$\left(\mathbb{E}\|\epsilon_k\|_2^2\right)^{1/2} \leq 2q(\sqrt{k}-1). \tag{19}$$

Proof of Lemma 4.7.

Unlike Lemma 4.6, Lemma 4.7 does not give a uniform drift bound when q is fixed. We therefore state the SPARQ convergence results below for constant momentum. Extending the vanishing-momentum guarantees requires the reconstruction error to decrease with the training horizon. For the unconstrained branch, the parameter update is

$$x^{k+1} = x^k + \gamma\,\mathrm{lmo}(B_k). \tag{20}$$

Theorem 4.8 (Convergence of unconstrained SPARQ)**.** *Suppose Assumptions 4.1 and 4.2 hold. Let the training horizon satisfy $n \in \mathbb{N}_+$, and let the iterates follow Eqs. 13 and 20 with $\gamma = 1/\sqrt{n}$, $\alpha_1 = 1$, and $\alpha_k = \alpha \in (0,1)$ for $k \geq 2$. If $\bar{x}^n$ is drawn independently and uniformly from $\{x^1, \ldots, x^n\}$ after the run, then*

$$\begin{aligned}\mathbb{E}[\|\nabla f(\bar{x}^n)\|_*] \leq{}& \frac{\mathbb{E}[f(x^1)] - f^\star}{\rho\sqrt{n}} + \frac{L\rho}{2\sqrt{n}} \\ &+ \left(\frac{\rho_2}{\rho} + \zeta\right)\left[\frac{(1-\alpha)L_2\rho_2}{\alpha\sqrt{n}} + \left(\sqrt{\frac{\alpha}{2-\alpha}} + \frac{1}{\alpha n}\right)\sigma + \frac{(1-\alpha)q}{\alpha}\right].\end{aligned} \tag{21}$$

Proof of Theorem 4.8.

For the constrained branch, let $x^1 \in D$ and use

$$x^{k+1} = (1-\gamma)x^k + \gamma\,\mathrm{lmo}(B_k). \tag{22}$$

This update preserves feasibility. Define the Frank–Wolfe gap

$$\mathcal{G}_D(x) := \max_{u \in D} \langle \nabla f(x), x - u\rangle. \tag{23}$$

Lemma 4.9. *Suppose Assumption 4.1(i) holds and the iterates follow Eq. 22. Then, for every k,*

$$\gamma\mathcal{G}_D(x^k) \leq f(x^k) - f(x^{k+1}) + \gamma D_2\left(\|\lambda_k\|_2 + \|\epsilon_k\|_2\right) + \frac{L_2\gamma^2 D_2^2}{2}, \tag{24}$$

where $\lambda_k := G_k^\star - \nabla f(x^k)$ and $\epsilon_k := B_k - G_k^\star$. Proof of Lemma 4.9.

Theorem 4.10 (Convergence of constrained SPARQ)**.** *Suppose Assumptions 4.1 and 4.2 hold. Let $n \in \mathbb{N}_+$ and $x^1 \in D$, and let the iterates follow Eqs. 13 and 22 with $\gamma = 1/\sqrt{n}$, $\alpha_1 = 1$, and $\alpha_k = \alpha \in (0,1)$ for $k \geq 2$. If $\bar{x}^n$ is drawn independently and uniformly from $\{x^1, \ldots, x^n\}$ after the run, then*

$$\begin{aligned}\mathbb{E}[\mathcal{G}_D(\bar{x}^n)] \leq{}& \frac{\mathbb{E}[f(x^1)] - f^\star}{\sqrt{n}} + \frac{L_2 D_2^2}{2\sqrt{n}} \\ &+ D_2\left[\frac{(1-\alpha)L_2 D_2}{\alpha\sqrt{n}} + \left(\sqrt{\frac{\alpha}{2-\alpha}} + \frac{1}{\alpha n}\right)\sigma + \frac{(1-\alpha)q}{\alpha}\right].\end{aligned} \tag{25}$$

Proof of Theorem 4.10.

The two convergence theorems are representation-agnostic: the storage representation enters only through the reconstruction-error level q. Setting $q = 0$ recovers the corresponding exact-storage guarantees. Under constant momentum, quantization preserves all terms that decay with the training horizon and adds an explicit error floor proportional to q.

4.2 SPECTRAL-LMO SENSITIVITY

Quantized storage perturbs B_k relative to $G_k^\star$, and the spectral LMO can amplify this perturbation. We therefore bound the resulting update deviation under a suitable spectral condition. Consider a matrix-shaped block with original dimensions $d_{\text{out}} \times d_{\text{in}}$. Let $A^\sharp = A$ if $d_{\text{out}} \geq d_{\text{in}}$ and $A^\sharp = A^\top$ otherwise, and if $A = U\Sigma V^\top$ is a reduced singular value decomposition, define its unitary polar factor as $\Phi(A) := UV^\top$. Set $m := \max\{d_{\text{out}}, d_{\text{in}}\}$, $p := \min\{d_{\text{out}}, d_{\text{in}}\}$, and write $c_\ell := \sqrt{d_{\text{out}}/d_{\text{in}}}$. If $\rho_\ell > 0$ is the radius assigned to this block, define the radius-aware spectral LMO by

$$\text{LMO}^{(\rho_\ell)}_{\text{spectral}}(A) := -\rho_\ell c_\ell \Phi(A). \tag{26}$$

The uppercase operator in Eq. 26 includes the block radius, whereas the lowercase unit-ball operator used in Algorithm 1 does not.

Lemma 4.11 (Li, 1995)**.** *Let $A, \widetilde{A} \in \mathbb{R}^{m\times p}$ with $m \geq p$, and suppose both matrices have full column rank. Their unitary polar factors satisfy the following bounds.*

(i) Square case ($m = p$). For any unitarily invariant norm $|||\cdot|||$, including the Frobenius norm,

$$|||\Phi(\widetilde{A}) - \Phi(A)||| \leq \frac{2}{\sigma_{\min}(A) + \sigma_{\min}(\widetilde{A})} |||\widetilde{A} - A|||. \tag{27}$$

(ii) Rectangular case ($m > p$).

$$\|\Phi(\widetilde{A}) - \Phi(A)\|_F \leq \left[\left(\frac{2}{\sigma_{\min}(A) + \sigma_{\min}(\widetilde{A})}\right)^2 + \left(\frac{1}{\max\{\sigma_{\min}(A), \sigma_{\min}(\widetilde{A})\}}\right)^2\right]^{1/2} \|\widetilde{A} - A\|_F. \tag{28}$$

Proof of Lemma 4.11.

Proposition 4.12. *Fix a block and an iteration k, and suppose $(G_k^\star)^\sharp$ and $B_k^\sharp$ have full column rank. If $m > p$, then,*

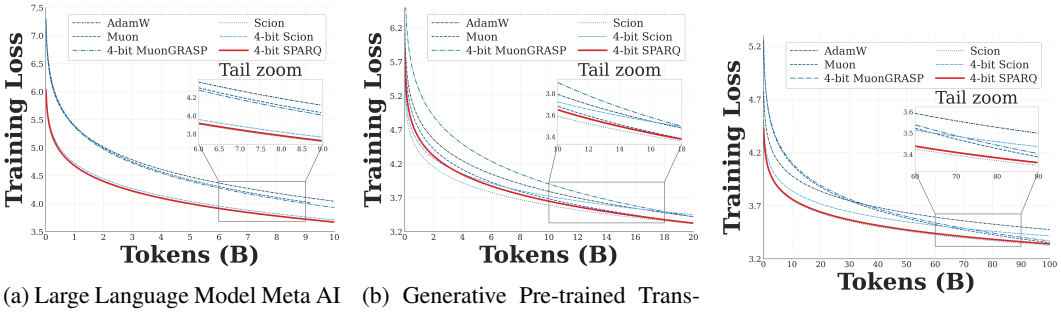
$$\begin{aligned} &\|\,\text{LMO}^{(\rho_\ell)}_{\text{spectral}}(B_k) - \text{LMO}^{(\rho_\ell)}_{\text{spectral}}(G_k^\star)\|_F \\ &\leq \rho_\ell c_\ell \sqrt{\left(\frac{2}{\sigma_{\min}((G_k^\star)^\sharp) + \sigma_{\min}(B_k^\sharp)}\right)^2 + \left(\frac{1}{\max\{\sigma_{\min}((G_k^\star)^\sharp), \sigma_{\min}(B_k^\sharp)\}}\right)^2}\, \|B_k - G_k^\star\|_F. \end{aligned} \tag{29}$$

If $m = p$, Eq. 27 gives the corresponding pointwise estimate with coefficient $2/[\sigma_{\min}((G_k^\star)^\sharp) + \sigma_{\min}(B_k^\sharp)]$. Proof of Proposition 4.12.

Proposition 4.12 connects the accumulated buffer deviation $B_k - G_k^\star$ to the resulting spectral-LMO update deviation. It shows that the update perturbation is controlled when the two coupled buffers stay away from rank deficiency, while also explaining why a buffer perturbation can be amplified when their smallest singular values are small. This statement uses exactly the pointwise full-rank conditions of Li (1995); no uniform spectral-floor assumption over the training trajectory is imposed.

5 EXPERIMENTS

We study SPARQ in the configuration called 4-bit SPARQ, which stores low-rank factors in INT8, the dense residual in INT4, and one-dimensional EMA states in FP32. We evaluate pre-training loss, zero-shot accuracy, and optimizer-state memory, while fixing routing and training settings to isolate state representation; additional ablations are given in Appendix D. We test whether 4-bit SPARQ matches full-precision Scion, improves over uniform INT4 storage, and works across models and optimizers. We pre-train LLaMA 124M, GPT-2 350M, and LLaMA 3B on DCLM (Li et al., 2024) with context length 256 and BF16. All Scion-family methods use spectral LMO routing for matrix-shaped parameters and RMS routing for one-dimensional norm and bias parameters. Full-precision Scion stores the EMA in full precision, 4-bit Scion uses uniform group-wise INT4 storage, and 4-bit SPARQ uses the subspace-preserving representation. We set $\alpha = 0.1$, $\alpha_1 = 1$, $\rho_{\text{hidden}} = 50$, and



(a) Large Language Model Meta AI (LLaMA) 124M

(b) Generative Pre-trained Transformer 2 (GPT-2) 350M

(c) LLaMA 3B

Figure 2: Training loss as a function of the number of training tokens on DataComp for Language Models (DCLM) for six optimizers on LLaMA 124M, GPT-2 350M, and LLaMA 3B.

use a warmup-stable-decay schedule with warmdown fraction 0.285; quantized runs use $s = 128$, and 4-bit SPARQ uses $\eta = 1/16$. Muon comparisons include full-precision Muon and 4-bit MuonGRASP for memory, training, and downstream results. Unless otherwise noted, experiments use multiple seeds and report aggregated results. We use `lm-evaluation-harness` (Gao et al., 2024) for zero-shot accuracy. Within each model configuration, methods share the same initialization, data order, training budget, schedule, weight decay, batch construction, and evaluation checkpoint.

5.1 Pre-training Results

Figure 2 shows the training-loss curves for six optimizers across three model configurations. Across all three configurations, 4-bit SPARQ remains close to full-precision Scion and generally stays below the corresponding 4-bit Scion curve, with the separation becoming more visible later in training. This pattern is consistent with the intended benefit of preserving the subspace information used by the spectral update. The reported comparisons are based on repeated runs, and the observed alignment between 4-bit SPARQ and full-precision Scion is consistent across these runs. The figure also provides a cross-family comparison with 4-bit MuonGRASP. Its training curves track full-precision Muon on LLaMA 124M and LLaMA 3B, whereas the gap is larger on GPT-2 350M. This comparison provides context for the behavior of subspace-aware state compression in the Muon family; it is not a direct evaluation of SPARQ applied to Muon. Overall, the results suggest that subspace-aware state representations are compatible with stable pre-training in both the Scion and Muon families.

5.2 Downstream Zero-Shot Evaluation

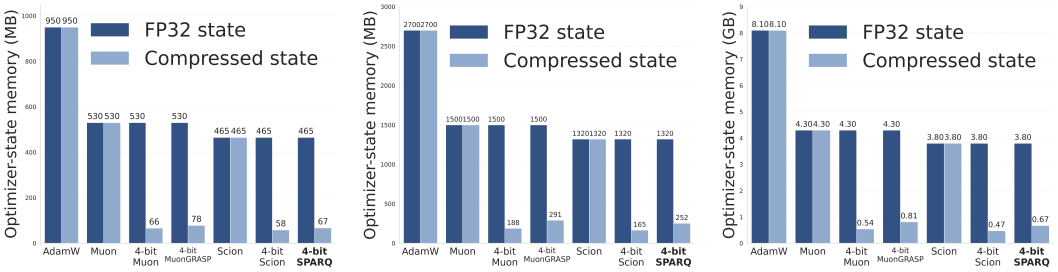
Because training loss is not a sufficient proxy for downstream quality, we evaluate the corresponding trained checkpoints in a zero-shot setting using `lm-evaluation-harness` on the AI2 Reasoning Challenge (ARC) subsets ARC-Challenge (ARC-C) and ARC-Easy (ARC-E), Boolean Questions (BoolQ), Language Modeling Broadened to Account for Discourse Appropriateness (LAMBADA), Physical Interaction: Question Answering (PIQA), ReAding Comprehension from Examinations (RACE), and HellaSwag. Unless otherwise noted, each reported value is the mean over repeated runs. We report the unweighted average accuracy across the seven tasks. Table 1 summarizes the results for LLaMA 124M, GPT-2 350M, and LLaMA 3B. Across all three model configurations, 4-bit SPARQ achieves the highest reported mean accuracy among the compared methods. Its advantage is particularly visible on LLaMA 124M, where it improves over uniform 4-bit Scion on several tasks. On GPT-2 350M and LLaMA 3B, 4-bit SPARQ also achieves the best average accuracy, although other methods remain strongest on some individual tasks.

5.3 Optimizer-State Memory

Figures 3a, 3b, and 3c report optimizer-state memory before and after compression. Within the Scion family, all variants use the same routing. We define the compression ratio as full-precision

Table 1: Zero-shot downstream accuracy (%) on LLaMA 124M, GPT-2 350M, and LLaMA 3B.

Optimizer	ARC-C	ARC-E	BoolQ	LAMBADA	PIQA	RACE	HellaSwag	Average
LLaMA 124M / DCLM, 10B tokens								
AdamW	21.8	34.2	**62.1**	7.2	56.1	23.0	25.8	32.9
Muon	21.1	34.9	57.7	11.3	57.9	23.0	26.2	33.2
4-bit MuonGRASP	21.6	35.0	48.5	10.5	**59.0**	23.1	26.0	32.0
4-bit Scion	21.9	37.4	49.1	19.4	**59.0**	24.7	26.4	34.0
4-bit SPARQ	**23.0**	**37.7**	46.6	**30.6**	58.9	**25.7**	**27.1**	**35.7**
GPT-2 350M / DCLM, 20B tokens								
AdamW	23.63	38.13	44.77	21.85	59.90	25.84	27.78	34.56
Muon	**24.49**	40.11	52.23	25.03	**62.19**	**26.51**	29.26	37.12
4-bit MuonGRASP	23.21	39.52	54.34	23.69	61.86	26.22	29.08	36.85
4-bit Scion	24.06	39.14	**57.28**	23.37	60.07	24.98	28.26	36.74
4-bit SPARQ	22.61	**40.61**	54.19	**26.47**	61.64	26.41	**29.45**	**37.34**
LLaMA 3B / DCLM, 100B tokens								
AdamW	24.66	41.58	52.87	25.42	62.13	27.27	30.63	37.79
Muon	25.17	**44.70**	54.65	30.31	**63.93**	26.41	32.47	39.66
4-bit MuonGRASP	24.74	42.72	54.22	29.58	63.00	27.37	32.47	39.16
4-bit Scion	24.83	41.54	**55.11**	27.85	63.00	27.56	30.82	38.67
4-bit SPARQ	**25.83**	44.53	53.67	**30.91**	62.84	**28.52**	**32.51**	**39.83**



(a) LLaMA 124M. (b) GPT-2 350M. (c) LLaMA 3B.

Figure 3: Measured optimizer-state memory before and after compression.

Scion memory divided by compressed memory. The full-precision Scion states use 465 MB, 1320 MB, and 3.80 GB on LLaMA 124M, GPT-2 350M, and LLaMA 3B, whereas 4-bit SPARQ uses 67 MB, 252 MB, and 0.67 GB. These values correspond to compression ratios of $465/67 = 6.9\times$, $1320/252 = 5.2\times$, and $3.80/0.67 = 5.7\times$, or memory reductions of 85.6%, 80.9%, and 82.4%, respectively. Thus, the largest compression ratio shown for 4-bit SPARQ is $6.9\times$. To compare storage efficiency with 4-bit MuonGRASP, we compare the compressed states directly: SPARQ uses 67 versus 78 MB, 252 versus 291 MB, and 0.67 versus 0.81 GB. Hence, the 4-bit MuonGRASP states are approximately $78/67 = 1.16\times$, $291/252 = 1.15\times$, and $0.81/0.67 = 1.21\times$ larger, meaning that SPARQ saves 14.1%, 13.4%, and 17.3% of the compressed-state memory, respectively. These results show that 4-bit SPARQ combines substantial compression relative to full-precision Scion with lower optimizer-state memory than 4-bit MuonGRASP.

6 Conclusion

We presented SPARQ, a low-bit LMO optimizer that applies spectral routing to matrix-shaped parameters and stores EMA states with subspace-preserving low-rank factors and a quantized residual. Under constant momentum and bounded reconstruction error, our analysis preserves the decaying terms of the full-precision convergence bounds, introduces an explicit error floor, and bounds the sensitivity of spectral-LMO updates to state perturbations under a full-rank condition. Across LLaMA 124M, GPT-2 350M, and LLaMA 3B, 4-bit SPARQ remains close to full-precision Scion, generally improves training loss and average zero-shot accuracy over uniform INT4 storage, and reduces estimated Scion state memory by up to $6.9\times$, enabling larger models training under fixed graphics processing unit (GPU) memory budgets.

AI USE STATEMENT

We used generative AI tools solely to assist with proofreading and polishing the language of this manuscript, including improving grammar, clarity, readability, and overall presentation. The tools were not used to generate research ideas, design experiments, generate synthetic datasets, formulate or prove mathematical claims, or produce substantive scientific content. All AI-assisted text was carefully reviewed, revised, and verified by the authors. The authors take full responsibility for the content and claims of this paper.

REFERENCES

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnović. QSGD: Communication-efficient SGD via gradient quantization and encoding. In Advances in Neural Information Processing Systems, volume 30, 2017.

DeepSeek-AI. DeepSeek-V3 technical report. arXiv preprint arXiv:2412.19437, 2024.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit optimizers via block-wise quantization. arXiv preprint arXiv:2110.02861, 2021. Published at ICLR 2022.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Aman Gupta, Rafael Celente, Abhishek Shivanna, D. T. Braithwaite, Gregory Dexter, Shao Tang, Hiroto Udagawa, Daniel Silva, Rohan Ramanath, and S. Sathiya Keerthi. Effective quantization of muon optimizer states. arXiv preprint arXiv:2509.23106, 2025.

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

Nicholas J. Higham. Computing the polar decomposition—with applications. SIAM Journal on Scientific and Statistical Computing, 7(4):1160–1174, 1986. doi: 10.1137/0907079.

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Proceedings of the 30th International Conference on Machine Learning, volume 28 of Proceedings of Machine Learning Research, pp. 427–435, 2013.

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

Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U. Stich, and Martin Jaggi. Error feedback fixes SignSGD and other gradient compression schemes. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pp. 3252–3261, 2019.

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

Bingrui Li, Jianfei Chen, and Jun Zhu. Memory efficient optimizers with 4-bit states. In Advances in Neural Information Processing Systems, volume 36, pp. 15136–15171, 2023.

Jeffrey Li, Alex Fang, Georgios Smyrnis, et al. DataComp-LM: In search of the next generation of training sets for language models. arXiv preprint arXiv:2406.11794, 2024.

Ren-Cang Li. New perturbation bounds for the unitary polar factor. SIAM Journal on Matrix Analysis and Applications, 16(1):327–332, 1995. doi: 10.1137/S0895479893256359.

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, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017. Published at ICLR 2019.

Yang Luo, Xiaozhe Ren, Zangwei Zheng, Zhuo Jiang, Xin Jiang, and Yang You. CAME: Confidence-guided adaptive memory efficient optimization. arXiv preprint arXiv:2307.02047, 2023.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In International Conference on Learning Representations, 2018.

Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, Naveen Mellempudi, Stuart Oberman, Mohammad Shoeybi, Michael Siu, and Hao Wu. FP8 formats for deep learning, 2022.

NVIDIA. Pretraining large language models with NVFP4. arXiv preprint arXiv:2509.25149, 2025.

Houwen Peng, Kan Wu, Yixuan Wei, Guoshuai Zhao, Yuxiang Yang, Ze Liu, Yifan Xiong, Ziyue Yang, Bolin Ni, Jingcheng Hu, Ruihang Li, Miaosen Zhang, Chen Li, Jia Ning, Ruizhe Wang, Zheng Zhang, Shuguang Liu, Joe Chau, Han Hu, and Peng Cheng. FP8-LM: Training FP8 large language models. arXiv preprint arXiv:2310.18313, 2023.

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

Artem Riabinin, Egor Shulgin, Kaja Gruntkowska, and Peter Richtárik. Gluon: Making muon & scion great again! (bridging theory and practice of LMO-based optimizers for LLMs). arXiv preprint arXiv:2505.13416, 2025.

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

Yupeng Su et al. MuonQ: Enhancing low-bit muon quantization via directional fidelity optimization. arXiv preprint arXiv:2605.11396, 2026.

Huaijin Wu, Bingrui Li, Yebin Yang, Yi Tu, Zhanpeng Zhou, Jianfei Chen, and Junchi Yan. Achieving low-bit muon through subspace preservation and grid quantization. In International Conference on Learning Representations, 2026. OpenReview id: g2l9bg9DWx.

Jiaxiang Wu, Weidong Huang, Junzhou Huang, and Tong Zhang. Error compensated quantized SGD and its applications to large-scale distributed optimization. In Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pp. 5325–5333, 2018.

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.

Zhenyu Zhang, Ajay Kumar Jaiswal, Lu Yin, Shiwei Liu, Jiawei Zhao, Yuandong Tian, and Zhangyang Wang. Q-GaLore: Quantized GaLore with INT4 projection and layer-adaptive low-rank gradients. In Conference on Parsimony and Learning, volume 280 of Proceedings of Machine Learning Research, pp. 1035–1050, 2025.

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.

Hanqing Zhu, Zhenyu Zhang, Wenyan Cong, Xi Liu, Sem Park, Vikas Chandra, Bo Long, David Z. Pan, Zhangyang Wang, and Jinwon Lee. APOLLO: SGD-like memory, AdamW-level performance. arXiv preprint arXiv:2412.05270, 2024.

Appendix Contents

A Related Work **14**

B Main Structure of SPARQ **14**

C Complete Theoretical Analysis **16**

C.1 Theoretical Preliminaries 16

C.2 Proof of Lemma 4.3 17

C.3 Proof of Proposition 4.4 18

C.4 Proof of Lemma 4.5 18

C.5 Proof of Lemma 4.6 19

C.6 Proof of Lemma 4.7 19

C.7 Proof of Theorem 4.8 19

C.8 Proof of Lemma 4.9 22

C.9 Proof of Theorem 4.10 22

C.10 Proof of Lemma 4.11 23

C.11 Proof of Proposition 4.12 24

D More Experiments **24**

D.1 Error Amplification Under Uniform Routing 24

D.2 Quantization Design Ablations 25

D.3 Memory and Computational Efficiency 27

A Related Work

Low-precision training and memory-efficient optimizer states. Large-model training has progressed from mixed FP16/BF16 computation to FP8 and, more recently, FP4 pipelines (Micikevicius et al., 2018; 2022; Peng et al., 2023; DeepSeek-AI, 2024; NVIDIA, 2025). Optimizer states remain harder to compress because Adam and AdamW maintain persistent first- and second-moment statistics (Kingma & Ba, 2015; Loshchilov & Hutter, 2017). Existing methods reduce this cost by factorizing or coarsening adaptive statistics (Shazeer & Stern, 2018; Luo et al., 2023; Zhang et al., 2024), quantizing the states (Dettmers et al., 2021; Li et al., 2023; Peng et al., 2023), or replacing dense statistics with low-dimensional projections (Zhao et al., 2024; Zhu et al., 2024). Q-GaLore further combines low-rank gradient projection with low-bit projection matrices and weights (Zhang et al., 2025). A parallel literature quantizes communicated gradients and uses error compensation or error feedback (Alistarh et al., 2017; Wu et al., 2018; Karimireddy et al., 2019), but it targets transient per-step messages rather than persistent optimizer state. SPARQ instead compresses the EMA consumed by a nonlinear LMO update, for which state reconstruction error need not directly control update error.

Structure-aware and LMO-based optimization. Beyond diagonal adaptivity, Shampoo exploits tensor structure through dimension-wise matrix preconditioners (Gupta et al., 2018). LMO methods originate in conditional-gradient optimization, where each step minimizes a linearization over a norm-constrained domain (Jaggi, 2013). Muon adapts this geometry to neural-network training by orthogonalizing the momentum of hidden matrices, typically through Newton–Schulz iterations, while using AdamW for the remaining parameters (Jordan et al., 2024; Liu et al., 2025; Higham, 1986). Scion generalizes the idea through norm-constrained LMOs and layerwise update radii (Pethick et al., 2025). Gluon further connects practical layerwise LMO routing with convergence analysis and scale-transfer behavior (Riabinin et al., 2025). Building on the state-compression design of MuonGRASP, SPARQ combines that representation with Scion's routing framework and applies the spectral path to every matrix-shaped parameter, while retaining the RMS path for one-dimensional tensors.

Low-bit LMO optimizers. Recent work has begun to quantize Muon's momentum state. Blockwise schemes demonstrate that Muon can tolerate 8-bit storage (Gupta et al., 2025), while MuonGRASP introduced a subspace-preserving low-rank-plus-residual representation, warm-started subspace iteration, and grid quantization for 4-bit Muon states (Wu et al., 2026). MuonQ similarly targets directional fidelity under low-bit quantization (Su et al., 2026). SPARQ does not claim these storage operations as new; it transfers the subspace-preserving representation to the Scion family and studies a different routing regime in which every matrix-shaped parameter uses the spectral LMO, while genuinely one-dimensional tensors retain the RMS path. Our analysis consequently studies both state accumulation and the sensitivity of the resulting LMO update.

B Main Structure of SPARQ

Figure 4 illustrates the state-compression pipeline of SPARQ. For a matrix-shaped parameter, the optimizer first forms the EMA buffer B_k using the reconstructed state from the previous iteration. A warm-started subspace iteration then obtains the low-rank factors $P_k \in \mathbb{R}^{m\times r}$ and $R_k \in \mathbb{R}^{n\times r}$, where $r \ll \min(m, n)$. The buffer is decomposed as $B_k = P_k R_k^\top + B_k^{\text{res}}$. The factors are stored in INT8 to preserve the dominant subspace, while the residual is stored in INT4 to reduce memory usage. At the next iteration, these components are dequantized and reconstructed before forming the new EMA buffer. The matrix-shaped parameter then uses the scaled spectral LMO, whereas one-dimensional parameters bypass subspace compression, retain an FP32 state, and use the RMS LMO. This design aligns the state representation with the geometry of the Scion update while reducing persistent optimizer-state memory.

MuonGRASP (Wu et al., 2026) is the closest prior work to SPARQ. It introduced subspace-preserving low-bit compression for Muon states by decomposing the momentum buffer into a retained low-rank component and a residual, updating the retained subspace with a warm-started iteration, and assigning INT8 and INT4 precision to the factors and residual, respectively. SPARQ adopts this state-compression principle but transfers it from the Muon setting to the Scion family.

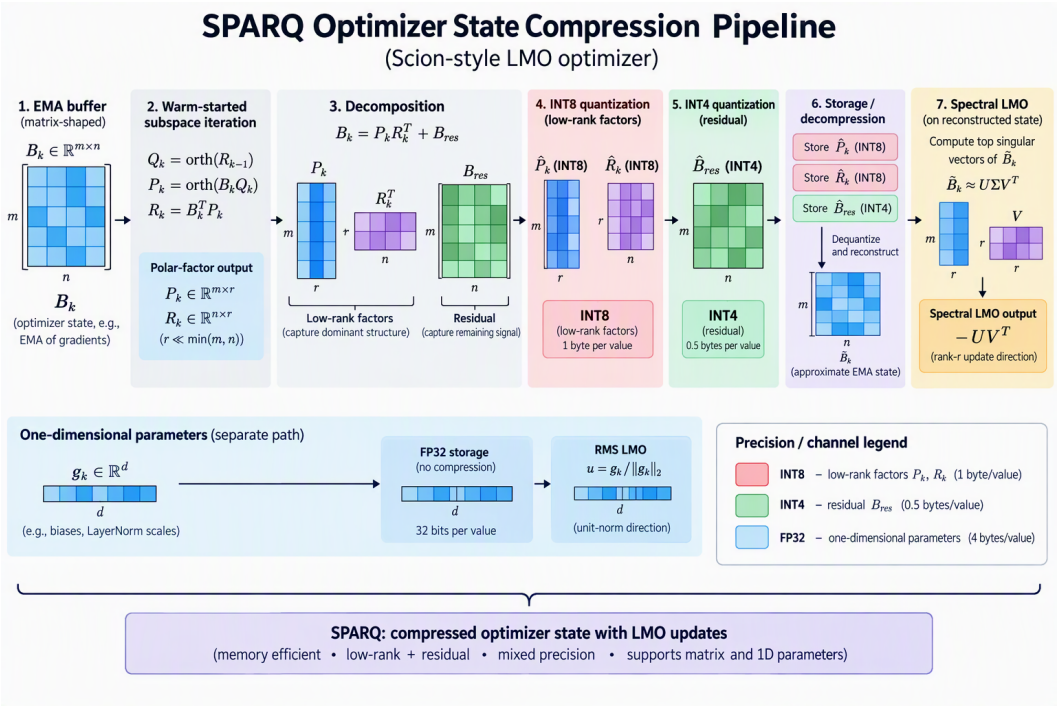


Figure 4: SPARQ optimizer-state compression pipeline. Matrix-shaped EMA states are decomposed into INT8 low-rank factors and an INT4 residual, while one-dimensional parameters retain FP32 states and use the RMS LMO.

Table 2: Block-wise comparison between MuonGRASP and SPARQ. Gray rows highlight the main differences in optimizer geometry and parameter routing.

Block / operation	MuonGRASP	SPARQ
Matrix EMA	$M_t = \mu M_{t-1} + g_t$	$B_k = \alpha_k h_k + (1-\alpha_k)\widetilde{B}_{k-1}$
Matrix update	$O_t = \mathrm{NS}(M_t)$; $W_t = W_{t-1} - \eta_t(O_t + \lambda W_{t-1})$	$O_k = \rho_b \,\mathrm{lmo}_{\mathrm{spectral}}(B_k)$; $X_{k+1} = X_k + \gamma_k O_k$ or $X_{k+1} = (1-\gamma_k)X_k + \gamma_k O_k$
Subspace iteration	$Q_t = \mathrm{colnorm}(R_{t-1})$; $P_t = \mathrm{orth}(M_t Q_t)$; $R_t = M_t^\top P_t$	$Q_k = \mathrm{orth}(\widetilde{R}_{k-1})$; $P_k = \mathrm{orth}(B_k Q_k)$; $R_k = B_k^\top P_k$
Low-rank split	$M_t = P_t R_t^\top + M_t^{\mathrm{res}}$; $M_t^{\mathrm{res}} = M_t - P_t R_t^\top$	$B_k = P_k R_k^\top + B_k^{\mathrm{res}}$; $B_k^{\mathrm{res}} = B_k - P_k R_k^\top$
State reconstruction	$\widetilde{M}_t = \widetilde{P}_t \widetilde{R}_t^\top + \widetilde{M}_t^{\mathrm{res}}$	$\widetilde{B}_k = \widetilde{P}_k \widetilde{R}_k^\top + \widetilde{B}_k^{\mathrm{res}}$
State precision	$P_t, R_t \in \mathrm{INT8}$; $M_t^{\mathrm{res}} \in \mathrm{INT4}$	$P_k, R_k \in \mathrm{INT8}$; $B_k^{\mathrm{res}} \in \mathrm{INT4}$
Matrix routing	$W \in \mathcal{P}_{\mathrm{hidden}} \Rightarrow \mathrm{Muon}(W)$; $W \notin \mathcal{P}_{\mathrm{hidden}} \Rightarrow \mathrm{AdamW}(W)$	$\mathrm{order}(W) \geq 2 \Rightarrow \rho_b \,\mathrm{lmo}_{\mathrm{spectral}}(B_k)$
One-dimensional routing	$W \in \mathcal{P}_{\mathrm{1D}} \Rightarrow \mathrm{AdamW}(W)$	$\mathrm{order}(W) = 1 \Rightarrow \rho_b \,\mathrm{lmo}_{\mathrm{RMS}}(B_k)$; $S_k = B_k \in \mathrm{FP32}$
Optimizer geometry	$\mathrm{Muon} \xrightarrow{\mathrm{NS}} \mathrm{orthogonalized\ update}$	$\mathrm{Scion} \xrightarrow{\mathrm{LMO}}$ $\{\mathrm{spectral\ matrices}, \mathrm{RMS\ vectors}\}$

The main difference is the routing regime: MuonGRASP applies Muon primarily to hidden-layer matrices and uses AdamW for embeddings, the language-model head, and one-dimensional parameters, whereas SPARQ applies the Scion spectral LMO to every matrix-shaped parameter and retains the RMS LMO only for one-dimensional parameters. Accordingly, SPARQ does not claim the underlying low-rank, warm-started, mixed-precision state representation as a new compression operation. Its contribution lies in combining this representation with Scion's norm-constrained LMO framework, uniform spectral routing across matrix-shaped parameters, and a Scion-specific analysis of EMA storage drift, convergence under bounded reconstruction error, and spectral-LMO sensitivity. The experiments further evaluate this transferred design across Scion-routed matrix parameters and compare it with full-precision Scion and uniformly quantized Scion states. Table 2 summarizes the block-wise differences between the two methods.

C Complete Theoretical Analysis

This appendix proves the formal results stated in Section 4. All notation, assumptions, and result statements are given in the main text.

C.1 Theoretical Preliminaries

This sub-section collects the mathematical definitions used in Section 4 and Appendix C.

Definition C.1 (Norm geometry). *Let $\mathcal{X}$ be a finite-dimensional real vector space with norm $\|\cdot\|$. Its dual norm is*

$$\|z\|_* := \max_{\|x\|\le 1} \langle z, x\rangle, \tag{30}$$

and Hölder's inequality gives

$$|\langle z, x\rangle| \le \|z\|_* \|x\|. \tag{31}$$

For the radius-ρ ball D, norm equivalence in finite dimension implies

$$\|x\|_2 \le \rho_2 \ (x \in D), \qquad \|x - y\|_2 \le D_2 \ (x, y \in D), \qquad \|z\|_* \le \zeta \|z\|_2. \tag{32}$$

For SPARQ, $\|\cdot\|$ is the product norm induced by its blockwise routing.

Definition C.2 (Smoothness). *If ∇f is L-Lipschitz from $\|\cdot\|$ to $\|\cdot\|_*$, then*

$$f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y - x\|^2. \tag{33}$$

Under Euclidean L_2-smoothness,

$$f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \frac{L_2}{2}\|y - x\|_2^2. \tag{34}$$

Definition C.3 (Linear minimization oracle). *For $D = \{x : \|x\| \le \rho\}$, the linear minimization oracle (LMO) is*

$$\mathrm{lmo}(z) \in \arg\min_{x \in D} \langle z, x\rangle. \tag{35}$$

It satisfies

$$\min_{x \in D} \langle z, x\rangle = -\rho \|z\|_*, \tag{36}$$

and its solution set is positively scale invariant:

$$\arg\min_{x \in D} \langle az, x\rangle = \arg\min_{x \in D} \langle z, x\rangle, \qquad a > 0. \tag{37}$$

Definition C.4 (Stochastic-gradient noise). *Let $\mathcal{F}_k$ contain the information available before sampling g_k, and define*

$$\delta_k := g_k - \nabla f(x^k). \tag{38}$$

Under Assumption 4.1(ii), $\{\delta_k\}$ is a martingale-difference sequence and, for $i < j$,

$$\mathbb{E}[\langle \delta_i, \delta_j\rangle] = 0. \tag{39}$$

Definition C.5 (Random output iterate)**.** *Let $\bar{x}^n$ be sampled uniformly from $\{x^1, \dots, x^n\}$. Then, for every nonnegative measurable function h,*

$$\mathbb{E}[h(\bar{x}^n)] = \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}[h(x^k)]. \tag{40}$$

Definition C.6 (Storage and EMA errors)**.** *Let B_k be the pre-storage quantized-trajectory buffer and $G_k^\star$ its coupled exact-storage reference from Eqs. 13 and 14. Define the storage drift by*

$$\epsilon_k := B_k - G_k^\star, \tag{41}$$

and the exact-storage EMA error by

$$\lambda_k := G_k^\star - \nabla f(x^k). \tag{42}$$

Thus, $B_k - \nabla f(x^k) = \epsilon_k + \lambda_k$.

Definition C.7 (Stationarity measures)**.** *For unconstrained optimization, we use the dual gradient norm $\|\nabla f(x)\|_*$. For optimization over D, we use the Frank–Wolfe gap*

$$\mathcal{G}_D(x) := \max_{u\in D}\langle \nabla f(x), x - u\rangle. \tag{43}$$

The constrained update is

$$x^{k+1} = (1-\gamma)x^k + \gamma\,\mathrm{lmo}(B_k), \tag{44}$$

which preserves $x^{k+1} \in D$ when $x^k \in D$ and $\gamma \in [0, 1]$.

Definition C.8 (Unitary polar factor)**.** *For a full-column-rank matrix with thin singular value decomposition (SVD)*

$$A = U\Sigma V^\top, \tag{45}$$

its unitary polar factor is

$$\Phi(A) := UV^\top. \tag{46}$$

The spectral LMO is a scaled negative polar factor. For wide matrices, the analysis applies the same definition to $A^\sharp = A^\top$.

Definition C.9 (Decaying term and error floor)**.** *A term is decaying if it converges to zero as the training horizon increases. An error floor is a nonvanishing term caused by persistent stochastic or storage error. The SPARQ bounds recover their exact-storage counterparts when the reconstruction-error bound satisfies $q = 0$.*

C.2 Proof of Lemma 4.3

Proof. Because the ambient space is finite-dimensional, the norm ball D is closed and bounded, and hence compact. The map $x \mapsto \langle z, x\rangle$ is continuous, so its minimum over D is attained. Therefore $\mathrm{lmo}(z) \in D$, and

$$\|\mathrm{lmo}(z)\| \le \rho.$$

Moreover, by the definition $\rho_2 := \sup_{x\in D}\|x\|_2$, membership in D also implies

$$\|\mathrm{lmo}(z)\|_2 \le \rho_2.$$

It remains to evaluate the minimum. Writing every $x \in D$ as $x = \rho u$ with $\|u\| \le 1$, and using the symmetry of a norm ball, gives

$$\begin{aligned}\min_{x\in D}\langle z, x\rangle &= \rho \min_{\|u\|\le 1}\langle z, u\rangle \\ &= -\rho \max_{\|u\|\le 1}\langle z, u\rangle \\ &= -\rho\|z\|_*.\end{aligned}$$

The last equality is the definition of the dual norm. Since $\mathrm{lmo}(z)$ attains this minimum,

$$\langle z, \mathrm{lmo}(z)\rangle = -\rho\|z\|_*.$$

This identity also holds at $z = 0$, for which both sides vanish. □

C.3 Proof of Proposition 4.4

Proof. We verify that the full-precision spectral-routing reference is an instance of the norm-constrained LMO framework analyzed by Pethick et al. (2025), and then apply their four convergence theorems. Write the joint parameter space as the finite product $\mathcal{X} = \prod_{b=1}^{B} \mathcal{X}_b$. For every matrix-shaped block, let ν_b be the scaled spectral norm assigned by the reference routing; for every one-dimensional block, let ν_b be its scaled RMS norm. The spectral norm and the RMS norm are norms, and multiplication by a positive scaling factor preserves the norm axioms. Hence every ν_b is a norm on $\mathcal{X}_b$. Let $\rho_b > 0$ be the radius assigned to block b. The blockwise feasible set is

$$D_{\mathrm{SP}} := \prod_{b=1}^{B} \{x_b \in \mathcal{X}_b : \nu_b(x_b) \le \rho_b\}.$$

For any fixed global radius $\rho > 0$, define

$$\|x\|_{\mathrm{SP}} := \rho \max_{1 \le b \le B} \frac{\nu_b(x_b)}{\rho_b}, \qquad x = (x_1, \dots, x_B).$$

The maximum of finitely many positively scaled block norms is a norm on the product space. Moreover,

$$D_{\mathrm{SP}} = \{x \in \mathcal{X} : \|x\|_{\mathrm{SP}} \le \rho\}.$$

Thus the routing used by the reference defines a valid norm ball of exactly the form required in the Scion analysis. It remains to verify that the LMO of this product ball agrees with the blockwise routing. For a joint direction $z = (z_1, \dots, z_B)$, separability of both the inner product and D_{SP} gives

$$\begin{aligned} \arg\min_{x \in D_{\mathrm{SP}}} \langle z, x \rangle &= \arg\min_{\substack{x=(x_1,\dots,x_B) \\ \nu_b(x_b) \le \rho_b \ \forall b}} \sum_{b=1}^{B} \langle z_b, x_b \rangle \\ &= \prod_{b=1}^{B} \arg\min_{x_b :\, \nu_b(x_b) \le \rho_b} \langle z_b, x_b \rangle. \end{aligned}$$

Each matrix block on the right-hand side is precisely the scaled spectral LMO, and each one-dimensional block is precisely the scaled RMS LMO. Therefore the joint LMO associated with $\|\cdot\|_{\mathrm{SP}}$ is exactly the update oracle used by the full-precision spectral-routing reference. In particular, Lemma 4.3 yields

$$\langle z, \mathrm{lmo}_{\mathrm{SP}}(z) \rangle = -\rho \|z\|_{\mathrm{SP},*}, \qquad \mathrm{lmo}_{\mathrm{SP}}(z) \in D_{\mathrm{SP}}.$$

The two reference dynamics now coincide with the algorithms covered by the cited results. Without the convex-combination term, the update is unconstrained SCG (uSCG), so Theorems 5.4 and 5.5 of Pethick et al. (2025) apply for constant and vanishing momentum, respectively. With the convex-combination update and a feasible initialization, it is constrained SCG, so Theorems 5.6 and 5.7 apply for constant and vanishing momentum, respectively. This application is valid by the premise of Proposition 4.4: for each cited theorem, its smoothness, stochastic-oracle, initialization, stepsize, momentum, and feasibility conditions are assumed to hold with respect to $\|\cdot\|_{\mathrm{SP}}$ and D_{SP}. Finally, changing the routing changes the norm and therefore may change the associated smoothness constant, dual norm, Euclidean radius or diameter, and other norm-equivalence constants. These quantities are fixed problem constants once $\|\cdot\|_{\mathrm{SP}}$ is chosen; they introduce no new dependence on the iteration horizon n. Substituting the new constants into Theorems 5.4–5.7 therefore leaves every power of n in their bounds unchanged. The corresponding convergence-rate orders are consequently preserved. This argument does not claim that the numerical constants, or their dependence on model width, remain unchanged. □

C.4 Proof of Lemma 4.5

Proof. By definition, $\epsilon_k = B_k - G_k^\star$. Substituting Eqs. 13 and 14 yields

$$\begin{aligned} \epsilon_k &= \alpha_k g_k + (1-\alpha_k)(B_{k-1} + e_{k-1}) - \left[\alpha_k g_k + (1-\alpha_k) G_{k-1}^\star\right] \\ &= (1-\alpha_k)\left(B_{k-1} - G_{k-1}^\star + e_{k-1}\right) \\ &= (1-\alpha_k)(\epsilon_{k-1} + e_{k-1}). \end{aligned}$$

At $k = 1$, the convention $\alpha_1 = 1$ gives $B_1 = g_1 = G_1^\star$, and therefore $\epsilon_1 = 0$. □

C.5 Proof of Lemma 4.6

Proof. Assumption 4.2 is conditional on $\mathcal{Q}_k$. Taking total expectation and applying the tower property gives

$$\begin{aligned}\mathbb{E}\|e_k\|_2^2 &= \mathbb{E}\big[\mathbb{E}\big[\|e_k\|_2^2 \mid \mathcal{Q}_k\big]\big] \\ &\le q^2.\end{aligned}$$

Thus, with $\|X\|_{L_2} := (\mathbb{E}\|X\|_2^2)^{1/2}$, $\|e_k\|_{L_2} \le q$ for every k. Set $\beta := 1-\alpha$. Iterating Lemma 4.5 from $\epsilon_1 = 0$ gives, for $k \ge 2$,

$$\epsilon_k = \sum_{t=1}^{k-1} \beta^{k-t} e_t.$$

Minkowski's inequality in L_2 therefore yields

$$\begin{aligned}\|\epsilon_k\|_{L_2} &\le \sum_{t=1}^{k-1} \beta^{k-t}\|e_t\|_{L_2} \\ &\le q\sum_{r=1}^{k-1}\beta^r \\ &= \frac{\beta q}{1-\beta}(1-\beta^{k-1}) \\ &= \frac{(1-\alpha)q}{\alpha}\big(1-(1-\alpha)^{k-1}\big).\end{aligned}$$

Finally, $0 \le (1-\alpha)^{k-1} \le 1$, so

$$\|\epsilon_k\|_{L_2} \le \frac{(1-\alpha)q}{\alpha}.$$

□

C.6 Proof of Lemma 4.7

Proof. Let $u_k := (\mathbb{E}\|\epsilon_k\|_2^2)^{1/2}$. As above, Assumption 4.2 and the tower property give $\|e_k\|_{L_2} \le q$. We prove

$$u_k \le 2q(\sqrt{k}-1)$$

by induction on k. At $k=1$, Lemma 4.5 gives $u_1 = 0 = 2q(\sqrt{1}-1)$. Now fix $k \ge 2$ and suppose $u_{k-1} \le 2q(\sqrt{k-1}-1)$. Since $\alpha_k = k^{-1/2}$, Lemma 4.5 and Minkowski's inequality give

$$\begin{aligned}u_k &\le \Big(1-\frac{1}{\sqrt{k}}\Big)\big(u_{k-1}+\|e_{k-1}\|_{L_2}\big) \\ &\le \Big(1-\frac{1}{\sqrt{k}}\Big)\big(2q(\sqrt{k-1}-1)+q\big) \\ &= \Big(1-\frac{1}{\sqrt{k}}\Big)\big(2q\sqrt{k-1}-q\big) \\ &\le \Big(1-\frac{1}{\sqrt{k}}\Big)\big(2q\sqrt{k}-q\big) \\ &= 2q\sqrt{k}-3q+\frac{q}{\sqrt{k}}.\end{aligned}$$

Because $q/\sqrt{k} \le q$, the final expression is at most $2q\sqrt{k}-2q = 2q(\sqrt{k}-1)$, completing the induction. □

C.7 Proof of Theorem 4.8

Proof. Fix $k \in \{1,\dots,n\}$ and abbreviate $\nabla_k := \nabla f(x^k)$ and $s_k := \mathrm{lmo}(B_k)$. Assumption 4.1(i) permits the descent lemma between consecutive iterates:

$$f(x^{k+1}) \le f(x^k) + \langle \nabla_k, x^{k+1}-x^k\rangle + \frac{L}{2}\|x^{k+1}-x^k\|^2.$$

By Eq. 20, $x^{k+1} - x^k = \gamma s_k$. Lemma 4.3 gives $\|s_k\| \le \rho$, and hence

$$\|x^{k+1} - x^k\|^2 = \gamma^2 \|s_k\|^2 \le \gamma^2 \rho^2.$$

Substitution into the descent lemma gives

$$f(x^{k+1}) \le f(x^k) + \gamma \langle \nabla_k, s_k \rangle + \frac{L\gamma^2\rho^2}{2}.$$

We decompose the remaining inner product as

$$\langle \nabla_k, s_k \rangle = \langle \nabla_k - B_k, s_k \rangle + \langle B_k, s_k \rangle.$$

By Lemma 4.3, $\langle B_k, s_k \rangle = -\rho \|B_k\|_*$. Define $\lambda_k := G_k^\star - \nabla_k$ and recall $\epsilon_k = B_k - G_k^\star$. Then

$$B_k - \nabla_k = \lambda_k + \epsilon_k.$$

The reverse triangle inequality for the dual norm implies

$$\|B_k\|_* = \|\nabla_k + (\lambda_k + \epsilon_k)\|_* \ge \|\nabla_k\|_* - \|\lambda_k + \epsilon_k\|_* \ge \|\nabla_k\|_* - \|\lambda_k\|_* - \|\epsilon_k\|_*.$$

Multiplying by $-\rho$ gives

$$-\rho\|B_k\|_* \le -\rho\|\nabla_k\|_* + \rho\|\lambda_k\|_* + \rho\|\epsilon_k\|_* \le -\rho\|\nabla_k\|_* + \zeta\rho\big(\|\lambda_k\|_2 + \|\epsilon_k\|_2\big),$$

where the final line uses $\|v\|_* \le \zeta\|v\|_2$. For the first inner-product term, Euclidean Cauchy–Schwarz and Lemma 4.3 give

$$\langle \nabla_k - B_k, s_k \rangle \le \|\nabla_k - B_k\|_2 \|s_k\|_2 = \|\lambda_k + \epsilon_k\|_2 \|s_k\|_2 \le \rho_2\big(\|\lambda_k\|_2 + \|\epsilon_k\|_2\big).$$

Combining these estimates with the descent inequality and rearranging gives

$$\gamma\rho\,\|\nabla_k\|_* \;\le\; f(x^k) - f(x^{k+1}) + \gamma\,(\rho_2 + \zeta\rho)\big(\|\lambda_k\|_2 + \|\epsilon_k\|_2\big) + \frac{L\gamma^2\rho^2}{2}. \tag{47}$$

Let $\beta := 1 - \alpha$ and $\delta_j := g_j - \nabla f(x^j)$. Since $\alpha_1 = 1$, unrolling Eq. 14 gives

$$G_k^\star = \sum_{j=1}^{k} w_{k,j} g_j, \qquad w_{k,1} = \beta^{k-1}, \qquad w_{k,j} = \alpha\beta^{k-j} \quad (j \ge 2).$$

The weights are nonnegative and sum to one because

$$\sum_{j=1}^{k} w_{k,j} = \beta^{k-1} + \alpha \sum_{r=0}^{k-2} \beta^r = \beta^{k-1} + \alpha\frac{1-\beta^{k-1}}{1-\beta} = 1.$$

Adding and subtracting the weighted current gradient gives

$$\lambda_k = G_k^\star - \nabla f(x^k) = \sum_{j=1}^{k} w_{k,j}\delta_j + \sum_{j=1}^{k} w_{k,j}\big(\nabla f(x^j) - \nabla f(x^k)\big).$$

For $i < j$, δ_i is measurable with respect to the information available before g_j is drawn. The martingale-difference property in Assumption 4.1(ii) therefore gives

$$\mathbb{E}\langle \delta_i, \delta_j \rangle = \mathbb{E}[\langle \delta_i, \mathbb{E}[\delta_j \mid \mathcal{F}_j] \rangle] = 0.$$

Thus all cross terms vanish. The conditional variance bound and the tower property imply

$$\mathbb{E}\left\| \sum_{j=1}^{k} w_{k,j}\delta_j \right\|_2^2 = \sum_{j=1}^{k} w_{k,j}^2 \mathbb{E}\|\delta_j\|_2^2 \le \sigma^2 \sum_{j=1}^{k} w_{k,j}^2.$$

The squared weights satisfy

$$\sum_{j=1}^{k} w_{k,j}^2 = \beta^{2(k-1)} + \alpha^2 \sum_{r=0}^{k-2} \beta^{2r} \le \beta^{2(k-1)} + \frac{\alpha^2}{1-\beta^2} = \beta^{2(k-1)} + \frac{\alpha}{2-\alpha}.$$

Taking square roots and using $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$ yields

$$\left\| \sum_{j=1}^{k} w_{k,j} \delta_j \right\|_{L_2} \le \sigma \left(\beta^{k-1} + \sqrt{\frac{\alpha}{2-\alpha}} \right).$$

We next control the gradient-drift term. Equation 20 and Lemma 4.3 imply

$$\|x^{t+1} - x^t\|_2 = \gamma \|\mathrm{lmo}(B_t)\|_2 \le \gamma \rho_2.$$

Therefore, for $j \le k$,

$$\|x^k - x^j\|_2 \le \sum_{t=j}^{k-1} \|x^{t+1} - x^t\|_2 \le \gamma \rho_2 (k-j).$$

By L_2-smoothness,

$$\|\nabla f(x^j) - \nabla f(x^k)\|_2 \le L_2 \gamma \rho_2 (k-j).$$

It follows that

$$\left\| \sum_{j=1}^{k} w_{k,j} \big(\nabla f(x^j) - \nabla f(x^k) \big) \right\|_2 \le L_2 \gamma \rho_2 a_k,$$

where $a_k := \sum_{j=1}^{k} w_{k,j}(k-j)$ is the mean age of the EMA weights. Directly from their recursion, $a_1 = 0$ and

$$a_k = \beta(a_{k-1} + 1).$$

Solving this recursion gives

$$a_k = \sum_{r=1}^{k-1} \beta^r = \frac{\beta(1-\beta^{k-1})}{1-\beta} \le \frac{1-\alpha}{\alpha}.$$

Minkowski's inequality now gives the pointwise estimate

$$\|\lambda_k\|_{L_2} \le \frac{(1-\alpha) L_2 \gamma \rho_2}{\alpha} + \sigma \left(\beta^{k-1} + \sqrt{\frac{\alpha}{2-\alpha}} \right).$$

Averaging this inequality and using $\sum_{k=1}^{n} \beta^{k-1} \le (1-\beta)^{-1} = \alpha^{-1}$ yields

$$\frac{1}{n} \sum_{k=1}^{n} \sqrt{\mathbb{E}\big[\|\lambda_k\|_2^2\big]} \;\le\; \frac{(1-\alpha) L_2 \rho_2 \gamma}{\alpha} + \left(\sqrt{\frac{\alpha}{2-\alpha}} + \frac{1}{\alpha n} \right) \sigma. \tag{48}$$

For every k, Cauchy–Schwarz inequality gives

$$\mathbb{E}\|\epsilon_k\|_2 \le \big(\mathbb{E}\|\epsilon_k\|_2^2 \big)^{1/2}.$$

Applying Lemma 4.6 term by term therefore yields

$$\frac{1}{n} \sum_{k=1}^{n} \mathbb{E}\big[\|\epsilon_k\|_2\big] \le \frac{1}{n} \sum_{k=1}^{n} \big(\mathbb{E}\|\epsilon_k\|_2^2 \big)^{1/2} \le \frac{(1-\alpha)\, q}{\alpha}. \tag{49}$$

Summing Eq. 47 over $k = 1, \dots, n$ gives

$$\begin{aligned} \gamma \rho \sum_{k=1}^{n} \|\nabla f(x^k)\|_* \le{}& \sum_{k=1}^{n} \big(f(x^k) - f(x^{k+1}) \big) \\ &+ \gamma(\rho_2 + \zeta\rho) \sum_{k=1}^{n} \big(\|\lambda_k\|_2 + \|\epsilon_k\|_2 \big) + \frac{n L \gamma^2 \rho^2}{2}. \end{aligned}$$

The objective terms telescope exactly:

$$\sum_{k=1}^{n} \big(f(x^k) - f(x^{k+1}) \big) = f(x^1) - f(x^{n+1}).$$

Taking expectations and using $f(x^{n+1}) \geq f^\star$ gives

$$\gamma\rho \sum_{k=1}^{n} \mathbb{E}\|\nabla f(x^k)\|_* \leq \mathbb{E}[f(x^1)] - f^\star + \gamma(\rho_2 + \zeta\rho) \sum_{k=1}^{n} (\mathbb{E}\|\lambda_k\|_2 + \mathbb{E}\|\epsilon_k\|_2) + \frac{nL\gamma^2\rho^2}{2}.$$

Divide by $\gamma\rho n$. Since $\bar{x}^n$ is sampled uniformly from $\{x^1, \ldots, x^n\}$,

$$\mathbb{E}\|\nabla f(\bar{x}^n)\|_* = \frac{1}{n} \sum_{k=1}^{n} \mathbb{E}\|\nabla f(x^k)\|_*.$$

Also, $\mathbb{E}\|\lambda_k\|_2 \leq (\mathbb{E}\|\lambda_k\|_2^2)^{1/2}$. Substituting Eqs. 48 and 49 therefore yields

$$\mathbb{E}\big[\|\nabla f(\bar{x}^n)\|_*\big] \leq \frac{\mathbb{E}[f(x^1)] - f^\star}{\rho\gamma n} + \frac{L\rho\gamma}{2} + \Big(\frac{\rho_2}{\rho} + \zeta\Big) \left(\frac{(1-\alpha)L_2\rho_2\gamma}{\alpha} + \left(\sqrt{\frac{\alpha}{2-\alpha}} + \frac{1}{\alpha n} \right) \sigma + \frac{(1-\alpha)q}{\alpha} \right).$$

Finally, substituting $\gamma = 1/\sqrt{n}$ into each occurrence of γ gives Eq. 21. □

C.8 Proof of Lemma 4.9

Proof. Let $s_k := \mathrm{lmo}(B_k)$ and fix an arbitrary $u \in D$. The constrained update is $x^{k+1} = x^k + \gamma(s_k - x^k)$. Since $x^k, s_k \in D$ and D_2 is the Euclidean diameter of D, $\|s_k - x^k\|_2 \leq D_2$. The Euclidean descent lemma gives

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L_2}{2}\|x^{k+1} - x^k\|_2^2 \\ &\leq f(x^k) + \gamma\langle \nabla f(x^k), s_k - x^k \rangle + \frac{L_2\gamma^2 D_2^2}{2}. \end{aligned}$$

By the optimality of s_k for the linear minimization problem, $\langle B_k, s_k \rangle \leq \langle B_k, u \rangle$, or equivalently $\langle B_k, s_k - u \rangle \leq 0$. Therefore

$$\begin{aligned} \langle \nabla f(x^k), s_k - x^k \rangle &= \langle \nabla f(x^k), u - x^k \rangle + \langle \nabla f(x^k), s_k - u \rangle \\ &= \langle \nabla f(x^k), u - x^k \rangle + \langle B_k, s_k - u \rangle + \langle \nabla f(x^k) - B_k, s_k - u \rangle \\ &\leq \langle \nabla f(x^k), u - x^k \rangle + \langle \nabla f(x^k) - B_k, s_k - u \rangle. \end{aligned}$$

Because $\nabla f(x^k) - B_k = -(\lambda_k + \epsilon_k)$ and $\|s_k - u\|_2 \leq D_2$, Cauchy–Schwarz yields

$$\langle \nabla f(x^k) - B_k, s_k - u \rangle \leq \|\lambda_k + \epsilon_k\|_2 \|s_k - u\|_2 \leq D_2\big(\|\lambda_k\|_2 + \|\epsilon_k\|_2\big).$$

Substitution into the descent bound and rearrangement give, for every $u \in D$,

$$\gamma\langle \nabla f(x^k), x^k - u \rangle \leq f(x^k) - f(x^{k+1}) + \gamma D_2\big(\|\lambda_k\|_2 + \|\epsilon_k\|_2\big) + \frac{L_2\gamma^2 D_2^2}{2}.$$

The right-hand side is independent of u. Maximizing the left-hand side over $u \in D$ produces $\gamma\mathcal{G}_D(x^k)$ and proves Eq. 24. □

C.9 Proof of Theorem 4.10

Proof. We first verify feasibility. Since $x^1 \in D$, $\mathrm{lmo}(B_k) \in D$ by Lemma 4.3, and $\gamma = 1/\sqrt{n} \in (0, 1]$, the update

$$x^{k+1} = (1-\gamma)x^k + \gamma\mathrm{lmo}(B_k)$$

is a convex combination of two points in the convex set D. Induction therefore gives $x^k \in D$ for every k. The EMA weights and stochastic-noise calculation from Step 2 of the proof of Theorem 4.8 depend only on the buffer recursion, so they remain unchanged. The only modification is the bound on the movement of the iterates. Under the constrained update,

$$\|x^{k+1} - x^k\|_2 = \gamma\|\mathrm{lmo}(B_k) - x^k\|_2 \leq \gamma D_2. \tag{50}$$

Consequently, for every $j \leq k$,

$$\|x^k - x^j\|_2 \leq \sum_{t=j}^{k-1} \|x^{t+1} - x^t\|_2 \leq \gamma D_2(k-j),$$

and L_2-smoothness gives

$$\|\nabla f(x^j) - \nabla f(x^k)\|_2 \le L_2\gamma D_2(k-j).$$

Using the same mean-age bound $\sum_{j=1}^{k} w_{k,j}(k-j) \le (1-\alpha)/\alpha$ gives

$$\left\| \sum_{j=1}^{k} w_{k,j}\big(\nabla f(x^j) - \nabla f(x^k)\big) \right\|_2 \le \frac{(1-\alpha)L_2\gamma D_2}{\alpha}.$$

The stochastic term is unchanged and satisfies

$$\left\| \sum_{j=1}^{k} w_{k,j}\delta_j \right\|_{L_2} \le \sigma\left((1-\alpha)^{k-1} + \sqrt{\frac{\alpha}{2-\alpha}}\right).$$

Minkowski's inequality, followed by averaging over k and $\sum_{k=1}^{n}(1-\alpha)^{k-1} \le 1/\alpha$, therefore yields

$$\frac{1}{n}\sum_{k=1}^{n}\sqrt{\mathbb{E}[\|\lambda_k\|_2^2]} \le \frac{(1-\alpha)L_2D_2\gamma}{\alpha} + \left(\sqrt{\frac{\alpha}{2-\alpha}} + \frac{1}{\alpha n}\right)\sigma. \tag{51}$$

For the storage drift, Lemma 4.6 and $\mathbb{E}\|\epsilon_k\|_2 \le (\mathbb{E}\|\epsilon_k\|_2^2)^{1/2}$ give

$$\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}[\|\epsilon_k\|_2] \le \frac{(1-\alpha)q}{\alpha}. \tag{52}$$

Summing Eq. 24 over $k = 1, \dots, n$ gives

$$\gamma\sum_{k=1}^{n}\mathcal{G}_D(x^k) \le \sum_{k=1}^{n}\big(f(x^k) - f(x^{k+1})\big) + \gamma D_2\sum_{k=1}^{n}\big(\|\lambda_k\|_2 + \|\epsilon_k\|_2\big) + \frac{nL_2\gamma^2D_2^2}{2}.$$

The first sum telescopes to $f(x^1) - f(x^{n+1})$. Taking expectations, using $f(x^{n+1}) \ge f^\star$, and dividing by $n\gamma$ gives

$$\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}[\mathcal{G}_D(x^k)] \le \frac{\mathbb{E}[f(x^1)] - f^\star}{n\gamma} + \frac{L_2\gamma D_2^2}{2} + \frac{D_2}{n}\sum_{k=1}^{n}\left(\mathbb{E}[\|\lambda_k\|_2] + \mathbb{E}[\|\epsilon_k\|_2]\right).$$

Because $\bar{x}^n$ is drawn uniformly after the run,

$$\mathbb{E}[\mathcal{G}_D(\bar{x}^n)] = \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}[\mathcal{G}_D(x^k)].$$

Furthermore, $\mathbb{E}\|\lambda_k\|_2 \le (\mathbb{E}\|\lambda_k\|_2^2)^{1/2}$. Applying Eqs. 51 and 52 gives

$$\mathbb{E}[\mathcal{G}_D(\bar{x}^n)] \le \frac{\mathbb{E}[f(x^1)] - f^\star}{n\gamma} + \frac{L_2\gamma D_2^2}{2} + D_2\left[\frac{(1-\alpha)L_2D_2\gamma}{\alpha} + \left(\sqrt{\frac{\alpha}{2-\alpha}} + \frac{1}{\alpha n}\right)\sigma + \frac{(1-\alpha)q}{\alpha}\right].$$

Finally, substituting $\gamma = 1/\sqrt{n}$ proves Eq. 25. □

C.10 Proof of Lemma 4.11

Proof. Let

$$A = \Phi(A)H, \qquad \widetilde{A} = \Phi(\widetilde{A})\widetilde{H}$$

be the polar decompositions, where H and $\widetilde{H}$ are positive definite because both matrices have full column rank. The perturbation results of Li (1995) apply to any unitarily invariant norm; the Frobenius norm used here is unitarily invariant. If $m = p$, Theorem 1 of Li (1995), applied with $\sigma_{\min}(A)$ and $\sigma_{\min}(\widetilde{A})$ as the two smallest singular values, gives

$$\|\Phi(\widetilde{A}) - \Phi(A)\|_F \le \frac{2}{\sigma_{\min}(A) + \sigma_{\min}(\widetilde{A})}\|\widetilde{A} - A\|_F,$$

which is Eq. 27.

If $m > p$, Theorem 3 of Li (1995) gives

$$\|\Phi(\widetilde{A}) - \Phi(A)\|_F \le \left[\left(\frac{2}{\sigma_{\min}(A) + \sigma_{\min}(\widetilde{A})}\right)^2 + \left(\frac{1}{\max\{\sigma_{\min}(A), \sigma_{\min}(\widetilde{A})\}}\right)^2\right]^{1/2} \cdot \|\widetilde{A} - A\|_F,$$

which is Eq. 28. No stochastic argument is used in either inequality; both are pointwise matrix perturbation bounds. □

C.11 Proof of Proposition 4.12

Proof. By Eq. 26, invariance of the Frobenius norm under transposition, and $\Phi(A^\top) = \Phi(A)^\top$, set

$$A_k := (G_k^\star)^\sharp, \qquad \widetilde{A}_k := B_k^\sharp.$$

The scaled spectral LMO then satisfies

$$\left\|\mathrm{LMO}^{(\rho_\ell)}_{\mathrm{spectral}}(B_k) - \mathrm{LMO}^{(\rho_\ell)}_{\mathrm{spectral}}(G_k^\star)\right\|_F = \rho_\ell c_\ell \left\|\Phi(\widetilde{A}_k) - \Phi(A_k)\right\|_F.$$

Moreover, because $B_k - G_k^\star = \epsilon_k$ and the operation $(\cdot)^\sharp$ is either the identity or transposition,

$$\|\widetilde{A}_k - A_k\|_F = \|B_k^\sharp - (G_k^\star)^\sharp\|_F = \|\epsilon_k^\sharp\|_F = \|\epsilon_k\|_F.$$

Suppose first that $m > p$. Applying Eq. 28 to $(A_k, \widetilde{A}_k)$ gives

$$\begin{aligned}&\left\|\mathrm{LMO}^{(\rho_\ell)}_{\mathrm{spectral}}(B_k) - \mathrm{LMO}^{(\rho_\ell)}_{\mathrm{spectral}}(G_k^\star)\right\|_F \\ &\le \rho_\ell c_\ell \left[\left(\frac{2}{\sigma_{\min}(A_k) + \sigma_{\min}(\widetilde{A}_k)}\right)^2 + \left(\frac{1}{\max\{\sigma_{\min}(A_k), \sigma_{\min}(\widetilde{A}_k)\}}\right)^2\right]^{1/2} \|\epsilon_k\|_F.\end{aligned}$$

Replacing A_k and $\widetilde{A}_k$ by their definitions is exactly Eq. 29. If $m = p$, Eq. 27 instead gives

$$\left\|\mathrm{LMO}^{(\rho_\ell)}_{\mathrm{spectral}}(B_k) - \mathrm{LMO}^{(\rho_\ell)}_{\mathrm{spectral}}(G_k^\star)\right\|_F \le \rho_\ell c_\ell \frac{2}{\sigma_{\min}((G_k^\star)^\sharp) + \sigma_{\min}(B_k^\sharp)} \|\epsilon_k\|_F,$$

which is the stated square-matrix coefficient. □

D More Experiments

This section provides additional experiments supporting the empirical findings in the main text. Unless otherwise noted, controlled ablations and robustness studies use LLaMA 124M, which enables broader sweeps at lower computational cost. The main-scale experiments use LLaMA 124M, GPT-2 350M, and LLaMA 3B. Larger-scale results on GPT-2 350M and LLaMA models, including the 3B configuration, are reported for the main comparisons and memory-scaling analysis rather than repeated for every ablation. Experimental settings are specified separately in each subsection.

D.1 Error Amplification Under Uniform Routing

Scion and 4-bit Scion use identical LMO routing, radii, update rules, and training hyperparameters. Their only difference is the representation of the EMA buffer: FP32 for Scion and uniform group-wise INT4 for 4-bit Scion. This controlled comparison isolates the effect of low-bit state storage. 4-bit Scion flattens each EMA buffer, partitions it into groups of s entries, and applies signed absmax quantization. For group b at step k,

$$\begin{aligned}a_{k,b} &= \max_j |[G_{k,b}]_j|, \\ [\widehat{G}_{k,b}]_j &= \mathrm{clip}\left(\mathrm{round}\left(\frac{7[G_{k,b}]_j}{a_{k,b}}\right), -7, 7\right), \\ [\widetilde{G}_{k,b}]_j &= \frac{a_{k,b}}{7}[\widehat{G}_{k,b}]_j.\end{aligned} \tag{53}$$

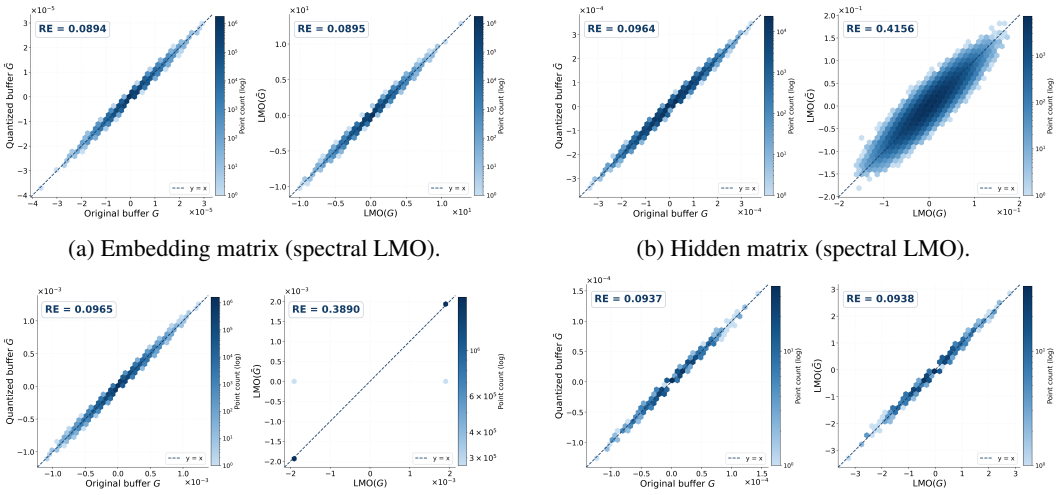
Groups with $a_{k,b} = 0$ are represented as zero. The trained 4-bit Scion baseline uses $s = 128$ unless stated otherwise. Its learning-rate schedule, weight decay, and all non-storage settings match the corresponding FP32 Scion run. For each sampled parameter tensor, we measure the relative error defined in Eq. 4 before and after applying its prescribed routing. Before the LMO, we compare $(X, \widetilde{X}) = (G_k, \widetilde{G}_k)$; after the LMO, we compare

$$(X, \widetilde{X}) = \big(\mathrm{lmo}(G_k), \mathrm{lmo}(\widetilde{G}_k)\big). \tag{54}$$

We repeat the diagnostic for three random seeds and group sizes $s \in \{32, 64, 128\}$. Each run uses a six-layer model with hidden dimension 512, feed-forward dimension 2048, vocabulary size 8192, 5 Newton–Schulz steps, and 300 training steps. We define the amplification ratio as

$$A := \frac{\mathrm{RE}_{\mathrm{after}}}{\mathrm{RE}_{\mathrm{before}}}. \tag{55}$$

Table 3 reports the mean and sample standard deviation over three seeds, while Figures 5– 7 show the corresponding entrywise density plots for seed 0. Across all nine seed–group-size configurations, quantization sensitivity depends strongly on the tensor and its routing: embedding and norm/bias parameters exhibit nearly unit amplification, whereas the hidden and output-head matrices, which both use the spectral LMO, amplify buffer errors by 4.07–$4.32\times$ and 3.68–$4.06\times$, respectively. Increasing the group size from 32 to 128 raises the mean post-LMO error from 0.4134 to 0.4791 for hidden matrices and from 0.3889 to 0.4330 for output heads, because each scale must cover a wider dynamic range. The slightly lower amplification ratios at larger group sizes therefore reflect faster growth of the pre-LMO error, rather than more accurate updates. These results motivate subspace-preserving state representations, while the diagnostic itself only measures update sensitivity and does not directly establish an effect on the training objective.



(a) Embedding matrix (spectral LMO).

(b) Hidden matrix (spectral LMO).

(c) Output head (spectral LMO).

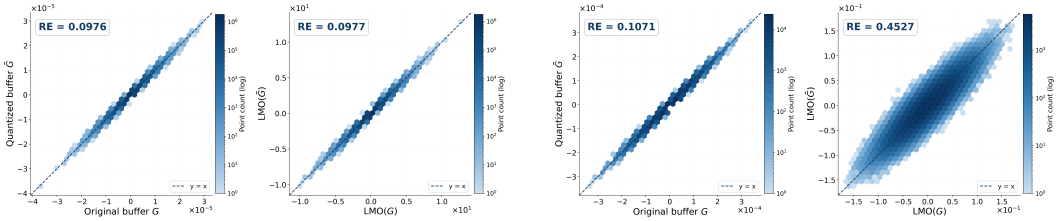
(d) Norm/bias (RMS LMO).

Figure 5: Entrywise effect of INT4 buffer quantization for group size $s = 32$ and seed 0. Each left panel compares the FP32 buffer with its INT4 reconstruction; each right panel compares the corresponding LMO outputs.

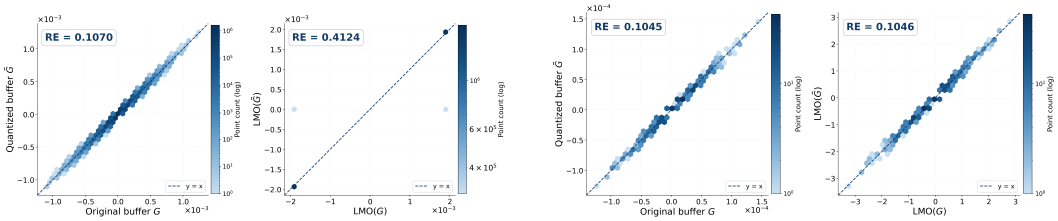
D.2 Quantization Design Ablations

We study three auxiliary design choices of 4-bit SPARQ on LLaMA 124M, using the full-precision Scion run as a reference. Unless otherwise stated, all quantized variants use the same nominal 4-bit configuration, group size, training schedule, and optimizer hyperparameters, with each comparison varying only the design choice under study. The results are summarized in Figure 8, which presents the rank-ratio trade-off, the comparison between 2D grid and group-wise quantization, and the effect of retaining the residual state.

We evaluate three design choices of 4-bit SPARQ. First, we vary the preserved-subspace rank ratio $\eta \in \{1/64, 1/32, 1/16, 1/8, 1/4\}$. Increasing η improves the downstream average from 61.2 at

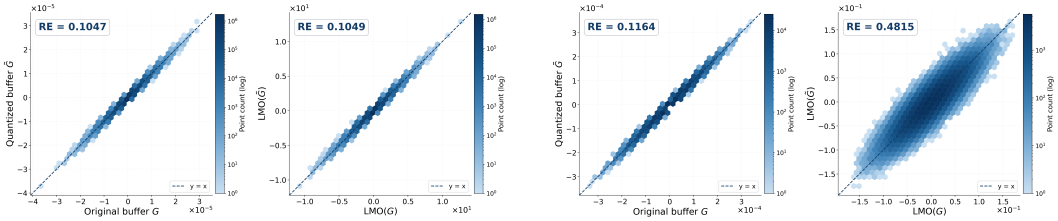


(a) Embedding matrix (spectral LMO). (b) Hidden matrix (spectral LMO).

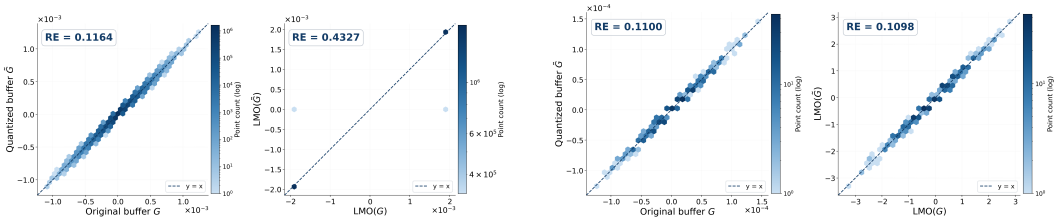


(c) Output head (spectral LMO). (d) Norm/bias (RMS LMO).

Figure 6: Entrywise effect of INT4 buffer quantization for group size $s = 64$ and seed 0. Panel definitions follow Figure 5.



(a) Embedding matrix (spectral LMO). (b) Hidden matrix (spectral LMO).



(c) Output head (spectral LMO). (d) Norm/bias (RMS LMO).

Figure 7: Entrywise effect of INT4 buffer quantization for group size $s = 128$ and seed 0. Panel definitions follow Figure 5.

Table 3: INT4 error before and after the prescribed LMO routing across group sizes. Values are mean ± sample standard deviation over three seeds.

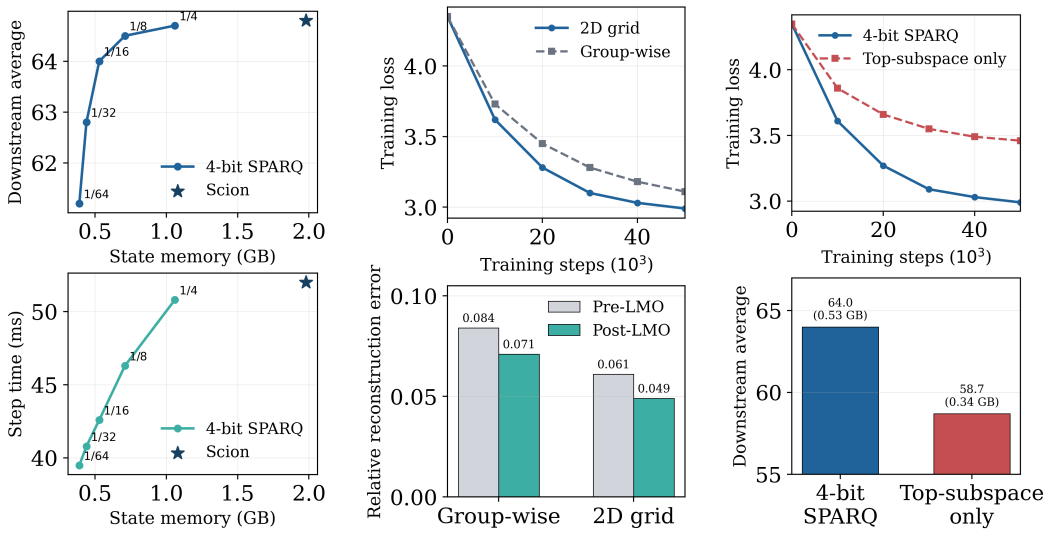
Group size	Parameter role	RE before LMO	RE after LMO	Amplification A
32	Embedding	0.08926 ± 0.00014	0.08940 ± 0.00013	1.0016 ± 0.0001
	Hidden	0.09630 ± 0.00024	0.41344 ± 0.00344	4.2932 ± 0.0428
	Output head	0.09647 ± 0.00004	0.38887 ± 0.00312	4.0309 ± 0.0314
	Norm / bias	0.09567 ± 0.00267	0.09574 ± 0.00266	1.0008 ± 0.0003
64	Embedding	0.09735 ± 0.00023	0.09753 ± 0.00022	1.0018 ± 0.0001
	Hidden	0.10683 ± 0.00020	0.44977 ± 0.00460	4.2102 ± 0.0401
	Output head	0.10690 ± 0.00006	0.41227 ± 0.00423	3.8567 ± 0.0386
	Norm / bias	0.10581 ± 0.00495	0.10586 ± 0.00497	1.0004 ± 0.0004
128	Embedding	0.10444 ± 0.00025	0.10465 ± 0.00024	1.0020 ± 0.0001
	Hidden	0.11610 ± 0.00023	0.47905 ± 0.00587	4.1260 ± 0.0464
	Output head	0.11643 ± 0.00006	0.43302 ± 0.00518	3.7191 ± 0.0425
	Norm / bias	0.11593 ± 0.00847	0.11584 ± 0.00868	0.9992 ± 0.0018

$\eta = 1/64$ to 64.0 at $\eta = 1/16$, but the gains become marginal thereafter, reaching only 64.5 and 64.7 at $\eta = 1/8$ and $\eta = 1/4$, respectively. In contrast, both state memory and step time continue to increase with η. The full-precision Scion reference achieves a downstream average of 64.8 using 1.98 GB of state memory and 52.0 ms per step, whereas 4-bit SPARQ with $\eta = 1/16$ achieves 64.0 using 0.53 GB and 42.6 ms per step. Thus, $\eta = 1/16$ retains 98.8% of the Scion downstream score while reducing state memory and step time by 73.2% and 18.1%, making it a favorable default trade-off. We next compare the 2D grid quantizer with conventional group-wise quantization under the same preserved rank, nominal bitwidth, group size, and optimization schedule. The 2D grid quantizer maintains a lower training loss and reaches a final loss of 2.99, compared with 3.11 for group-wise quantization. It also reduces the relative Frobenius reconstruction error from 0.061 to 0.049, whereas group-wise quantization yields errors of 0.084 and 0.071 before and after LMO routing, respectively. These results suggest that row–column scale normalization better preserves update structure. Finally, we compare 4-bit SPARQ, which retains the quantized residual, with a top-subspace-only variant that discards it. With $\eta = 1/16$ and the same learning-rate schedule, removing the residual causes the training curves to separate early, increases the final loss from 2.99 to 3.46, and decreases the downstream average from 64.0 to 58.7. Although this reduces state memory from 0.53 GB to 0.34 GB, the substantial loss in optimization and downstream performance shows that the residual state is important for retaining information outside the preserved top subspace.

D.3 Memory and Computational Efficiency

We estimate the optimizer-state memory of LLaMA 124M, GPT-2 350M, and LLaMA 3B. The compressed-state values include quantized states and the factors required for state reconstruction. The reported reductions concern optimizer-state memory only and do not represent total peak GPU memory.

For LLaMA 124M in Table 4, 4-bit Muon reduces the optimizer state from $\sim$ 530 MB to $\sim$ 66 MB, corresponding to an $8.0\times$ reduction and an 87.5% saving. 4-bit MuonGRASP reduces the state to $\sim$ 78 MB, corresponding to a $6.8\times$ reduction and an 85.3% saving. For Scion, uniform 4-bit quantization reduces the state from $\sim$ 465 MB to $\sim$ 58 MB, whereas 4-bit SPARQ reduces it to $\sim$ 67 MB. These correspond to reduction factors of $8.0\times$ and $6.9\times$, respectively. Thus, under the values reported in the table, 4-bit SPARQ uses approximately 9 MB more memory than uniform 4-bit Scion. For GPT-2 350M in Table 5, 4-bit Muon reduces the state from $\sim$ 1500 MB to $\sim$ 188 MB, corresponding to an $8.0\times$ reduction and an 87.5% saving. 4-bit MuonGRASP reduces the state to $\sim$ 291 MB, corresponding to a $5.2\times$ reduction and an 80.6% saving. For Scion, uniform 4-bit quantization reduces the state from $\sim$ 1320 MB to $\sim$ 165 MB, whereas 4-bit SPARQ reduces it to $\sim$ 252 MB. The corresponding reduction factors are $8.0\times$ and $5.2\times$, with state savings of 87.5% and 80.9%, respectively. Consequently, 4-bit SPARQ uses approximately 87 MB more memory than uniform 4-bit Scion. For LLaMA 3B in Table 6, 4-bit Muon reduces the state from $\sim$ 4.30



(a) Rank-ratio trade-off for 4-bit SPARQ.

(b) 2D grid versus group-wise quantization.

(c) Residual-state retention.

Figure 8: Quantization-design ablations for 4-bit SPARQ on LLaMA 124M. (a) Rank-ratio trade-offs in downstream score, state memory, and step time. (b) 2D grid versus group-wise quantization. (c) Residual-state retention.

GB to $\sim$ 0.54 GB, corresponding to an $8.0\times$ reduction and an 87.4% saving. 4-bit MuonGRASP reduces the state to $\sim$ 0.81 GB, corresponding to a $5.3\times$ reduction and an 81.2% saving. For Scion, uniform 4-bit quantization reduces the state from $\sim$ 3.80 GB to $\sim$ 0.47 GB, whereas 4-bit SPARQ reduces it to $\sim$ 0.67 GB. These correspond to reduction factors of $8.1\times$ and $5.7\times$, and state savings of 87.6% and 82.4%, respectively. Thus, 4-bit SPARQ uses approximately 0.20 GB more memory than uniform 4-bit Scion.

Table 4: Estimated optimizer-state memory on LLaMA 124M.

Optimizer	FP32 state	Compressed state	Reduction	State saving
AdamW	∼950 MB	∼950 MB	1.0×	0%
Muon	∼530 MB	∼530 MB	1.0×	0%
4-bit Muon	∼530 MB	∼66 MB	8.0×	87.5%
4-bit MuonGRASP	∼530 MB	∼78 MB	6.8×	85.3%
Scion	∼465 MB	∼465 MB	1.0×	0%
4-bit Scion	∼465 MB	∼58 MB	8.0×	87.5%
4-bit SPARQ	∼465 MB	∼67 MB	6.9×	85.6%

Table 5: Estimated optimizer-state memory on GPT-2 350M.

Optimizer	FP32 state	Compressed state	Reduction	State saving
AdamW	∼2700 MB	∼2700 MB	1.0×	0%
Muon	∼1500 MB	∼1500 MB	1.0×	0%
4-bit Muon	∼1500 MB	∼188 MB	8.0×	87.5%
4-bit MuonGRASP	∼1500 MB	∼291 MB	5.2×	80.6%
Scion	∼1320 MB	∼1320 MB	1.0×	0%
4-bit Scion	∼1320 MB	∼165 MB	8.0×	87.5%
4-bit SPARQ	∼1320 MB	∼252 MB	5.2×	80.9%

We next analyze the effects of the group size s and the preserved-subspace rank ratio η on the SPARQ compressed state. The group size s specifies the number of residual entries that share one

Table 6: Estimated optimizer-state memory on LLaMA 3B.

Optimizer	FP32 state	Compressed state	Reduction	State saving
AdamW	∼8.10 GB	∼8.10 GB	1.0×	0%
Muon	∼4.30 GB	∼4.30 GB	1.0×	0%
4-bit Muon	∼4.30 GB	∼0.54 GB	8.0×	87.4%
4-bit MuonGRASP	∼4.30 GB	∼0.81 GB	5.3×	81.2%
Scion	∼3.80 GB	∼3.80 GB	1.0×	0%
4-bit Scion	∼3.80 GB	∼0.47 GB	8.1×	87.6%
4-bit SPARQ	∼3.80 GB	∼0.67 GB	5.7×	82.4%

quantization scale. For a fixed rank ratio η, increasing s reduces the number of quantization scales and the associated metadata. The rank ratio η determines the retained subspace dimension through

$$\kappa = \max\{1, \lfloor \eta \min(d_{\text{out}}, d_{\text{in}}) \rfloor\}.$$

For a matrix-shaped buffer $B \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, the storage required by the low-rank factors grows with $\kappa(d_{\text{out}} + d_{\text{in}})$. For example, when $B \in \mathbb{R}^{1024 \times 4096}$ and $\eta = 1/16$,

$$\kappa = \left\lfloor \frac{1}{16} \min(1024, 4096) \right\rfloor = 64.$$

Thus, increasing η retains more subspace directions and increases the storage required by the low-rank factors.

Figure 9 summarizes the measured optimizer-state memory of 4-bit SPARQ across LLaMA 124M, GPT-2 350M, and LLaMA 3B. For LLaMA 124M, when $\eta = 1/32$, increasing s from 64 to 128 and then to 256 reduces the 4-bit SPARQ state from 22 MB to 20 MB and then to 19 MB; when $\eta = 1/16$, it decreases from 27 MB to 23 MB and then to 22 MB. At $s = 128$, increasing η from $1/32$ to $1/16$ and then to $1/8$ increases the 4-bit SPARQ state from 20 MB to 23 MB and then to 30 MB. The same trend holds for GPT-2 350M, where the corresponding values are 61, 59, and 57 MB for $\eta = 1/32$, 70, 66, and 64 MB for $\eta = 1/16$, and 59, 66, and 79 MB as η increases at $s = 128$. For LLaMA 3B, the corresponding values are 0.52, 0.49, and 0.47 GB for $\eta = 1/32$, 0.61, 0.57, and 0.52 GB for $\eta = 1/16$, and 0.49, 0.57, and 0.66 GB as η increases at $s = 128$. Across all three models, increasing s reduces the 4-bit SPARQ state through lower quantization-scale metadata, whereas increasing η increases the storage required by the low-rank factors.

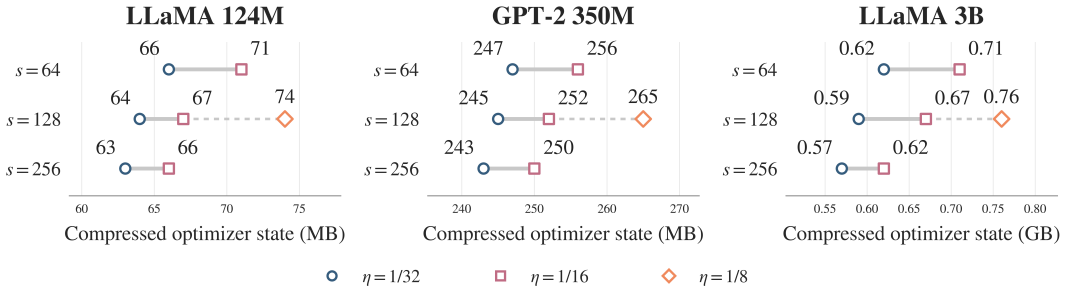


Figure 9: Measured optimizer-state memory of 4-bit SPARQ across models for different group sizes and preserved-subspace rank ratios.

Figure 9 summarizes the measured optimizer-state memory of 4-bit SPARQ across LLaMA 124M, GPT-2 350M, and LLaMA 3B. For LLaMA 124M, increasing s from 64 to 128 and then to 256 reduces the compressed state from 66 MB to 64 MB and then to 63 MB when $\eta = 1/32$, and from 71 MB to 67 MB and then to 66 MB when $\eta = 1/16$. At $s = 128$, increasing η from $1/32$ to $1/16$ and then to $1/8$ increases the state size from 64 MB to 67 MB and then to 74 MB. The same trend is observed for GPT-2 350M. For $\eta = 1/32$, increasing s from 64 to 128 and then to 256 reduces the state size from 247 MB to 245 MB and then to 243 MB, while for $\eta = 1/16$ it decreases from 256 MB to 252 MB and then to 250 MB. At $s = 128$, increasing η from $1/32$ to $1/16$ and then to $1/8$ increases the state size from 245 MB to 252 MB and then to 265 MB. For LLaMA 3B, the

corresponding values are 0.62, 0.59, and 0.57 GB for $\eta = 1/32$, and 0.71, 0.67, and 0.62 GB for $\eta = 1/16$ as s increases from 64 to 128 and then to 256. At $s = 128$, increasing η from $1/32$ to $1/16$ and then to $1/8$ increases the state size from 0.59 GB to 0.67 GB and then to 0.76 GB. Overall, increasing s consistently reduces the compressed optimizer state, whereas increasing η increases the storage required by the low-rank factors.

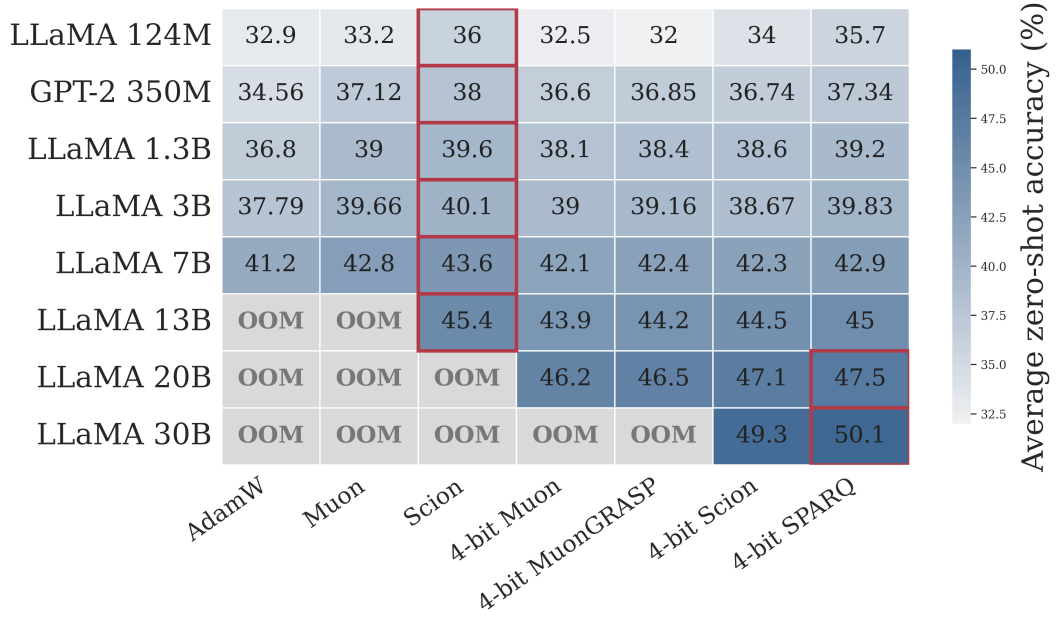


Figure 10: Measured average zero-shot accuracy (%) and memory feasibility across model sizes. Each cell reports the unweighted mean accuracy over ARC-Challenge, ARC-Easy, BoolQ, LAMBADA, PIQA, RACE, and HellaSwag. OOM denotes configurations that do not fit within the assumed memory budget. Red boxes mark the highest accuracy among the feasible methods in each row.