ROTATE ALL, STORE FEW: DECOUPLING COVERAGE FROM STORAGE FOR MEMORY-EFFICIENT TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adam improves when the gradient is normalized in its singular basis, which we call *rotation*. Full-rank rotation exceeds the memory of Adam by storing a basis and states for every direction, whereas low-rank gradient projection rotates only the top-r directions but degrades sharply as r decreases. This is because a single rank r sets both the *rotation coverage*, the number of rotated directions, and the *state storage*, the number of directions with optimizer states. The two decisions nonetheless need not be coupled. The rotation coverage should be full since the whitened update magnitude stays nearly flat along the spectral index, whereas the state storage can be reduced since adjacent second moments have high cosine similarity. We, therefore, propose **COAST**, which decouples rotation **CO**verage **A**nd state **ST**orage. **COAST** rotates every direction but maintains optimizer states for only r stored directions and estimates the updates of the rest from their stored neighbors. On LLaMA pre-training, **COAST** outperforms Adam with 0.42–$0.63\times$ Adam's optimizer memory. It also improves on low-rank methods at similar memory. Halving the storage budget changes perplexity by less than 1% at every scale.

1 INTRODUCTION

LLMs have shown remarkable performance across a wide range of domains (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2019; He et al., 2020; Touvron et al., 2023), and Adam (Kingma & Ba, 2015) has become the standard optimizer for training them. Adam estimates the first and second moments of the gradient coordinate-wise via exponential moving averages, and updates by dividing the first moment by the square root of the second. Such an update is approximately invariant to rescaling of the gradient, which allows stable training with a single learning rate (Kingma & Ba, 2015). However, it requires maintaining gradients and optimizer states for all parameters, which becomes a major bottleneck that limits the scale at which training is feasible as models grow (Shazeer & Stern, 2018; Zhao et al., 2024).

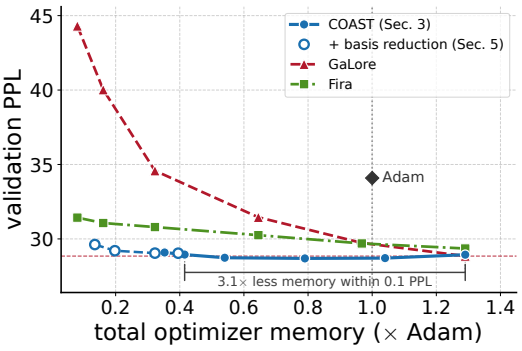
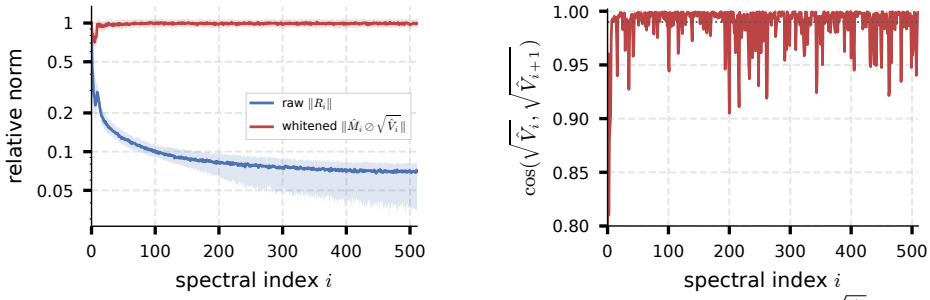


Figure 1: Pareto front of validation PPL vs. total optimizer memory on LLaMA-60M.

One factor that determines the nature of the update is the *basis* in which normalization is performed. Adam normalizes each entry of the gradient independently, so its update is tied to the standard basis. This basis is not a neutral choice: Zhang et al. (2025a) show that Adam degrades under global random rotations of the parameter space, whereas normalizing in the basis of the singular vectors of the gradient improves training over the standard basis. The basis is thus a design choice, and the singular directions are one effective choice. We refer to the alignment of the normalization basis with these directions as *rotation*.

A line of work normalizes the gradient in a rotated basis rather than the standard one. EKFAC (George et al., 2018) normalizes a diagonal preconditioner in the eigenbasis of the K-FAC factors (Martens & Grosse, 2015), and SOAP (Vyas et al., 2025) runs Adam in the eigenbasis of

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.



(a) Raw vs. whitened magnitude (b) Adjacent similarity of $\sqrt{\hat{V}}$

Figure 2: Statistics of the optimizer state from rotated gradient along the spectral index. (a) The raw norm $\|R_i\|$ decays, while the whitened magnitude $\|\hat{M}_i \oslash \sqrt{\hat{V}_i}\|$ stays flat. (b) The adjacent cosine similarity of $\sqrt{\hat{V}}$ exceeds 0.9 for almost all i. Additional visualizations are in Appendix L.

the Shampoo preconditioners (Gupta et al., 2018). Both store full two-sided bases in addition to a second moment for every rotated direction, so the memory exceeds that of Adam.

A different approach rotates only part of the basis. GaLore (Zhao et al., 2024), which was introduced to reduce the memory of optimizer states, takes the top-r singular directions of the gradient as its basis at fixed intervals and runs Adam within that subspace alone. Directions outside the top-r are excluded from both the rotation and the update, so the benefit of rotation is confined to the top-r directions and the resulting performance falls short of Adam. The cause of this memory-performance trade-off is that the rank r makes two decisions at once: ***rotation coverage***, how many directions are rotated and updated, and ***state storage***, how many directions maintain optimizer states. Since GaLore stores states only for the rotated directions, both are bound to a single r, and reducing r to save memory necessarily narrows the coverage. This leads to a natural question.

Can rotation coverage and state storage be decoupled?

We start from two observations about the rotated gradient and the Adam states along the singular directions. We index the rows of the rotated gradient by the axis number of the basis throughout and call it the *spectral index*, which orders the directions by the scale of the gradient along them. Figure 2 (a) shows that the two magnitudes attached to each direction respond to this scale differently. The raw magnitude $\|R_i\|$ of the rotated gradient along direction i follows the rapid decay of the singular values in LLM training (Zhao et al., 2024), whereas the update magnitude $\|\hat{M}_i \oslash \sqrt{\hat{V}_i}\|$ of Adam's step along the same direction stays nearly flat.

Rotation coverage depends on the update magnitude. Since the update magnitude is flat along the index, the top-r directions carry only about r/m of its squared norm and a rank-r truncation discards most of the update rather than a small residual of the gradient. The tail directions are thus small in the gradient yet matter for learning (Liu et al., 2025). What they need is a whitened update rather than the residual itself. GaLore gains almost nothing when the residual is added back at its own magnitude (Chen et al., 2025), whereas Fira recovers part of the gap to full-rank rotation by scaling the residual with a ratio measured on the head. The scale is nonetheless taken from the head rather than from the tail itself, which is suboptimal (See section 6). *State storage* depends on the optimizer state. Figure 2 (a) shows that the update magnitude is nearly constant along the spectral index, and Figure 2 (b) shows that adjacent second moments share their column profile with a cosine similarity above 0.9. These two scales are all an update takes from the state while its direction comes from the current gradient. The second moment whitens the direction and the update magnitude sets the length of the step. Both can be interpolated from stored neighbors, so a few stored directions stand in for the rest and storage needs only a subset of the directions.

Therefore, coverage should be full since the update magnitude is flat along the spectral index whereas storage may be a subset since the state along the same index is largely redundant. The two decisions thus need not be bound to a single rank. We, therefore, propose **COAST**, which decouples rotation **CO**verage **A**nd state **ST**orage: the rotation is performed at full coverage, while optimizer states are maintained for only r *stored directions*, a block at the head and a random sample of the

tail. Each non-stored direction is whitened by the second moment interpolated from its nearest stored neighbors, and its update magnitude is interpolated likewise and scaled by a correction constant. The constant compensates for the difference that arises from following the current gradient rather than the momentum and is obtained in closed form. Across pre-training and fine-tuning, **COAST** surpasses Adam and existing low-rank methods with less memory than Adam even when the storage budget is halved. Section 5 reduces the basis storage further while keeping full coverage. Figure 1 places **COAST** and the low-rank methods on the axis of total optimizer memory on LLaMA-60M. Low-rank GaLore discards every component outside the top-r directions and thus degrades sharply as the coverage shrinks, whereas full-rank GaLore lowers the perplexity substantially below Adam but exceeds its memory by storing a basis and states for every direction. **COAST** keeps the rotation at full rank while reducing the storage budget, attains a lower perplexity than GaLore and Fira at every memory budget below that of Adam, and stays within 0.1 PPL of full-rank GaLore at $0.4\times$ the memory of Adam. Our key contributions are summarized as follows:

- We provide a novel perspective on memory-efficient training that separates the rotation coverage from the state storage. This perspective allows us to reduce memory by narrowing the storage alone, since the update magnitude is flat along the spectral index whereas the second moment along the same index is largely redundant.
- We propose **COAST**, which rotates every direction while maintaining optimizer states for only r of them. The remaining directions are updated from interpolated states with a correction constant derived in closed form (Section 3).
- We demonstrate that **COAST** outperforms Adam with about half of its optimizer memory, and that its perplexity is nearly unchanged when the storage budget is halved, so that performance is determined by the coverage rather than the storage (Section 4).
- We devise a basis reduction that shares a single second moment across the directions outside the stored basis and keeps full coverage down to the memory budget of low-rank methods (Section 5).

2 Related Work & Preliminaries

Full-rank training with Adam. Consider a weight matrix $W_t \in \mathbb{R}^{m\times n}$ at step t, with $m \le n$ assumed throughout without loss of generality. Given the gradient $G_t = \nabla_W f_t(W_t) \in \mathbb{R}^{m\times n}$ of the objective f, full-rank training takes the form

$$W_{t+1} = W_t - \eta\, \psi_t(G_t), \tag{1}$$

where η is the learning rate and ψ_t is the gradient correction function of the optimizer; vanilla SGD corresponds to the identity map $\psi_t(G_t) = G_t$. For LLM training, the de facto choice of ψ_t is Adam (Kingma & Ba, 2015), which tracks exponential moving averages of the gradient and its element-wise square,

$$M_t = \beta_1 M_{t-1} + (1-\beta_1)\, G_t, \qquad V_t = \beta_2 V_{t-1} + (1-\beta_2)\, G_t^{\odot 2}, \tag{2}$$

and normalizes each coordinate of the (bias-corrected) first moment by the square root of the second:

$$\psi_t(G_t) = \hat{M}_t \oslash \big(\sqrt{\hat{V}_t} + \epsilon\big), \qquad \hat{M}_t = \frac{M_t}{1-\beta_1^t}, \quad \hat{V}_t = \frac{V_t}{1-\beta_2^t}, \tag{3}$$

where $\oslash$ denotes element-wise division and ϵ is a small constant for numerical stability. Adam thus stores M_t and V_t, $2mn$ entries in addition to the weights; for billion-parameter models, this state alone often dominates the memory budget of training.

Low-rank gradient projection. GaLore (Zhao et al., 2024) exploits the low-rank structure of the gradient rather than of the weights, reducing the memory of optimizer states while retaining full-parameter learning. Every T steps, the current gradient is decomposed as $G_t = U\Sigma V^\top$, and the top-r left singular vectors form a projection matrix:

$$P_t = [\, u_1, u_2, \ldots, u_r \,] \in \mathbb{R}^{m\times r}. \tag{4}$$

Between two consecutive SVDs, each gradient is projected onto this subspace, Adam is applied to the projected gradient, and the update is projected back to the original space:

$$R_t = P_t^\top G_t \in \mathbb{R}^{r\times n}, \qquad W_{t+1} = W_t - \eta\,\alpha\,P_t\,\psi_t(R_t), \tag{5}$$

where α is a scale factor. Since the moments inside ψ_t are maintained for R_t instead of G_t, the optimizer state shrinks from $2mn$ to $2nr$ entries, plus mr for storing P_t; Flora (Hao et al., 2024) further replaces the SVD with a random projection, lowering the cost of the projection matrix. The residual $G_t - P_tR_t$, however, is discarded: it contributes to neither the optimizer states nor the update, leaving a performance gap relative to full-rank Adam. Fira (Chen et al., 2025) retains this residual by adding it back to the update with a norm-based scaling factor:

$$W_{t+1} = W_t - \eta\,P_t\,\psi_t(R_t) - \eta\,\phi_t(R_t)\,(G_t - P_tR_t), \tag{6}$$

where $\phi_t(R_t)$ is a column-wise vector computed from the low-rank states as $\phi_t(R_t)_i = \|\psi_t(R_{t,:,i})\|/\|R_{t,:,i}\|$, approximating the correction that an adaptive optimizer would apply to the residual. Directions outside the top-r thus take part in the update, but no optimizer state is maintained for the residual, so the correction relies on statistics obtained on the head. FRUGAL (Zmushko et al., 2024) likewise keeps Adam states only for the subspace and updates the residual with state-free signSGD, so the residual is neither rotated nor normalized by a second moment and enters the update as its sign in the standard basis. In each of these methods the rotation covers at most the top-r directions and the residual is either discarded or updated in the standard basis.

Optimizers in a rotated basis. Zhang et al. (2025a) show that Adam degrades under a random rotation of the parameter space and improves when normalized in the singular basis of the gradient. Several optimizers rotate every direction rather than the top-r alone. EKFAC (George et al., 2018) normalizes the gradient by a second moment tracked in the eigenbasis of the K-FAC factors (Martens & Grosse, 2015). SOAP (Vyas et al., 2025) runs Adam in the eigenbasis of the row and column preconditioners of Shampoo (Gupta et al., 2018). Both store two-sided bases together with a second moment for every rotated direction, so their memory exceeds that of Adam. Muon (Liu et al., 2025) explores a similar approach, applying Newton–Schulz orthogonalization to the first moment rather than to the raw gradient and taking the resulting $U_tV_t^\top$ as the update.

Compressing optimizer states. A line of work compresses the optimizer states themselves. Adafactor (Shazeer & Stern, 2018) stores only the row and column sums of the second moment and reconstructs the matrix as their outer product, whereas Adam-mini (Zhang et al., 2025b) partitions the parameters into blocks and shares a single second moment within each block. LoRA-Pre (Wang et al., 2026) reinterprets the exponential moving average (EMA) as an online linear regression on the incoming gradients and compresses the momentum matrix into a low-rank subspace by fitting the regression with two low-rank factors. These methods operate in the coordinate basis and approximate the states by a fixed form such as a factorization or a shared value.

3 Proposed Method

The projection of GaLore makes two decisions at once: (i) *which directions are rotated and updated*, and (ii) *which directions maintain optimizer states*. We formalize the two as follows:

Definition 3.1 (Rotation coverage and state storage)**.** Given a rotated gradient $R_t = P^\top G_t$, we define two index sets over the spectral directions:

$$\mathcal{C} = \big\{\, i : \text{the } i\text{-th direction is rotated and updated} \,\big\}, \tag{7}$$

$$\mathcal{S} = \big\{\, i : \text{optimizer states are maintained for the } i\text{-th direction} \,\big\}. \tag{8}$$

We refer to $\mathcal{C}$ as the *rotation coverage* and $\mathcal{S}$ as the *state storage*. Directions in $\mathcal{S}$ are called *stored directions* and those in $\bar{\mathcal{S}} = \mathcal{C} \setminus \mathcal{S}$ *non-stored directions*.

GaLore couples *rotation coverage* and *state storage*. In GaLore, a single rank r forces both: since Adam states are kept exactly for the projected directions,

$$\mathcal{C} = \mathcal{S} = \{1, \ldots, r\}, \tag{9}$$

so that widening the coverage necessarily inflates the state storage, and shrinking the storage necessarily narrows the coverage. The memory for optimizer states is reduced from $2mn$ to $mr + 2nr$, but the two decisions cannot be controlled separately.

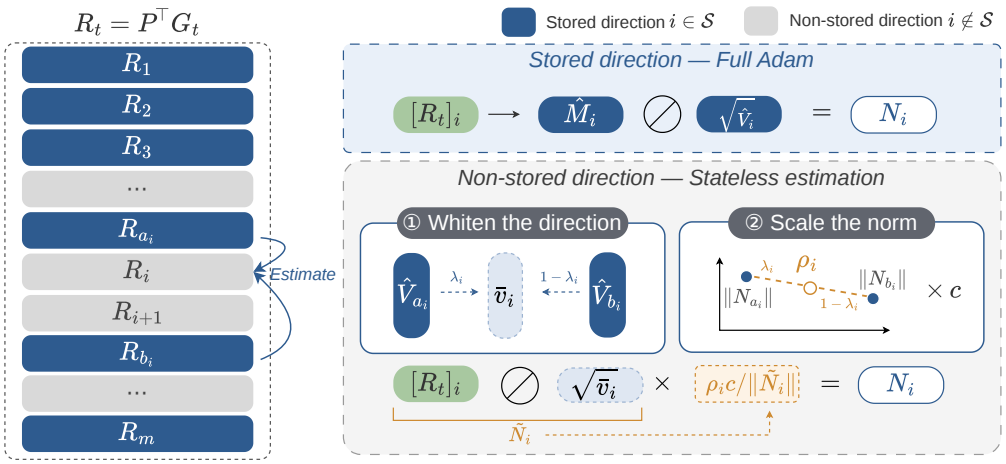


Figure 3: Overall architecture of **COAST**

3.1 Decoupling Rotation Coverage from State Storage

We decouple the two decisions: the rotation coverage is extended to all directions, while optimizer states are maintained only for a small subset:

$$\mathcal{C} = [m] \quad \text{(full-rank rotation, } P \in \mathbb{R}^{m\times m}\text{)}, \qquad \mathcal{S} \subset [m], \quad |\mathcal{S}| = r, \tag{10}$$

where $[m] := \{1, \dots, m\}$. The full-rank rotation discards no singular direction: every direction is whitened in the rotated space, so the tail directions, whose updates are suppressed under the dominant ones in GaLore, participate in the update on an equal footing.

Selecting the stored directions. The storage budget r is split into r_{head} directions kept contiguously at the top and $r_{\text{tail}} = r - r_{\text{head}}$ directions sampled from the tail:

$$\mathcal{S} = \mathcal{S}_{\text{head}} \cup \mathcal{S}_{\text{tail}}, \quad \mathcal{S}_{\text{head}} = \{1, \dots, r_{\text{head}}\}, \quad \mathcal{S}_{\text{tail}} \sim \text{Unif}\big(\{r_{\text{head}}+1, \dots, m\}, r_{\text{tail}}\big), \tag{11}$$

where $\text{Unif}(A, s)$ denotes a subset of size s drawn uniformly at random from A. The two regions follow the two regimes of Figure 2: in the head, every direction holds a state; over the tail, the r_{tail} stored directions are spread across it and each non-stored direction is recovered from its nearest stored directions on either side. We simply split the budget evenly, $r_{\text{head}} = \lceil r/2 \rceil$, throughout.

3.2 Update Rules

3.2.1 Stored Directions: Full Adam

For each stored direction $i \in \mathcal{S}$, we apply standard Adam. Writing $R_t^{\mathcal{S}} = [R_t]_{\mathcal{S}}$ for the stored rows of the rotated gradient,

$$M_t = \beta_1 M_{t-1} + (1-\beta_1)\, R_t^{\mathcal{S}}, \qquad V_t = \beta_2 V_{t-1} + (1-\beta_2)\, \big(R_t^{\mathcal{S}}\big)^{\odot 2}, \tag{12}$$

$$N_i = \hat{M}_i \oslash \big(\sqrt{\hat{V}_i} + \epsilon\big), \qquad i \in \mathcal{S}, \tag{13}$$

where $M_t, V_t \in \mathbb{R}^{r\times n}$ hold one row per stored direction, costing $2nr$ regardless of coverage.

3.2.2 Non-stored Directions: Stateless Estimation

The states of a non-stored direction can be estimated by interpolating those of its neighboring stored directions. For each non-stored direction $i \in \bar{\mathcal{S}}$, we define its neighboring stored directions and the interpolation coefficient as:

$$a_i = \max\{\, j \in \mathcal{S} : j < i \,\}, \qquad b_i = \min\{\, j \in \mathcal{S} : j > i \,\}, \qquad \lambda_i = \frac{i - a_i}{b_i - a_i}. \tag{14}$$

Whitening via interpolated second moment. The denominator is estimated by linearly interpolating the second moments of the two neighboring stored directions, and the raw rotated gradient is whitened by it:

$$\bar{v}_i = (1-\lambda_i)\, \hat{V}_{a_i} + \lambda_i\, \hat{V}_{b_i}, \qquad \tilde{N}_i = [R_t]_i \oslash \big(\sqrt{\bar{v}_i} + \epsilon\big). \tag{15}$$

Norm scaling via interpolated first moment. The update magnitude is aligned to a reference obtained by interpolating the update norms of the two neighboring stored directions:

$$\rho_i = (1-\lambda_i)\left\|N_{a_i}\right\|_2 + \lambda_i \left\|N_{b_i}\right\|_2, \qquad N_i = c \cdot \rho_i \cdot \frac{\tilde{N}_i}{\|\tilde{N}_i\|_2}, \tag{16}$$

where c is a *correction constant* applied only to the non-stored directions (not to the stored ones), which is derived in Section 3.3.

3.2.3 Assembly, Norm-Growth Limiter, and Update

The stored and non-stored rows are assembled into $N_t \in \mathbb{R}^{m\times n}$, and a norm-growth limiter (Chen et al., 2025) is applied row-wise to suppress excessive amplification:

$$s_i = \min\left(1,\ \zeta\, \frac{\|N_i^{(t-1)}\|_2}{\|N_i^{(t)}\|_2}\right), \qquad N_i \leftarrow s_i\, N_i, \tag{17}$$

where $N_i^{(t)}$ denotes the i-th row of N_t at step t, and ζ is the limiter threshold. Finally, the update is rotated back to the original space:

$$W_{t+1} = W_t - \eta\, \alpha\, P N_t. \tag{18}$$

3.3 Scale Correction for Non-stored Directions

Equation 16 sets the magnitude of a non-stored update to $c\rho_i$, where ρ_i estimates the norm of the update that direction i would receive if it were stored. The two updates place the same ρ_i along different directions, and comparing them requires a common reference. We omit the direction subscript and write a row of the rotated gradient as $R_t \in \mathbb{R}^n$ and its bias-corrected EMA as $\hat{M}_t = \sum_{k\ge 0} w_k R_{t-k}$ with the weights of equation 23. Both are taken as whitened by the vector of the current step as in Appendix F.1. The reference is the *momentum axis* $u_t := \hat{M}_t/\|\hat{M}_t\|$. If direction i were stored, its update would be $\hat{M}_t = \rho_i u_t$ and ρ_i would be the displacement per step along the axis. The axis is a function of the gradient sequence rather than of a stored state, so it is defined even for a non-stored direction. A non-stored direction instead spends ρ_i along the current gradient R_t, of which the fraction $\cos\theta_t := \langle R_t, \hat{M}_t\rangle/(\|R_t\|\,\|\hat{M}_t\|)$ lies along the axis.

Assumption 3.2 (Local wide-sense stationarity)**.** There exist $\tau^2 \in (0,\infty)$ and a sequence $\{\varrho_k\}_{k\ge 0}$ with $\varrho_0 = 1$ such that $\mathbb{E}\langle R_{t-k}, R_{t-l}\rangle = \tau^2\, \varrho_{|k-l|}$ for all lags $k, l \ge 0$.

The case $k = l$ states that $\mathbb{E}\|R_{t-k}\|^2$ takes the common value τ^2 at every lag, and the case $k \ne l$ states that the correlation between two rows depends only on the lag between them. We call τ^2 the scale and $\{\varrho_k\}$ the profile. The case $k = l$ is the stationary second moment under which Kingma & Ba (2015) derive the bias correction of Adam, and they note that a departure from stationarity far in the past has little effect because the moving average assigns it small weight. For the same reason only lags within a few multiples of $1/(1-\beta_1)$ contribute appreciably to the sums below and the assumption matters only over this window. The second moment is uncentered as in Adam, so no condition is imposed on the mean of R_t and the assumption is wide-sense stationarity (Brockwell & Davis, 2002) stated for the uncentered second moment rather than the covariance. Since $|\varrho_k| \le 1$ by Cauchy–Schwarz, the series $S_1 := \sum_{k\ge 0} \beta_1^k \varrho_k$ and $S_2 := \sum_{k,l\ge 0} \beta_1^{k+l} \varrho_{|k-l|}$ converge absolutely. Appendix G validates both cases of the assumption empirically.

Lemma 3.3 (Update geometry)**.** *Under Assumption 3.2, as ratios of expectations,*

$$\cos\theta := \frac{\mathbb{E}\langle R_t, \hat{M}_t\rangle}{\sqrt{\mathbb{E}\|R_t\|^2\, \mathbb{E}\|\hat{M}_t\|^2}} = \frac{S_1}{\sqrt{S_2}}, \qquad \sqrt{\frac{\mathbb{E}\|\hat{M}_t\|^2}{\mathbb{E}\|R_t\|^2}} = (1-\beta_1)\sqrt{S_2}, \tag{19}$$

independently of τ^2. For every profile, $\sqrt{1-\beta_1^2} \le \cos\theta \le 1$, with equality on the left iff $S_1 = 1$. The uncorrelated case $\varrho_k = 0$ for $k \ge 1$ has $S_1 = 1$ and gives $\sqrt{1-\beta_1^2}$ and $\sqrt{(1-\beta_1)/(1+\beta_1)}$ for the two ratios.

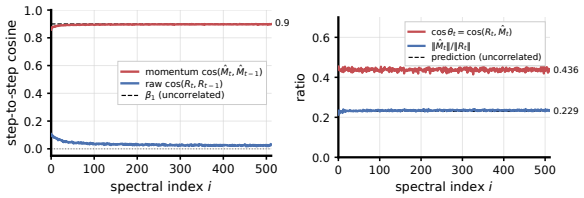
Table 1: Comparison with low-rank algorithms on pre-training various sizes of LLaMA models on the C4 dataset. We report the validation perplexity (↓) on a hold-out C4 test set. The best and second-best performance within the low-rank optimizers are highlighted with **bold** and underline.

Model Size	60M	130M	350M	1B
r/d_{model}	128 / 512	256 / 768	256 / 1024	512 / 2048
Training Tokens	1.1B	2.2B	6.4B	13.1B
Adam (Kingma & Ba, 2015)	34.09	25.08	18.80	15.56
GaLore (Zhao et al., 2024)	34.88	25.36	18.95	15.64
Low-Rank (Kamalakara et al., 2022)	78.18	45.51	37.41	142.53
LoRA (Hu et al., 2021)	34.99	33.92	25.58	19.21
ReLoRA (Lialin et al., 2023)	37.04	29.37	29.08	18.33
SLTrain (Han et al., 2024)	34.15	26.04	19.42	16.14
LORO (Mo et al., 2025)	33.96	24.59	18.84	15.19
Fira (Chen et al., 2025)	30.80	24.51	17.22	14.31
LoRA-Pre (Wang et al., 2026)	32.57	23.78	16.36	13.53
COAST$_r$	**28.74**	**22.11**	**16.14**	13.44
COAST$_{r/2}$	28.95	22.14	16.15	**13.42**

The correction constant c. By Lemma 3.3, $c = 1/\cos\theta = \sqrt{S_2}/S_1$ depends only on $\{\varrho_k\}$ and β_1, not on the gradient magnitude, so a value obtained on stored directions applies to non-stored ones as well. Over all profiles, $1 \le \sqrt{S_2}/S_1 \le 1/\sqrt{1-\beta_1^2}$, with the upper bound attained iff $S_1 = 1$ (Corollary F.1). We fix c at this bound,

$$c = \frac{1}{\sqrt{1-\beta_1^2}}, \tag{20}$$

the value at $\varrho_k = 0$ for $k \ge 1$. Figure 4 (a) illustrates that the step-to-step correlation of the rotated gradient after warmup is about 0.1 in the head and 0.05 in the tail. Figure 4 (b) shows the measured $\cos\theta_t$ within a few percent of $\sqrt{1-\beta_1^2}$ across the spectrum.



(a) Step cosine (b) Prediction vs. measured

Figure 4: (a) Cosine similarities of consecutive rotated gradients and of consecutive first moments. (b) The measured $\cos\theta_t$ and $\|\hat{M}\|/\|R\|$.

4 Experiments

4.1 Memory-Efficient Pre-training

Experimental setup. We follow the protocol of Fira (Chen et al., 2025): LLaMA (Touvron et al., 2023) models of four sizes (60M–1B) are pre-trained on the C4 dataset (Raffel et al., 2020) without data repetition, and we report validation perplexity on a held-out C4 test set. We compare against full-rank Adam and low-rank methods. As in Fira, the method is applied to the linear layers of the attention and MLP blocks. Details are provided in Appendix I. Since the storage budget of our method is decoupled from the coverage, we evaluate two settings: **COAST**$_r$ sets the storage budget equal to the rank r of the baselines, and **COAST**$_{r/2}$ halves it.

Experimental results. Table 1 shows that **COAST**$_r$ reduces validation perplexity by 11.9% to 15.7% relative to Adam while using 0.54–$0.63\times$ of its optimizer memory. Halving the storage budget reduces the memory to 0.42–$0.46\times$ of Adam and changes perplexity by less than one percent at every size, which indicates that the performance is determined by the coverage rather than the storage. Appendix K shows the same ordering up to 350M at $r/4$, where the memory approaches that of GaLore and Fira.

4.2 Memory-Efficient Fine-tuning

Experimental setup. Following the fine-tuning setup of Fira (Chen et al., 2025), we evaluate LLaMA-7B on eight commonsense reasoning sub-tasks. As in Hu et al. (2023), a single model is trained on the merged training sets of the eight sub-tasks and evaluated separately on each test set. As in Fira, **COAST** is applied to the LoRA adapters. Hyper-parameters are detailed in Appendix I.

Table 2: Comparison with fine-tuning methods on commonsense reasoning tasks with LLaMA-7B. The baseline results are taken from Chen et al. (2025) and memory is estimated as in Appendix I.2.

Method	Memory	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
Full-rank	56.00G	64.2	68.1	68.0	42.3	66.5	55.6	43.9	60.0	58.6
Flora	14.34G	50.1	77.5	74.2	53.8	45.5	79.0	64.6	74.8	64.9
ReLoRA	14.45G	68.9	81.2	77.8	46.0	79.4	80.2	64.2	79.6	72.2
LoRA	14.45G	68.9	80.7	77.4	78.1	78.8	77.8	61.3	74.8	74.7
GaLore	14.34G	69.5	82.0	75.1	32.2	18.0	80.7	65.8	78.0	62.7
Fira	14.34G	69.4	82.6	**78.0**	76.8	81.2	82.2	64.4	**80.8**	76.9
COAST$_r$	14.34G	**70.3**	82.3	76.7	**85.5**	80.6	**81.8**	**66.6**	80.0	**78.0**
COAST$_{r/2}$	14.28G	69.9	**83.1**	76.8	85.3	**81.7**	81.7	65.2	78.6	77.8

Experimental results. Table 2 shows that **COAST**$_r$ attains the highest average accuracy and exceeds both Fira and LoRA. The gain is concentrated on HellaSwag, where **COAST**$_r$ improves over Fira by 8.7 points. The two methods are comparable on the remaining tasks. With half the storage budget, **COAST**$_{r/2}$ gives up 0.2 of the average and remains above Fira.

5 Reducing the Memory Cost of the Rotation Basis

As long as the coverage is full-rank, reducing the storage budget leaves performance nearly unchanged. For layers that are close to square ($n \approx m$), however, the basis $P \in \mathbb{R}^{m\times m}$ accounts for a larger share of the total memory. We therefore store only $p := \lceil \gamma m \rceil$ basis columns for a ratio $\gamma \in [r/m, 1]$. Let $\mathcal{H} \subset \{1, \dots, m\}$ denote the p selected indices and $\mathcal{B} := \{1, \dots, m\} \setminus \mathcal{H}$ their complement. We select $\mathcal{H}$ by a rule analogous to equation 11: the top $\lceil r/2 \rceil$ indices are kept contiguously, and the remaining $p - \lceil r/2 \rceil$ are spread over the rest of the spectrum at uniform spacing. On $\mathcal{H}$ we follow the procedure of Section 3, with interpolation weights computed in spectral-index coordinates; at $\gamma = r/m$ every selected direction is an anchor and the interpolation region is empty. The set $\mathcal{B}$ is treated as a single block, whose projection is obtained from the stored $P \in \mathbb{R}^{m\times p}$ alone:

$$G_t^{\text{res}} := G_t - P(P^\top G_t) = P_{\mathcal{B}} P_{\mathcal{B}}^\top G_t. \tag{21}$$

Closed-form block coefficient. The magnitude correction for the block is restricted to a scalar that commutes with the projection, so we target the total magnitude prescribed by the row-wise rule, $\rho^{\text{eff}} := (\sum_{i\in\mathcal{B}} \rho_i^2)^{1/2}$. Since $\|\hat{M}\|/\|R\|$ in Lemma 3.3 does not depend on $\{\tau_j\}$ and is invariant to diagonal scaling, ρ^{eff} admits a closed form (Proof in Appendix H), and the block update becomes

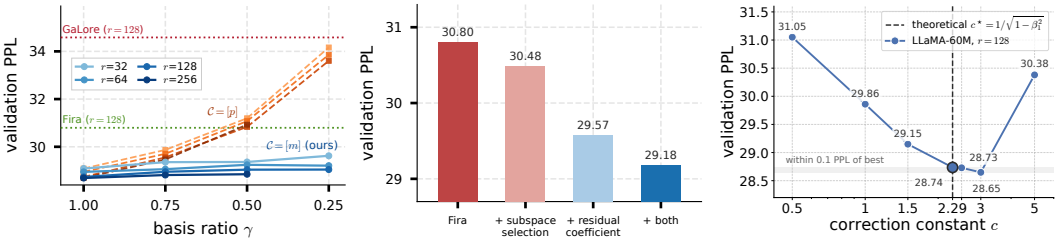
$$N_t^{\mathcal{B}} = \frac{1}{1+\beta_1} G_t^{\text{res}} \oslash \left(\sqrt{\hat{v}^{\mathcal{B}}} + \epsilon\right), \tag{22}$$

where the coefficient $1/(1+\beta_1) = c \cdot \|\hat{M}\|/\|R\|$ involves no free parameter. Here $\hat{v}^{\mathcal{B}}$ is the mean second moment of the tail anchors $\mathcal{S}_{\text{tail}}$; since these anchors are themselves spread over the spectrum, their mean estimates the average second moment of $\mathcal{B}$, obtained without additional state.

Experimental results. Figure 5 (a) varies γ on LLaMA-60M for $r \in \{32, 64, 128, 256\}$. The setting $C = [m]$ updates the complement B as a block by equation 22, whereas $C = [p]$ discards it and covers only the p directions in H. Both settings store p basis columns and $2nr$ entries of state. At $\gamma = r/m$ the latter differs from GaLore only in the selection of the r directions. Reducing γ from 1 to 0.25 raises the PPL of $C = [m]$ by less than 1 at every r and that of $C = [p]$ by about 6 to the level of GaLore. The four curves of $C = [m]$ lie within 0.5 PPL of one another at every γ, and $r = 32$ at $\gamma = 0.25$ still improves on Fira by 2.5% with the basis of Fira and a quarter of its state.

6 Discussion

Memory complexity. **COAST** holds $2nr$ entries of state and m^2 entries of basis; the non-stored directions add no state, since $\bar{v}_i$ and ρ_i are formed from two stored rows at use time, and the row-wise limiter of equation 17 keeps one scalar per direction, m in total. The total $2nr + m^2 + m$ compares with $2mn$ for Adam. Up to the limiter scalars, the ratio to Adam is $r/m + m/(2n)$, largest for a square layer, where it equals $r/m + 1/2$; since $m \le n$, COAST uses less memory than Adam whenever $r < m/2$, as at the ranks used here, and the limiter scalars shift this threshold by $m/(2n) \le 1/2$. The basis reduction of Section 5 shrinks the basis to mp with $p = \lceil \gamma m \rceil$ while the state stays at $2nr$; $\hat{v}^{\mathcal{B}}$ is a mean of stored rows of $\hat{V}_t$ and adds nothing, and the limiter keeps p scalars



(a) Validation PPL against γ (b) Ablation on components (c) Sensitivity to c

Figure 5: Ablation and sensitivity analyses on LLaMA-60M. (a) Validation PPL against the basis ratio γ, where $\mathcal{C} = [m]$ updates the directions outside the basis as a block and $\mathcal{C} = [p]$ discards them. (b) Component-wise ablation over Fira. (c) Sensitivity to the correction constant c.

for the selected rows and one for the block. The total $2nr + mp + p + 1$ decreases linearly in γ and meets GaLore ($2nr + mr$) and Fira ($2nr + mr + 1$) at $\gamma = r/m$ to within the $r + 1$ limiter scalars. Lowering r at $\gamma = 1$ brings the total close to the same budget, and Appendix K reports perplexity against r in this setting. For the architectures of Table 1, this amounts to 0.54–$0.63\times$ of Adam for **COAST**$_r$ and 0.42–$0.46\times$ for **COAST**$_{r/2}$, compared with $1.29\times$ for full-rank rotation. Appendix J reports the measured step time and peak GPU memory.

Comparison with low-rank methods. Low-rank methods rotate and store the same top-r directions and differ only in the remaining ones, which GaLore discards and Fira restores as one block, scaled column-wise by a ratio measured on the top-r. **COAST** differs in three components. (i) ***Structure.*** **COAST** separates the rotated axes from the stored directions. Directions inside the basis but outside the stored set are updated individually from interpolated states, and only the set $\mathcal{B}$ outside the basis $\mathcal{H}$ forms a block (Equation 22). (ii) ***Subspace selection.*** Instead of the top-r, **COAST** keeps the top $\lceil r/2 \rceil$ contiguously and spreads the rest over the whole spectrum, exploiting the structure of the rotated gradient along the spectral index. (iii) ***Residual coefficient.*** GaLore sets it to zero, and Fira multiplies each column of the whole region outside the top-r by one value, the same for every direction. In **COAST**, each interpolated direction receives its own second moment $\bar{v}_i$ and norm $c\rho_i$ through λ_i. The block $\mathcal{B}$ carries a per-column scalar as in Fira but is whitened by the second moment of its own range estimated as the mean over the tail stored directions rather than by a head aggregate, and its magnitude is referred to the axis component of Section 3.3 rather than to the update norm. Under basis reduction at $\gamma = r/m$, **COAST** holds the same budgets as Fira and the interpolated region is empty, so (i) disappears while (ii) and (iii) remain. Figure 5 (b) starts from Fira and changes only (ii) and (iii). Each change lowers the perplexity and both together lower it further.

Sensitivity on c. Figure 5 (c) varies the correction constant on LLaMA-60M at $r = 128$. Between $c = c^\star$ and $c = 3$ the curve is flat and PPL moves by less than 0.1 around the minimum of the sweep, so $c^\star$ lands on the plateau without tuning. Outside this range PPL degrades in both directions. Reducing c to 1 amounts to matching the norm of a non-stored update rather than its component along the momentum axis, and PPL worsens monotonically as c decreases. As c grows the non-stored directions take an excessive share of the update, and PPL worsens again.

7 Conclusions

We proposed **COAST**, which decouples the rotation coverage from the state storage in low-rank gradient projection. The decoupling rests on two observations along the spectral index. The whitened update magnitude stays nearly flat, whereas adjacent second moments have high cosine similarity. **COAST** thus performs the rotation at full rank but maintains optimizer states for only r stored directions. The updates of the non-stored directions are estimated from their stored neighbors with a correction constant derived in closed form. In LLaMA pre-training, **COAST** attains lower perplexity than Adam with about half of its optimizer memory and outperforms the low-rank baselines at a comparable memory budget. Its perplexity changes only slightly as the storage budget is reduced. The basis reduction further stores only part of the basis and updates the remaining directions as a single block. With the same basis size as the low-rank methods and a smaller storage budget, it still attains lower perplexity than them. The ablation confirms that the subspace selection and the residual coefficient each lower the perplexity.

REFERENCES

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 7432–7439, 2020.

Peter J Brockwell and Richard A Davis. *Introduction to time series and forecasting*. Springer, 2002.

Xi Chen, Kaituo Feng, Changsheng Li, Xunhao Lai, Xiangyu Yue, Ye Yuan, and Guoren Wang. Fira: Can we achieve full-rank training of llms under low-rank constraint? *Advances in Neural Information Processing Systems*, 38:120680–120712, 2025.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Thomas George, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent. Fast approximate natural gradient descent in a kronecker factored eigenbasis. *Advances in neural information processing systems*, 31, 2018.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pp. 1842–1850. PMLR, 2018.

Andi Han, Jiaxiang Li, Wei Huang, Mingyi Hong, Akiko Takeda, Pratik Jawanpuria, and Bamdev Mishra. Sltrain: a sparse plus low rank approach for parameter and memory efficient pretraining. *Advances in Neural Information Processing Systems*, 37:118267–118295, 2024.

Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: low-rank adapters are secretly gradient compressors. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, 2024.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 5254–5276, 2023.

Siddhartha Rao Kamalakara, Acyr Locatelli, Bharat Venkitesh, Jimmy Ba, Yarin Gal, and Aidan N Gomez. Exploring low rank training of deep neural networks. *arXiv preprint arXiv:2209.13569*, 2022.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. Relora: High-rank training through low-rank updates. In *The Twelfth International Conference on Learning Representations*, 2023.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Zhanfeng Mo, Long-Kai Huang, and Sinno Jialin Pan. Parameter and memory efficient pretraining via low-rank riemannian optimization. In *International Conference on Learning Representations*, volume 2025, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Taming momentum: Rethinking optimizer states through low-rank approximation. *arXiv preprint arXiv:2602.24283*, 2026.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Tianyue Zhang, Lucas Maes, Alan Milligan, Alexia Jolicoeur-Martineau, Ioannis Mitliagkas, Damien Scieur, Simon Lacoste-Julien, and Charles Guille-Escuret. Understanding adam requires better rotation dependent assumptions. *Advances in Neural Information Processing Systems*, 2025a.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik Durk Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. In *International Conference on Learning Representations*, volume 2025, pp. 28033–28063, 2025b.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 61121–61143. PMLR, 2024.

Philip Zmushko, Aleksandr Beznosikov, Martin Takáč, and Samuel Horváth. Frugal: Memory-efficient optimization by reducing state overhead for scalable training. *arXiv preprint arXiv:2411.07837*, 2024.

A LIMITATION

A limitation of **COAST** is that the split r_{head} of the storage budget between the head and the tail is an additional hyperparameter. We fix the even split throughout without tuning and the ablation in Appendix I.3 shows that perplexity varies by less than 0.4 as the split moves from 0 to $0.75r$. **COAST** is evaluated under the pre-training protocol of GaLore and Fira on LLaMA models up to 1B parameters and under their fine-tuning protocol on LLaMA-7B, so its behavior at larger pre-training scales remains to be examined.

B AI USE STATEMENT

In accordance with ICLR 2027 policy, we acknowledge the use of LLMs in the preparation of this paper. LLM was used to support translation and improve the grammar and readability of the text. The proof strategies presented in Appendices F and H were written by the authors themselves. LLM assisted in writing these proofs and reviewing the derivation process, and the authors verified every step of the final proof.

C ETHICS STATEMENT

We adopt publicly available datasets, including C4 and commonsense reasoning, which are commonly employed in academic research, and all sources have been appropriately cited. This research does not involve any personal or confidential information, thereby eliminating concerns related to privacy. Our proposed approach and the resulting insights contribute to the advancement of artificial intelligence while adhering to principles of ethical innovation and responsibility.

D REPRODUCIBILITY STATEMENT

To ensure the reproducibility and completeness of this paper, we include the Appendix. Appendix E provides our pseudo code for our **COAST** method. The pseudo code helps to implement our **COAST**. All experiments in the paper are reproducible with additional implementation details provided in Appendix I. Our **COAST** code to reproduce the experiments can be found at `https://anonymous.4open.science/r/COAST-28A3`

E COAST IMPLEMENTATION

Algorithms 1 and 2 summarize one step of Sections 3 and 5, respectively.

F PROOFS FOR SECTION 3.3

F.1 PRELIMINARIES

Expanding $M_t = \beta_1 M_{t-1} + (1-\beta_1)R_t$ from $M_0 = 0$ and applying the bias correction $\hat{M}_t = M_t/(1-\beta_1^t)$,

$$\hat{M}_t = \sum_{k=0}^{t-1} w_k R_{t-k}, \qquad w_k = \frac{(1-\beta_1)\beta_1^k}{1-\beta_1^t} \xrightarrow{t\gg 1} (1-\beta_1)\beta_1^k. \tag{23}$$

Below we use the $t \gg 1$ weights and extend the sum to $k \geq 0$.

Let $d_t := (\sqrt{\bar{v}_i} + \epsilon)^{-1}$ be the whitening vector of direction i at step t (Equation 15) and R_s^{raw} the rows before whitening. The applied update is $d_t \odot \sum_k w_k R_{t-k}^{\text{raw}} = \sum_k w_k (d_t \odot R_{t-k}^{\text{raw}})$, so with $R_{t-k} := d_t \odot R_{t-k}^{\text{raw}}$ for every lag, equation 23 holds in the whitened coordinates, $\tilde{N}_i = R_t$, and u_t, $\cos\theta_t$ refer to the applied update. Assumption 3.2 is imposed on these whitened rows. All expectations are over the gradient sequence of direction i with d_t and ρ_i fixed. For a non-stored direction both are built from the states of the neighbors a_i, b_i; for a stored direction, on which $\cos\theta$

Algorithm 1 COAST with Adam. Blue: stored directions $\mathcal{S}$, updated by Adam. Orange: non-stored directions $\bar{\mathcal{S}}$, completed by interpolation from stored neighbors.

Input: step size η, decay rates $\{\beta_1, \beta_2\}$, ϵ; weight $W \in \mathbb{R}^{m\times n}$ $(m \le n)$; storage budget r, split as $r_{\text{head}} + r_{\text{tail}}$ with $r_{\text{head}} = \lceil r/2 \rceil$; switching frequency T, scale factor α, limiter threshold $\zeta = 1.01$; correction constant $c = 1/\sqrt{1-\beta_1^2}$

$M_0, V_0 \in \mathbb{R}^{r\times n} \leftarrow 0, \quad t \leftarrow 1$

repeat

$G_t \leftarrow \nabla_W f_t(W_t) \in \mathbb{R}^{m\times n}$ ▷ full-rank gradient

BASIS AND STORED SET (EVERY T STEPS)

if $(t-1) \bmod T = 0$ **then**

$U, \Sigma, V^\top \leftarrow \text{SVD}(G_t), \quad P \leftarrow U \in \mathbb{R}^{m\times m}$ ▷ full-rank rotation basis

$\mathcal{S}_{\text{head}} \leftarrow \{1, \dots, r_{\text{head}}\}, \quad \mathcal{S}_{\text{tail}} \leftarrow \text{Unif}\big(\{r_{\text{head}}+1, \dots, m\}, r_{\text{tail}}\big), \quad \mathcal{S} \leftarrow \mathcal{S}_{\text{head}} \cup \mathcal{S}_{\text{tail}}$ ▷ eq. 11

end if

$R_t \leftarrow P^\top G_t \in \mathbb{R}^{m\times n}$ ▷ rotate; nothing is discarded

STORED DIRECTIONS $\mathcal{S}$: ADAM

$M_t \leftarrow \beta_1 M_{t-1} + (1-\beta_1)\, R_t^{\mathcal{S}}, \quad V_t \leftarrow \beta_2 V_{t-1} + (1-\beta_2)\, \big(R_t^{\mathcal{S}}\big)^{\odot 2}$ ▷ $r \times n$ states, eq. 12

$\hat{M}_t \leftarrow M_t/(1-\beta_1^t), \quad \hat{V}_t \leftarrow V_t/(1-\beta_2^t)$ ▷ bias correction

$N_i \leftarrow \hat{M}_i \oslash \big(\sqrt{\hat{V}_i} + \epsilon\big)$ for $i \in \mathcal{S}$ ▷ eq. 13

NON-STORED DIRECTIONS $\bar{\mathcal{S}}$: INTERPOLATION

for $i \in \bar{\mathcal{S}}$ **do**

$a_i \leftarrow \max\{j \in \mathcal{S} : j < i\}, \quad b_i \leftarrow \min\{j \in \mathcal{S} : j > i\}, \quad \lambda_i \leftarrow (i-a_i)/(b_i - a_i)$ ▷ eq. 14

$\bar{v}_i \leftarrow (1-\lambda_i)\, \hat{V}_{a_i} + \lambda_i\, \hat{V}_{b_i}, \quad \tilde{N}_i \leftarrow [R_t]_i \oslash \big(\sqrt{\bar{v}_i} + \epsilon\big)$ ▷ whitening, eq. 15

$\rho_i \leftarrow (1-\lambda_i)\, \|N_{a_i}\|_2 + \lambda_i\, \|N_{b_i}\|_2, \quad N_i \leftarrow c\, \rho_i\, \tilde{N}_i / \|\tilde{N}_i\|_2$ ▷ scale + correction, eq. 16

end for

ASSEMBLE, LIMITER, UPDATE

$N_t \leftarrow [\, N_1;\ N_2;\ \dots;\ N_m\,] \in \mathbb{R}^{m\times n}$ ▷ rows in spectral order, $\mathcal{S}$ and $\bar{\mathcal{S}}$ interleaved

$s_i \leftarrow \min\big(1,\ \zeta\, \|N_i^{(t-1)}\|_2 / \|N_i^{(t)}\|_2\big), \quad N_i \leftarrow s_i N_i$ for all i ▷ limiter, eq. 17

$W_{t+1} \leftarrow W_t - \eta\, \alpha\, P N_t, \quad t \leftarrow t+1$ ▷ rotate back, eq. 18

until convergence criteria met

return W_t

is measured, $d_t = (\sqrt{\hat{V}_t} + \epsilon)^{-1}$ varies on the time scale $1/(1-\beta_2)$ of $\hat{V}_t$ and is treated as constant over the window.

F.2 Proof of Lemma 3.3

Proof. By Assumption 3.2, $\mathbb{E}\langle R_{t-k}, R_{t-l}\rangle = \varrho_{|k-l|}\, \tau^2$ for all lags $k, l \ge 0$. Inserting equation 23 termwise,

$$\begin{aligned} \mathbb{E}\langle R_t, \hat{M}_t\rangle &= (1-\beta_1) \sum_{k\ge 0} \beta_1^k \varrho_k\, \tau^2 = (1-\beta_1)\, S_1\, \tau^2, \\ \mathbb{E}\|\hat{M}_t\|^2 &= (1-\beta_1)^2 \sum_{k,l\ge 0} \beta_1^{k+l} \varrho_{|k-l|}\, \tau^2 = (1-\beta_1)^2 S_2\, \tau^2, \\ \mathbb{E}\|R_t\|^2 &= \tau^2, \end{aligned} \tag{24}$$

the series converging absolutely since $|\varrho_k| \le 1$ and $\beta_1 < 1$. The scale τ^2 cancels in both ratios of equation 19. In the uncorrelated case $S_1 = 1$ and $S_2 = \sum_k \beta_1^{2k} = 1/(1-\beta_1^2)$, giving $\cos\theta = \sqrt{1-\beta_1^2}$ and $(1-\beta_1)\sqrt{S_2} = \sqrt{(1-\beta_1)/(1+\beta_1)}$. The bounds on $\cos\theta$ are Corollary F.1. □

Corollary F.1 (Range of S_1 and $\cos\theta$). *$S_2 = (2S_1 - 1)/(1-\beta_1^2)$ and $\frac{1}{1+\beta_1} \le S_1 \le \frac{1}{1-\beta_1}$. Consequently $S_2 > 0$ and*

$$\cos\theta = \sqrt{1-\beta_1^2}\, \frac{S_1}{\sqrt{2S_1 - 1}} \in \Big[\sqrt{1-\beta_1^2},\ 1\Big], \tag{25}$$

with the lower end iff $S_1 = 1$ and the upper end at the two endpoints of the range of S_1.

Algorithm 2 COAST with basis reduction. Blue: stored directions $\mathcal{S}$, updated by Adam. Orange: non-stored directions in $\mathcal{H}$, completed by interpolation from stored neighbors. Green: non-stored directions in $\mathcal{B}$, completed as a single block from the mean tail state, without their basis. $\gamma = 1$ recovers Algorithm 1.

Input: step size η, decay rates $\{\beta_1, \beta_2\}$, ϵ; weight $W \in \mathbb{R}^{m\times n}$ ($m \le n$); storage budget r, split as $r_{\text{head}} + r_{\text{tail}}$ with $r_{\text{head}} = \lceil r/2 \rceil$; basis ratio $\gamma \in [r/m, 1]$, basis size $p = \lceil \gamma m \rceil$; switching frequency T, scale factor α, limiter threshold $\zeta = 1.01$; correction constant $c = 1/\sqrt{1-\beta_1^2}$

$M_0, V_0 \in \mathbb{R}^{r\times n} \leftarrow 0, \quad t \leftarrow 1$

repeat

$G_t \leftarrow \nabla_W f_t(W_t) \in \mathbb{R}^{m\times n}$ ▷ full-rank gradient

BASIS AND INDEX SETS (EVERY T STEPS)

if $(t-1) \bmod T = 0$ **then**

$U, \Sigma, V^\top \leftarrow \text{SVD}(G_t)$

$\mathcal{H} \leftarrow \{1, \dots, r_{\text{head}}\} \cup \left\{ r_{\text{head}} + \lfloor j\,(m - r_{\text{head}})/(p - r_{\text{head}}) \rfloor : j = 1, \dots, p - r_{\text{head}} \right\}$ ▷ head contiguous, $p - r_{\text{head}}$ at uniform spacing; $\mathcal{B} = [m] \setminus \mathcal{H}$

$P \leftarrow U_{:,\mathcal{H}} \in \mathbb{R}^{m\times p}$ ▷ stored basis; $U_{:,\mathcal{B}}$ is discarded

$\mathcal{S}_{\text{head}} \leftarrow \{1, \dots, r_{\text{head}}\}, \quad \mathcal{S}_{\text{tail}} \leftarrow \text{Unif}\big(\mathcal{H} \setminus \mathcal{S}_{\text{head}},\, r_{\text{tail}}\big), \quad \mathcal{S} \leftarrow \mathcal{S}_{\text{head}} \cup \mathcal{S}_{\text{tail}}$ ▷ eq. 11 within $\mathcal{H}$; $\mathcal{S} = \mathcal{H}$ at $\gamma = r/m$

end if

$R_t \leftarrow P^\top G_t \in \mathbb{R}^{p\times n}$ ▷ rotate $\mathcal{H}$ only; $[R_t]_i$ is the row of direction $i \in \mathcal{H}$

$G_t^{\text{res}} \leftarrow G_t - P R_t \in \mathbb{R}^{m\times n}$ ▷ $= P_{\mathcal{B}} P_{\mathcal{B}}^\top G_t$, obtained without $P_{\mathcal{B}}$, eq. 21

STORED DIRECTIONS $\mathcal{S}$: ADAM

$M_t \leftarrow \beta_1 M_{t-1} + (1-\beta_1)\, R_t^{\mathcal{S}}, \quad V_t \leftarrow \beta_2 V_{t-1} + (1-\beta_2)\, \big(R_t^{\mathcal{S}}\big)^{\odot 2}$ ▷ $r \times n$ states, eq. 12

$\hat{M}_t \leftarrow M_t/(1-\beta_1^t), \quad \hat{V}_t \leftarrow V_t/(1-\beta_2^t)$ ▷ bias correction

$N_i \leftarrow \hat{M}_i \oslash \big(\sqrt{\hat{V}_i} + \epsilon\big) \;\text{ for } i \in \mathcal{S}$ ▷ eq. 13

NON-STORED DIRECTIONS IN $\mathcal{H}$: INTERPOLATION

for $i \in \mathcal{H} \setminus \mathcal{S}$ **do**

$a_i \leftarrow \max\{j \in \mathcal{S} : j < i\}, \quad b_i \leftarrow \min\{j \in \mathcal{S} : j > i\}, \quad \lambda_i \leftarrow (i - a_i)/(b_i - a_i)$ ▷ eq. 14, spectral-index coordinates

$\bar{v}_i \leftarrow (1-\lambda_i)\, \hat{V}_{a_i} + \lambda_i\, \hat{V}_{b_i}, \quad \tilde{N}_i \leftarrow [R_t]_i \oslash \big(\sqrt{\bar{v}_i} + \epsilon\big)$ ▷ whitening, eq. 15

$\rho_i \leftarrow (1-\lambda_i)\, \|N_{a_i}\|_2 + \lambda_i\, \|N_{b_i}\|_2, \quad N_i \leftarrow c\, \rho_i\, \tilde{N}_i / \|\tilde{N}_i\|_2$ ▷ scale + correction, eq. 16

end for

NON-STORED DIRECTIONS IN $\mathcal{B}$: BLOCK

$\hat{v}^{\mathcal{B}} \leftarrow \frac{1}{r_{\text{tail}}} \sum_{j\in\mathcal{S}_{\text{tail}}} \hat{V}_j \in \mathbb{R}^n$ ▷ mean second moment of the tail anchors; no additional state

$N_t^{\mathcal{B}} \leftarrow \frac{1}{1+\beta_1}\, G_t^{\text{res}} \oslash \big(\sqrt{\hat{v}^{\mathcal{B}}} + \epsilon\big) \in \mathbb{R}^{m\times n}$ ▷ closed-form coefficient

$1/(1+\beta_1) = c\sqrt{(1-\beta_1)/(1+\beta_1)}$, eq. 22

ASSEMBLE, LIMITER, UPDATE

$N_t \leftarrow [\, N_i \,]_{i\in\mathcal{H}} \in \mathbb{R}^{p\times n}$ ▷ p rows in spectral order, $\mathcal{S}$ and $\mathcal{H} \setminus \mathcal{S}$ interleaved

$s_i \leftarrow \min\big(1,\, \zeta\, \|N_i^{(t-1)}\|_2 / \|N_i^{(t)}\|_2\big), \quad N_i \leftarrow s_i N_i \;\text{ for } i \in \mathcal{H}$ ▷ limiter, eq. 17

$s^{\mathcal{B}} \leftarrow \min\big(1,\, \zeta\, \|N_{t-1}^{\mathcal{B}}\|_F / \|N_t^{\mathcal{B}}\|_F\big), \quad N_t^{\mathcal{B}} \leftarrow s^{\mathcal{B}} N_t^{\mathcal{B}}$ ▷ limiter on the block as a whole

$W_{t+1} \leftarrow W_t - \eta\, \alpha\, \big(P N_t + N_t^{\mathcal{B}}\big), \quad t \leftarrow t+1$ ▷ rotate back on $\mathcal{H}$; $N_t^{\mathcal{B}}$ is already in the original coordinates

until convergence criteria met

return W_t

Proof. Splitting the double sum into $k = l$ and $k \neq l$, with $q = |k - l| \ge 1$ in the latter, $S_2 = \sum_k \beta_1^{2k} + 2\sum_{q\ge1} \varrho_q \beta_1^q \sum_{k\ge0} \beta_1^{2k} = (2S_1 - 1)/(1-\beta_1^2)$. By Cauchy–Schwarz, $\mathbb{E}\langle R_t, \hat{M}_t\rangle \le \sqrt{\mathbb{E}\|R_t\|^2\, \mathbb{E}\|\hat{M}_t\|^2}$, which by equation 24 reads $S_1^2 \le S_2 = (2S_1 - 1)/(1-\beta_1^2)$. The solution set of $(1-\beta_1^2)S_1^2 - 2S_1 + 1 \le 0$ is the interval $[\frac{1}{1+\beta_1}, \frac{1}{1-\beta_1}]$, on which $2S_1 - 1 \ge \frac{1-\beta_1}{1+\beta_1} > 0$; both endpoints are attained, by $\varrho_k \equiv 1$ and $\varrho_k = (-1)^k$. Hence $\cos\theta = S_1/\sqrt{S_2}$ is equation 25. $\cos^2\theta \le 1$ is the same inequality, with equality at the two endpoints, and $\cos^2\theta \ge 1 - \beta_1^2$ is $S_1^2 \ge 2S_1 - 1$, i.e. $(S_1 - 1)^2 \ge 0$. □

F.3 THE CORRECTION CONSTANT

Lemma 3.3 gives $\cos\theta$ as a ratio of expectations, whereas the fraction that a non-stored update places along the axis at step t is the realized $\cos\theta_t$. The following lemma shows that $c = 1/\cos\theta$ places ρ_i on the axis in expectation up to an error of second order in the step-to-step fluctuations of the inner product and the norms. Write $\bar{b} := \sqrt{\mathbb{E}\|R_t\|^2\,\mathbb{E}\|\hat{M}_t\|^2}$ for the denominator of equation 19 and $\xi_t := \|R_t\|\,\|\hat{M}_t\|/\bar{b}$ for the realized norm product relative to it. Beyond $\cos\theta > 0$ from Corollary F.1 the bound uses only the finiteness of $\mathbb{E}[\|R_t\|^2\|\hat{M}_t\|^2]$ needed for $\mathrm{Var}(\xi_t)$ to exist.

Lemma F.2 (Identification). *Let $c := 1/\cos\theta$. The non-stored update $N_i = c\rho_i R_t/\|R_t\|$ satisfies $\mathbb{E}\langle N_i, u_t\rangle = \rho_i(1+\delta)$ with*

$$|\delta| \le c\Big[(1-\mathbb{E}\xi_t) + \sqrt{\mathrm{Var}(\xi_t)\,\mathrm{Var}(\cos\theta_t)}\Big] \tag{26}$$

and $0 \le \mathbb{E}\xi_t \le 1$. By Corollary F.1 the constant satisfies $c \le c_0 := 1/\sqrt{1-\beta_1^2}$.

Proof. Since $N_i = c\rho_i R_t/\|R_t\|$ and $u_t = \hat{M}_t/\|\hat{M}_t\|$, the definition of $\cos\theta_t$ gives $\langle N_i, u_t\rangle = c\rho_i\cos\theta_t$ at every step. Taking the expectation with ρ_i and d_t fixed as in Appendix F.1 and inserting $c = 1/\cos\theta$ yields

$$\mathbb{E}\langle N_i, u_t\rangle = \rho_i\,\frac{\mathbb{E}[\cos\theta_t]}{\cos\theta} \qquad \text{so that} \qquad \delta = \frac{\mathbb{E}[\cos\theta_t] - \cos\theta}{\cos\theta}. \tag{27}$$

It remains to bound the numerator. The definition of $\cos\theta_t$ reads $\langle R_t, \hat{M}_t\rangle = \|R_t\|\,\|\hat{M}_t\|\cos\theta_t$ at every step. Taking the expectation and dividing by the constant $\bar{b}$ turns the left side into $\cos\theta$ by equation 19 and the right side into $\mathbb{E}[\xi_t\cos\theta_t]$. The definition of the covariance expands the right side as $\mathbb{E}[\xi_t]\,\mathbb{E}[\cos\theta_t] + \mathrm{Cov}(\xi_t, \cos\theta_t)$. Subtracting the resulting expression for $\cos\theta$ from $\mathbb{E}[\cos\theta_t]$ gives

$$\mathbb{E}[\cos\theta_t] - \cos\theta = (1-\mathbb{E}\xi_t)\,\mathbb{E}[\cos\theta_t] - \mathrm{Cov}(\xi_t, \cos\theta_t). \tag{28}$$

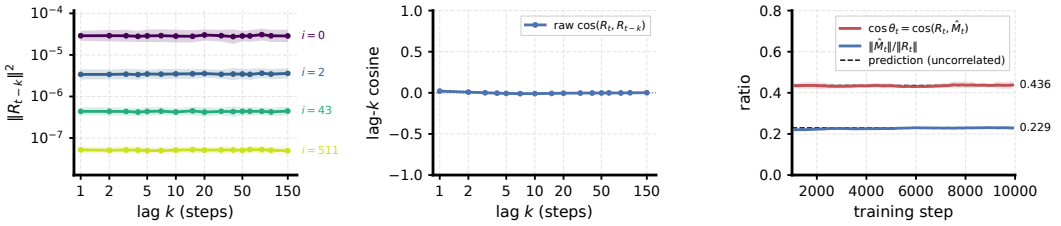
Cauchy–Schwarz gives $\mathbb{E}[\|R_t\|\,\|\hat{M}_t\|] \le \bar{b}$ and hence $\mathbb{E}\xi_t \le 1$, while $\mathbb{E}\xi_t \ge 0$ because $\xi_t \ge 0$. Since $|\cos\theta_t| \le 1$, the first term of equation 28 is at most $1 - \mathbb{E}\xi_t$ in absolute value. The covariance is at most $\sqrt{\mathrm{Var}(\xi_t)\,\mathrm{Var}(\cos\theta_t)}$ in absolute value by Cauchy–Schwarz. The triangle inequality and $|\delta| = c\,|\mathbb{E}[\cos\theta_t] - \cos\theta|$ give equation 26. The bound $c \le c_0$ is $\cos\theta \ge \sqrt{1-\beta_1^2}$ of Corollary F.1. □

Both terms of equation 26 are of second order in these fluctuations. With $z_1 := \|R_t\|/\sqrt{\mathbb{E}\|R_t\|^2}$ and $z_2 := \|\hat{M}_t\|/\sqrt{\mathbb{E}\|\hat{M}_t\|^2}$ we have $\mathbb{E}z_1^2 = \mathbb{E}z_2^2 = 1$ and $\xi_t = z_1 z_2$. The identity $z_1 z_2 = \frac{1}{2}(z_1^2 + z_2^2) - \frac{1}{2}(z_1 - z_2)^2$ then gives $1 - \mathbb{E}\xi_t = \frac{1}{2}\,\mathbb{E}(z_1 - z_2)^2$ and the second term is a product of two standard deviations. The three quantities $\langle R_t, \hat{M}_t\rangle$, $\|R_t\|^2$ and $\|\hat{M}_t\|^2$ are sums over the n coordinates of the row. Under weak dependence across coordinates their relative fluctuations are of order $n^{-1/2}$ and both terms are $O(1/n)$. The averaged $\cos\theta_t$ in Figure 4 (b) estimates $\mathbb{E}[\cos\theta_t]$ and lies within a few percent of $\cos\theta$. By Corollary F.1 fixing c at c_0 as in equation 20 multiplies the expected axis component $\rho_i(1+\delta)$ by $c_0\cos\theta = (1-(1-1/S_1)^2)^{-1/2} = 1 + O((1-1/S_1)^2)$.

G EMPIRICAL VALIDATION OF ASSUMPTION 3.2

Assumption 3.2 has two conditions: the scale $\mathbb{E}\|R_{t-k}\|^2$ is the same τ^2 at every lag k in the window, and the correlation between two rows depends only on the lag between them. We check both, and their consequence for c, on LLaMA-60M with $r = m = 512$, so that optimizer states are kept at every spectral index.

Validation on scale τ^2. The first condition of Assumption 3.2 is $\mathbb{E}\|R_{t-k}\|^2 = \tau^2$ for every lag k in the window, i.e., the squared norm of a row does not depend on how many steps ago it was taken. Figure 6 (a) plots $\|R_{t-k}\|^2$ against the lag k: every line is flat, so the condition holds. Between $i = 0$ and $i = 511$, $\|R_{t-k}\|^2$ differs by about 10^3, but this does not affect c: by Lemma 3.3 the numerator and the denominator of $\cos\theta$ are both proportional to τ^2, so τ^2 cancels and only the profile can change $\cos\theta$, and hence c.



(a) $\|R_{t-k}\|^2$ along the lag (b) $\cos(R_t, R_{t-k})$ along the lag (c) Prediction vs. measured

Figure 6: Empirical validation of Assumption 3.2. (a) $\|R_{t-k}\|^2$ against the lag k at selected spectral indices. (b) $\cos(R_t, R_{t-k})$ against the lag, averaged over spectral indices and steps. (c) $\cos\theta_t$ and $\|\hat{M}_t\|/\|R_t\|$ over training.

Validation on profile ϱ_k. The second condition of Assumption 3.2 is that the correlation between two rows depends only on the lag between them, $\mathbb{E}\langle R_{t-k}, R_{t-l}\rangle = \tau^2 \varrho_{|k-l|}$; it does not restrict the shape of $\{\varrho_k\}$, so what matters for c is how large the profile is. Figure 6 (b) plots the cosine between R_t and R_{t-k} against the lag k, averaged over spectral indices and steps, which estimates ϱ_k: it is close to zero at every lag. By Corollary F.1, a profile this small moves $c = 1/\cos\theta$ away from $1/\sqrt{1-\beta_1^2}$ only to second order in $1 - 1/S_1$, so the correction constant of equation 20, derived for $\varrho_k = 0$, remains valid.

Consequence for c. Figure 6 (c) plots $\cos\theta_t$ and $\|\hat{M}_t\|/\|R_t\|$ over training. Both stay at the values $\sqrt{1-\beta_1^2} = 0.436$ and $\sqrt{(1-\beta_1)/(1+\beta_1)} = 0.229$ that Lemma 3.3 predicts for the uncorrelated case, and Figure 4 (b) shows the same agreement against the spectral index. Both conditions and their consequence are thus confirmed: Assumption 3.2 holds over the window of $\hat{M}_t$, and $c = 1/\sqrt{1-\beta_1^2}$ agrees with the measurements throughout training and across the spectrum.

H Proof for Section 5

Proposition H.1 (Block update without its basis)**.** *If all directions in $\mathcal{B}$ share a common second moment $\hat{v}^{\mathcal{B}} \in \mathbb{R}^n$ and $D := \operatorname{diag}\big((\sqrt{\hat{v}^{\mathcal{B}}} + \epsilon)^{-1}\big)$, then*

$$P_{\mathcal{B}}\big(R_t^{\mathcal{B}} D\big) = G_t^{res} D, \qquad \big\|R_t^{\mathcal{B}} D\big\|_F = \big\|G_t^{res} D\big\|_F. \tag{29}$$

Proof. Under the hypothesis every row of $\mathcal{B}$ is whitened by the same vector, so the element-wise whitening of the block is the right multiplication $R_t^{\mathcal{B}} \oslash (\sqrt{\hat{v}^{\mathcal{B}}} + \epsilon) = R_t^{\mathcal{B}} D$; this is the only place the hypothesis enters. By associativity and equation 21, $P_{\mathcal{B}}(R_t^{\mathcal{B}} D) = (P_{\mathcal{B}} P_{\mathcal{B}}^\top G_t) D = G_t^{\text{res}} D$, which is the first relation: the block update depends on $P_{\mathcal{B}}$ only through the projector $P_{\mathcal{B}} P_{\mathcal{B}}^\top = I - PP^\top$, with $P \in \mathbb{R}^{m\times p}$ the stored basis. The second follows from the orthonormal columns of $P_{\mathcal{B}}$: $\|P_{\mathcal{B}} X\|_F^2 = \operatorname{tr}(X^\top P_{\mathcal{B}}^\top P_{\mathcal{B}} X) = \|X\|_F^2$ with $X = R_t^{\mathcal{B}} D$. □

Closed-form block coefficient. Applying equation 16 to the block, with $R_t^{\mathcal{B}} D$ in place of $\tilde{N}_i$ and its norm taken over the block as a whole since its rows are not available individually, and rotating back with equation 29, the block update in the original coordinates is

$$N_t^{\mathcal{B}} = c\,\rho^{\text{eff}} \frac{G_t^{\text{res}} D}{\|G_t^{\text{res}} D\|_F}, \qquad \rho^{\text{eff}} := \Big(\sum_{i\in\mathcal{B}} \rho_i^2\Big)^{1/2}. \tag{30}$$

By the convention of Appendix F, $\rho_i = \|\hat{M}_i D\|$ is the norm of the whitened momentum that direction i would carry if it were stored; equation 16 estimates it from stored neighbors, whereas here it is obtained from the gradient of the block itself. Summing over $i \in \mathcal{B}$ gives $(\rho^{\text{eff}})^2 = \|\hat{M}_t^{\mathcal{B}} D\|_F^2$. For each row of $\mathcal{B}$, Lemma 3.3 in the uncorrelated case gives $\mathbb{E}\|\hat{M}_i D\|^2 = \frac{1-\beta_1}{1+\beta_1}\mathbb{E}\|R_i D\|^2$; the

constant involves neither $\{\tau_j\}$ nor D, so it is the same for every row and the relation survives the sum over $\mathcal{B}$. Identifying the realized squared norms with their expectations then gives

$$\rho^{\text{eff}} = \sqrt{\frac{1-\beta_1}{1+\beta_1}}\,\left\|R_t^{\mathcal{B}} D\right\|_F = \sqrt{\frac{1-\beta_1}{1+\beta_1}}\,\left\|G_t^{\text{res}} D\right\|_F, \tag{31}$$

the second equality by equation 29. The identification is accurate to relative order $((m-p)n)^{-1/2}$ under the weak dependence across coordinates invoked in Appendix F.3 since the sums run over $(m-p)n$ summands rather than n. Substituting into equation 30 cancels $\|G_t^{\text{res}} D\|_F$, and with $c = 1/\sqrt{1-\beta_1^2}$,

$$c\sqrt{\frac{1-\beta_1}{1+\beta_1}} = \frac{1}{\sqrt{(1-\beta_1)(1+\beta_1)}} \cdot \sqrt{\frac{1-\beta_1}{1+\beta_1}} = \frac{1}{1+\beta_1}, \tag{32}$$

which is equation 22.

I Experimental Setup

I.1 Memory-Efficient Pre-training

This section describes the LLaMA architectures and the hyper-parameters used for pre-training. To ensure a fair comparison, we adopt the settings of GaLore (Zhao et al., 2024) and Fira (Chen et al., 2025) without modification. Table 3 lists the architectures across model sizes. For all sizes, the maximum sequence length is 256 and the batch size is 131K tokens. The learning rate is warmed up over the first 10% of the training steps and then follows cosine annealing down to 10% of its peak value. All models are trained in BF16 on the C4 dataset without data repetition. As in the GaLore implementation, the rotation is applied to every linear layer of the attention and MLP blocks, while the embedding and output layers are trained with regular Adam. Results of the baselines in Table 1 are taken from Chen et al. (2025) and Wang et al. (2026); Fira at 60M is reproduced with the official implementation under the same setting.

Table 3: Hyper-parameters of LLaMA architectures for pre-training

Params	Hidden	Intermediate	Heads	Layers	Steps
60M	512	1376	8	8	10K
130M	768	2048	12	12	20K
350M	1024	2736	16	24	60K
1B	2048	5461	24	32	100K

Optimizer hyper-parameters. **COAST** shares all common hyper-parameters with Fira and GaLore without further tuning: the learning rate $\eta = 0.01$, the switching frequency $T = 200$, and the limiter threshold $\zeta = 1.01$ for all model sizes, and the scale factor $\alpha = 0.25$. Adam uses $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ throughout. Beyond these, **COAST** has the storage budget r and its split into $r_{\text{head}} + r_{\text{tail}}$: **COAST**$_r$ sets r equal to the rank of the low-rank baselines and **COAST**$_{r/2}$ halves it (Table 4), with the even split $r_{\text{head}} = r_{\text{tail}} = r/2$ in both cases (Section 3.1). The correction constant is not tuned but fixed to its closed form $c = 1/\sqrt{1-\beta_1^2} \simeq 2.29$ at $\beta_1 = 0.9$ (Equation 20). For the basis-reduction experiments of Section 5, the basis ratio $\gamma \in \{0.25, 0.5, 0.75, 1\}$ is varied with $r \in \{32, 64, 128, 256\}$ on LLaMA-60M with all other settings unchanged.

I.2 Memory-Efficient fine-tuning

We fine-tune the pre-trained LLaMA-7B (Touvron et al., 2023) on the commonsense reasoning benchmark of Hu et al. (2023), which consists of eight sub-tasks: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-e (Clark et al., 2018), ARC-c (Clark et al., 2018), and OBQA (Mihaylov et al., 2018). A single model is trained on the union of the eight training sets

Table 4: Hyper-parameter configurations of pre-training for **COAST**

Model Size	60M	130M	350M	1B
d_{model}	512	768	1024	2048
r (**COAST**$_r$)	128	256	256	512
r (**COAST**$_{r/2}$)	64	128	128	256
α	0.25			

and evaluated on the test set of each sub-task. Table 5 lists the configuration. All hyper-parameters are taken from Fira (Chen et al., 2025); **COAST**$_r$ uses the storage budget $r = 16$, equal to the rank of Fira, and **COAST**$_{r/2}$ uses $r = 8$.

Table 5: Hyper-parameter configurations of fine-tuning for **COAST**

Hyper-parameters	Setting
r_{LoRA}	32
r (**COAST**$_r$ / **COAST**$_{r/2}$)	16 / 8
T	200
ζ	1.01
Dropout	0.05
Base optimizer	Adam
LR	1e-4
LR Scheduler	Linear
Batch size	16
Warm-up steps	100
Epochs	3
Where	Q, K, V, Up, Down

Memory estimates. We adopt the memory estimation of Fira (Chen et al., 2025). The parameters, gradients and optimizer states of each method are counted and the total is multiplied by 2 bytes as in BF16 storage. For fine-tuning the frozen LLaMA-7B backbone is counted as 7B parameters and the adapter states are counted from the optimizer state tensors. However, the fine-tuning memory values reported by Chen et al. (2025) could not be reproduced from their own procedure, so Table 2 takes only the accuracies from their paper and reports memory recomputed under the same procedure.

I.3 Ablation on the Budget Split

Table 6 varies the split of the storage budget r between the head and the tail on LLaMA-60M at $r = 128$. At $r_{\text{head}} = 0$, all r stored directions are spread over the spectrum, whereas a larger r_{head} keeps more of the head directions exact and leaves fewer to be spread over the tail. Perplexity varies by less than 0.4 across the four settings, which indicates that the head directions can also be recovered by interpolation. Even so, $r_{\text{head}} = 0$ gives the highest perplexity and $r_{\text{head}} = 0.75r$ is slightly worse than the even split, since the former leaves the largest-gradient directions interpolated and the latter leaves fewer stored directions for the tail. The even split $r_{\text{head}} = \lceil r/2 \rceil$ is the minimum of the sweep and is used throughout without tuning.

Table 6: Validation perplexity of **COAST** on LLaMA-60M ($r = 128$) for different splits of the storage budget.

r_{head}	0	$0.25r$	$0.5r$ (default)	$0.75r$
PPL	29.13	28.89	**28.74**	28.84

J Empirical Complexity analysis

Wall-clock time. Table 7 reports the wall-clock time per step of **COAST** and GaLore for pre-training LLaMA-130M on C4 and for fine-tuning LLaMA-7B on commonsense reasoning. Each value is the average over the first 1,000 steps on a single A100 with the same batch size for every method. In pre-training the step time of both **COAST** variants is to that of GaLore and differs by less than 1%. Halving the number of stored directions from r to $r/2$ does not change the step time in either setting.

Table 7: Wall-clock time per step (s), averaged over the first 1,000 steps.

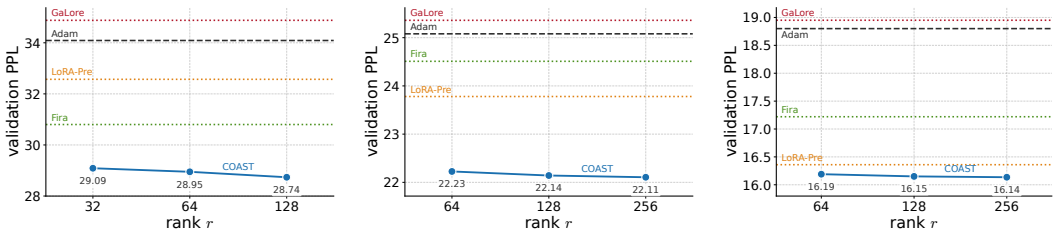
Method	Pre-training	Fine-tuning
GaLore	1.075	0.964
COAST$_r$	1.068	1.021
COAST$_{r/2}$	1.067	1.019

Peak memory usage. Table 8 reports the peak GPU memory of Adam, GaLore and **COAST** for pre-training LLaMA-130M on C4 and for fine-tuning LLaMA-7B on commonsense reasoning. The peak memory is the maximum memory allocated on the GPU over the first 1,000 steps as reported by PyTorch. Every method uses the same batch size within each setting. In fine-tuning only the LoRA adapters are trainable and the optimizer state is kept for the adapters while the base model is held in half precision. In pre-training the peak memory of both **COAST** variants is within 0.4% of GaLore and below Adam. In fine-tuning **COAST**$_r$ matches GaLore, and **COAST**$_{r/2}$ lowers the peak memory by 0.5%.

Table 8: Peak GPU memory (GB) for pre-training LLaMA-130M on C4 and LoRA fine-tuning LLaMA-7B on Commonsense-170K.

	Pre-training (LLaMA-130M)		Fine-tuning (LLaMA-7B, LoRA)	
Optimizer	r	Peak (GB)	r	Peak (GB)
Adam	–	20.75	–	20.25
GaLore	256	20.57	16	20.04
COAST$_r$	256	20.64	16	20.04
COAST$_{r/2}$	128	20.59	8	19.94

K Perplexity against the storage budget



(a) 60M (b) 130M (c) 350M

Figure 7: Validation perplexity against the storage budget r at $\gamma = 1$. Horizontal lines mark Adam and the low-rank baselines of Table 1 at the rank of the table.

Figure 7 reduces the storage budget r on LLaMA-60M, 130M and 350M with the coverage held at $\gamma = 1$ and marks Adam together with GaLore, Fira and LoRA-Pre of Table 1 at their reported ranks. The curves are nearly flat and dividing r by four raises the perplexity by only 1.2% on 60M, 0.5% on

130M and 0.3% on 350M. The memory at $r/4$ comes close to that of GaLore and Fira at r and falls below that of LoRA-Pre whose two factors of rank r per moment amount to $2(m+n)r$ entries, so the reduced settings meet the baselines at a comparable budget. Even at this budget every setting of **COAST** stays below every low-rank baseline of Table 1 and the gap to the closest line of the figure is 5.6% on 60M, 6.5% on 130M and 1.0% on 350M. Perplexity thus follows the coverage rather than the number of stored directions and the budget of the low-rank methods is reached without giving up the advantage.

L Statistics of Optimizer State Across Layers and Modules

Figure 2 shows a single module (`k_proj`, layer 0). Figures 8–9 repeat the same measurements for every module of layers 0, 4, and 7 of LLaMA-60M, with $r = m = 512$ so that $\hat{M}_i$ and $\hat{V}_i$ are kept at every spectral index. Each curve is the mean after warmup and the band is one standard deviation; the norms in Figure 8 are shown relative to the value at $i = 0$ of the same step. The pattern of Figure 2 holds in every panel: the whitened magnitude stays flat while the raw norm decays, the first-moment scale decays gradually, and the adjacent cosine similarity of $\sqrt{\hat{V}}$ exceeds 0.9. In addition, Figure 10 measures the same cosine similarity between non-adjacent directions, $\cos(\sqrt{\hat{V}_i}, \sqrt{\hat{V}_{i+d}})$ at spectral distance $d \in \{1, 7, 31\}$. At every distance, the similarity exceeds 0.9 for most i.

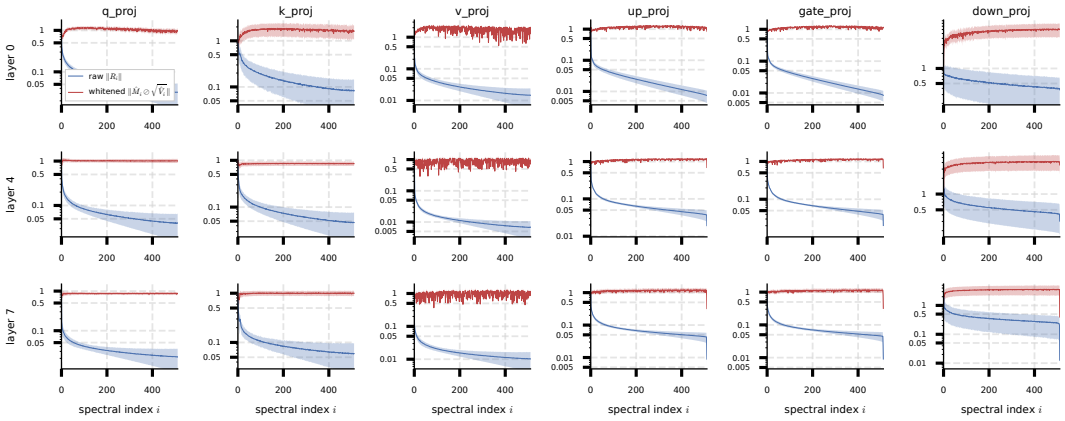


Figure 8: Raw vs. whitened magnitude (Figure 2 (a)) for every module (columns) of layers 0, 4, and 7 (rows)

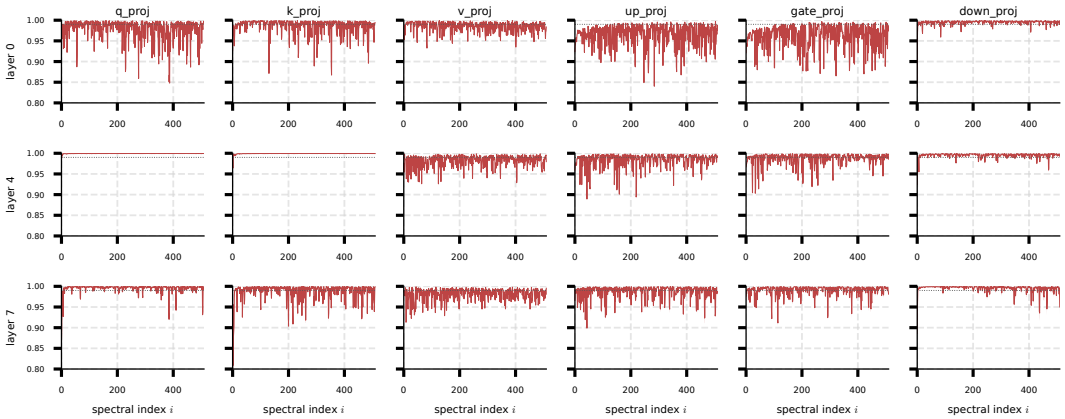


Figure 9: Adjacent similarity of $\sqrt{\hat{V}}$ (Figure 2 (b)) for every module (columns) of layers 0, 4, and 7 (rows). The cosine similarity exceeds 0.9 for almost all i

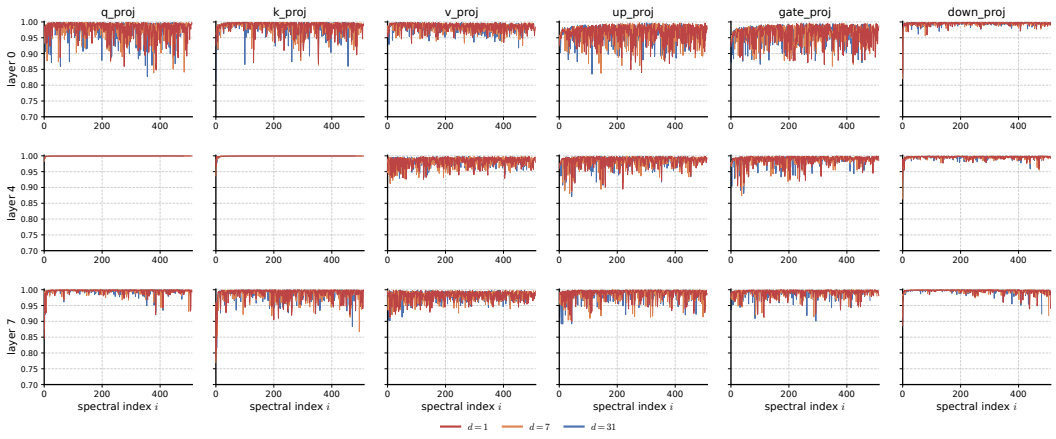


Figure 10: Similarity of $\sqrt{\hat{V}}$ at spectral distance $d \in \{1, 7, 31\}$ for every module (columns) of layers 0, 4, and 7 (rows). The cosine similarity exceeds 0.9 for almost all i at every d.