Two-Sided Low-Rank SOAP for Efficient LLM Training

Fedor Romashov [1] **Nikolay Yudin** [1] **Maxim Rakhuba** [1]

Abstract

Matrix-aware adaptive optimizers such as Shampoo and SOAP exploit row and column structure in neural-network weights, but their optimizer states are expensive for large language models. Recent low-rank optimizers reduce memory by keeping adaptive statistics in a small subspace, yet most existing variants are one-sided: they compress only one matrix axis and therefore approximate only a restricted part of SOAP's two-sided geometry. To combat this issue, we propose **A**daptive **L**ow-Dimensional Subspace **SO**AP Method (**ALSO**), a compact two-sided extension of Alice optimizer that estimates adaptive low-dimensional gradient subspaces and compensates discarded directions. For a weight matrix, ALSO maintains low-dimensional row and column subspaces, applies Adam-like scaling in the resulting core coordinates, and compensates the three residual blocks outside the core. The method can be viewed as a two-sided low-rank approximation to SOAP that keeps full-parameter updates while reducing optimizer-state cost. By conducting pre-training experiments with LLaMA-based models on C4 dataset, we demonstrate that ALSO achieves validation loss that is comparable to SOAP/OSOAP using less memory.

1. Introduction

Adaptive optimizers are central to modern large language model training. Adam optimizer (Kingma & Ba, 2015) remains strong default because it is robust, simple, and cheap to implement. It also has a cheaper alternative, Adafactor (Shazeer & Stern, 2018), which reduces matrix-state memory by factorizing second moments into row and column statistics. However, their limitations are connected with structure: the second moment is considered to be diagonal so row and column correlations are only weakly modeled.

To combat this issue, one can consider a more expressive matrix structure to estimate the curvature of the optimization space. Matrix-aware methods such as K-FAC (Martens & Grosse, 2015) and Shampoo (Gupta et al., 2018) use Kronecker or tensor factorizations of curvature-like statistics; K-FAC is explicitly motivated by Fisher-block approximations, while Shampoo maintains one preconditioner per tensor mode. SOAP (Vyas et al., 2025) further connects Shampoo to Adam-like adaptation by running Adam optimizer in eigenbases of Shampoo's preconditioners. These methods improve update geometry, but their matrix statistics and eigendecompositions are costly for modern LLMs at scale.

To address the issue of heavy optimizer state, one of possible approach is to certainly exploit low-rank gradient structure. LoRA (Hu et al., 2022) restricts adaptation to low-rank parameter updates, while GaLore (Zhao et al., 2024) keeps full parameters but projects gradients into low-rank subspaces before applying an optimizer. LDAdam (Robert et al., 2025) gives a projection-aware update rule for low-dimensional adaptive statistics. FiRA (Chen et al., 2026) and Alice (Gong et al., 2026) (Adaptive low-dimensional subspace estimation) further emphasize that low-rank training should recover or compensate information outside the compressed core. Alice is the closest point of comparison for our work: in the matrix-optimizer setting, its structured-Fisher low-rank extension can be interpreted as a low-rank approximation of one-sided SOAP (OSOAP) with residual compensation.

This interpretation makes the missing piece concrete. Alice compresses only one axis of OSOAP, while full SOAP rotates both row and column coordinates. The Alice paper explicitly lists low-rank counterparts of SOAP among the future directions opened by its structured-Fisher framework (Gong et al., 2026). In this paper, by introducing Adaptive Low-Dimensional Subspace SOAP Method (ALSO), we propose a particular realization of one of these directions. ALSO optimizer preserves the compact-subspace and residual-compensation principle from the one-sided setting of OSOAP and Alice while utilizes SOAP's two-sided geometry. ALSO stores two rank-r bases, for rows and for columns respectively, and applies adaptive scaling inside the $r \times r$ core. The three residual blocks are not discarded: they receive separate Fisher-scale compensation terms. The resulting update is still full-rank in the parameter space,

[1]HSE University. Correspondence to: Fedor Romashov <fvromashov@edu.hse.ru>.

Preprint. June 8, 2026.

but the expensive adaptive state is concentrated in the low-dimensional core plus small residual summaries.

In this sense, ALSO is intended as a practical low-memory alternative to OSOAP: it keeps a two-sided approximation to SOAP's geometry, rather than only one rotated axis, while avoiding the dominant square optimizer states of OSOAP and full SOAP.

Our contributions are as follows:

- We propose ALSO optimizer a two-sided low-rank structured-Fisher approximation to SOAP; its structure is described in Section 3. Additionally in Prop. 3.1, we derive a compensation rule for the three residual blocks outside the biaxial core.
- We evaluate our method on a C4 pre-training evaluation task against Adam, SOAP, OSOAP, and Alice-M[1] optimizers in Section 4.

2. Preliminaries: SOAP, OSOAP, and Alice

We use this section to recover the optimizer geometry through the structured-Fisher viewpoint used to derive Alice. For a matrix parameter $W \in \mathbb{R}^{m\times n}$ and corresponding stochastic gradient $G_t \in \mathbb{R}^{m\times n}$, the empirical Fisher Information Matrix is defined as follows:

$$\mathcal{F}_t = \mathbb{E}[\mathrm{vec}(G_t)\,\mathrm{vec}(G_t)^\top]. \tag{1}$$

Since the full Fisher block is intractable to store or invert, the common pattern in a major part of matrix optimizers is replace it by a structured approximation $\widehat{\mathcal{F}}_t$ and use the corresponding preconditioned first moment. Abstractly, this direction can be written in vectorized form as

$$\mathrm{vec}(N_t) \approx (\widehat{\mathcal{F}}_t^{1/2} + \epsilon I)^{-1}\,\mathrm{vec}(M_t),$$

where M_t is the first-moment estimate. We therefore compare the methods below by the structure they impose on $\widehat{\mathcal{F}}_t$; Table 1 summarizes these approximations in a common notation.

Adam (Kingma & Ba, 2015) optimizer corresponds to a diagonal Fisher approximation in the original parameter coordinates, while Adafactor (Shazeer & Stern, 2018) reduces matrix-state memory by factorizing this diagonal information into row and column statistics. Shampoo (Gupta et al., 2018) takes a different matrix-preconditioning route: for a matrix gradient, its row and column covariance factors are the Fisher marginals

$$L_t = \mathbb{E}[G_t G_t^\top], \qquad R_t = \mathbb{E}[G_t^\top G_t], \tag{2}$$

[1]A modified version of Alice. See the end of Section 2 for the definition of Alice-M.

Table 1. Structured approximations $\widehat{\mathcal{F}}_t$ to the matrix Fisher block $\mathcal{F}_t$.

Method	Fisher approximation
Adam	$\mathrm{Diag}(\mathrm{vec}(D))$
Shampoo	$R_t^{2/\alpha} \otimes L_t^{2/\alpha}$
SOAP	$\widetilde{Q}\ \mathrm{Diag}(\mathrm{vec}(D))\,\widetilde{Q}^\top$
OSOAP	$\widetilde{Q}_L\ \mathrm{Diag}(\mathrm{vec}(D^L))\,\widetilde{Q}_L^\top$
Alice	$\widetilde{U}\ \mathrm{Diag}(\mathrm{vec}(C_A))\,\widetilde{U}^\top + \mathrm{Diag}(p^A) \otimes P_U^\perp$
ALSO	$\widehat{\mathcal{F}}(C,p)$ from Eq. (15)

Here $\widetilde{Q} = Q_R \otimes Q_L$, $\widetilde{Q}_L = I_n \otimes Q_L$, $\widetilde{U} = I_n \otimes U$, and $P_U^\perp = I - UU^\top$.

and preconditions by matrix roots of these factors:

$$W_{t+1} = W_t - \eta L^{-1/\alpha} G_t R^{-1/\alpha}, \quad \alpha = 2, 4. \tag{3}$$

The authors of SOAP (Vyas et al., 2025) observe that the $1/2$-power variant of Shampoo can be interpreted as Adafactor optimizer in the "rotated gradient space". In particular, if we are given eigendecompositions $L_t = Q_L\,\mathrm{Diag}\,(\lambda_L)Q_L^\top$ and $R_t = Q_R\,\mathrm{Diag}\,(\lambda_R)Q_R^\top$, the update of Shampoo optimizer can be rewritten as follows

$$L_t^{-1/\alpha} G_t R_t^{-1/\alpha} = Q_L \left(\frac{Q_L^\top G_t Q_R}{(\lambda_L \lambda_R^\top)^{\odot\alpha}} \right) Q_R^\top, \tag{4}$$

where for matrices $\frac{A}{B}$ denotes elementwise division and $A^{\odot k}$ denotes an elementwise k-th power.

SOAP updates eigenbases of L and R via 1-iteration of the QR algorithm every K training steps, but replaces the factored Adafactor scale by an Adam-style diagonal approximation to the Fisher in the rotated coordinates. Overall, SOAP maintains the first moment M_t in the ambient matrix space and a diagonal Fisher estimate D_t in the rotated coordinates:

$$\begin{aligned} D_t &= \beta_2 D_{t-1} + (1-\beta_2)(Q_L^\top G_t Q_R)^{\odot 2}, \\ N_t^{\mathrm{SOAP}} &= Q_L \left(\frac{Q_L^\top M_t Q_R}{\sqrt{D_t} + \epsilon} \right) Q_R^\top, \end{aligned} \tag{5}$$

As a result, SOAP differs from Adam not by changing the elementwise adaptive rule, but by changing the coordinate system in which the diagonal Fisher approximation is formed. It is *two-sided*: both the row basis Q_L and the column basis Q_R are used.

OSOAP is a memory-reducing variant of SOAP that keeps only one SOAP/Shampoo basis and keeps the other basis constant and equal to the identity matrix. For example, with

only a row basis Q_L, its update is the following:

$$D_t^L = \beta_2 D_{t-1}^L + (1-\beta_2)(Q_L^\top G_t)^{\odot 2},$$
$$N_t^{\mathrm{OSOAP}} = Q_L \left(\frac{Q_L^\top M_t}{\sqrt{D_t^L} + \epsilon} \right). \tag{6}$$

OSOAP removes one square Shampoo statistic R_t and basis Q_R compared with full SOAP. The update still captures correlations across rows, but treats columns coordinatewise. Alice (Gong et al., 2026) applies the low-rank extension principle to this OSOAP update. The idea of Alice optimizer is to replace the whole row basis $Q_L \in \mathbb{R}^{m\times m}$ with $U_t \in \mathbb{R}^{m\times r}$, keep the core Fisher state only for $U_t^\top G_t$, and lift the low-dimensional update back:

$$U_t \left(\frac{U_t^\top M_t}{\sqrt{D_t^A} + \epsilon} \right), \qquad D_t^A \approx \mathbb{E}[(U_t^\top G_t)^{\odot 2}]. \tag{7}$$

The price of this compression can be seen from the decomposition:

$$M_t = U_t U_t^\top M_t + (I - U_t U_t^\top) M_t.$$

The low-rank core update acts on the tracked first-moment component $U_t U_t^\top M_t$. A pure low-rank update would drop the second, residual component $(I - U_t U_t^\top) M_t$. Alice keeps a residual correction through an averaged residual Fisher scale, but in the original algorithm this correction is applied to the current residual gradient rather than to the residual first moment. Schematically, the compensated one-sided update has the form

$$N_t^A \approx U_t \left(\frac{U_t^\top M_t}{\sqrt{D_t^A} + \epsilon} \right) + (I - U_t U_t^\top) G_t D(p_t^A),$$
$$D(p_t^A) = \mathrm{Diag}\left(\left(\sqrt{p_t^A} + \epsilon \right)^{-1} \right), \tag{8}$$

where $(p_t^A)_j$ estimates the averaged second moment of the residual row directions for column j. Thus Alice does not merely add back the residual gradient; it scales it using a Fisher estimate for the untracked subspace. The resulting optimizer is a low-rank approximation of the one-sided Fisher family underlying OSOAP.

In our experiments, we use a modified Alice-style baseline, denoted Alice-M. Unlike the most memory-minimal version of the original Alice algorithm, Alice-M keeps an ambient first moment and uses this moment also in the residual compensation. This choice is intentional: to compare against OSOAP and ALSO, we want a one-sided baseline that preconditions the same first-moment update rather than mixing a core moment update with a residual gradient correction. Alice-M also keeps explicit Shampoo-action statistics for basis updates. This deliberately increases optimizer-state memory, but it substantially improves the quality of the one-sided approximation and makes the comparison with OSOAP and ALSO more direct. Thus Alice-M should be read as a quality-oriented, SOAP-aligned modification of Alice rather than as the original memory-minimal Alice optimizer.

3. ALSO

In a nutshell, ALSO keeps the same Fisher principle but applies it to SOAP's two-sided geometry. Instead of truncating only Q_L, we truncate both Q_L and Q_R to thin bases U_t and V_t. In this case, the moment splits into four orthogonal blocks:

$$M_t = P_{U_t} M_t P_{V_t} + P_{U_t}^\perp M_t P_{V_t} + P_{U_t} M_t P_{V_t}^\perp + P_{U_t}^\perp M_t P_{V_t}^\perp,$$

where P_A and $P_A^\perp$ denote orthogonal projectors onto the images of $\mathrm{Im}(A)$ and $\mathrm{Im}(A)^\perp$ respectively. The first block is the $r \times r$ SOAP-like core, whose diagonal Fisher is estimated in the rotated coordinates. The other three blocks are residual Fisher components outside the biaxial core; in Proposition 3.1 we derive the Frobenius-optimal averaged Fisher scales for them.

In more detail, for a weight matrix $W \in \mathbb{R}^{m\times n}$ and a stochastic gradient $G_t \in \mathbb{R}^{m\times n}$, ALSO replaces SOAP's full row and column basis with two matrices $U_t \in \mathbb{R}^{m\times r}, V_t \in \mathbb{R}^{n\times r}, r \ll \min(m,n)$ with orthonormal columns and forms the low-rank core

$$S_t = U_t^\top G_t V_t \in \mathbb{R}^{r\times r}. \tag{9}$$

An Adam-like second moment C_t is stored only for this core:

$$C_t = \beta_2 C_{t-1} + (1-\beta_2) S_t^{\odot 2}. \tag{10}$$

The first moment M_t is maintained in the ambient matrix space:

$$M_t = \beta_1 M_{t-1} + (1-\beta_1) G_t. \tag{11}$$

The core contribution to the update is the biaxial projection of rows and columns onto r-dimensional subspaces with a subsequent Adam-like division before returning to the original space:

$$N_t^{11} = U_t \left(\frac{U_t^\top M_t V_t}{\sqrt{C_t} + \epsilon} \right) V_t^\top. \tag{12}$$

Using such an update, columns of N_t^{11} belong to $\mathrm{span}(U_t)$, while rows lie inside $\mathrm{span}(V_t)$. Following Alice routine, we should add the information from the remaining subspaces into the update to make it full rank.

Therefore, biaxial low-rank projection explicitly decomposes the ambient first moment as a sum of 4 matrices, whose row and column subspaces are all possible

pairs are $(U_t, V_t), (U_t, V_t^\perp), (U_t^\perp, V_t), (U_t^\perp, V_t^\perp)$ respectively. Defining corresponding orthogonal projectors $P_{U_t} = U_t U_t^\top$, $P_{V_t} = V_t V_t^\top$, and $P_{U_t}^\perp = I - P_{U_t}$, $P_{V_t}^\perp = I - P_{V_t}$, the residual coordinates for the gradient and the momentum are as follows:

$$\begin{aligned} A_t^{01} &= P_{U_t}^\perp M_t V_t, & B_t^{01} &= P_{U_t}^\perp G_t V_t, \\ A_t^{10} &= P_{V_t}^\perp M_t^\top U_t, & B_t^{10} &= P_{V_t}^\perp G_t^\top U_t, \\ A_t^{00} &= P_{U_t}^\perp M_t P_{V_t}^\perp, & B_t^{00} &= P_{U_t}^\perp G_t P_{V_t}^\perp. \end{aligned} \tag{13}$$

Here 01 denotes a row-residual/column-core block, 10 a row-core/column-residual block, and 00 the fully residual block. Instead of storing a full diagonal state for these large blocks, ALSO stores compressed residual second moments:

$$\begin{aligned} p_t^{01} &= \beta_2 p_{t-1}^{01} + \frac{1-\beta_2}{m-r} \mathbf{1}^\top (B_t^{01})^{\odot 2}, \\ p_t^{10} &= \beta_2 p_{t-1}^{10} + \frac{1-\beta_2}{n-r} \mathbf{1}^\top (B_t^{10})^{\odot 2}, \\ p_t^{00} &= \beta_2 p_{t-1}^{00} + \frac{1-\beta_2}{(m-r)(n-r)} \|B_t^{00}\|_F^2. \end{aligned} \tag{14}$$

Here, the mixed blocks use vectors $p_t^{01}, p_t^{10} \in \mathbb{R}^r$ with one scale per retained core direction, while fully residual block uses a single averaged scalar $p_t^{00} \in \mathbb{R}$. Here t denotes the optimization step; when the time index is dropped, i, j index entries of the residual-scale vectors.

In the following proposition we derive formulas for the optimal update of the Adam-like second moment and for p_t^{01}, p_t^{10} and p_t^{00}.

Proposition 3.1 (Optimal compensation)**.** *Fix orthonormal bases U_t, V_t with columns u_i, v_j, and let $\mathcal{F}_t = \mathbb{E}[\mathrm{vec}(G_t)\,\mathrm{vec}(G_t)^\top]$. Let $U_t^\perp, V_t^\perp$ be any orthogonal complements and define $Q_{11} = V_t \otimes U_t$, $Q_{01} = V_t \otimes U_t^\perp$, $Q_{10} = V_t^\perp \otimes U_t$, and $Q_{00} = V_t^\perp \otimes U_t^\perp$. Consider Fisher approximations that are diagonal on the core and isotropic on the three residual blocks:*

$$\begin{aligned} \widehat{\mathcal{F}}(C, p) = {} & Q_{11}\,\mathrm{Diag}(\mathrm{vec}(C)) Q_{11}^\top \\ & + Q_{01}\big(\mathrm{Diag}(p^{01}) \otimes I_{m-r}\big) Q_{01}^\top \\ & + Q_{10}\big(I_{n-r} \otimes \mathrm{Diag}(p^{10})\big) Q_{10}^\top \\ & + p^{00} Q_{00} Q_{00}^\top. \end{aligned} \tag{15}$$

Then the Frobenius projection onto this structured class:

$$(C^\star, p^\star) = \underset{C,p}{\mathrm{argmin}} \|\widehat{\mathcal{F}}(C, p) - \mathcal{F}_t\|_{\mathrm{F}}^2$$

admits the following form

$$\begin{aligned} C_{ij}^\star &= \mathbb{E}(u_i^\top G_t v_j)^2, & p_j^{01,\star} &= \frac{\mathbb{E}\|P_{U_t}^\perp G_t v_j\|_2^2}{m-r}, \\ p_i^{10,\star} &= \frac{\mathbb{E}\|u_i^\top G_t P_{V_t}^\perp\|_2^2}{n-r}, & p^{00,\star} &= \frac{\mathbb{E}\|P_{U_t}^\perp G_t P_{V_t}^\perp\|_F^2}{(m-r)(n-r)}. \end{aligned} \tag{16}$$

Algorithm 1 ALSO for one matrix parameter

Require: Rank r, learning rate η, decay rates β_1, β_2, basis decay ρ, update interval K, compensation weight α_c.
Initialize orthonormal bases $U \in \mathbb{R}^{m\times r}$, $V \in \mathbb{R}^{n\times r}$ from the rank-r SVD of the first gradient.
Initialize $M = 0$, $C = 0$, $Y^U = 0$, $Y^V = 0$, $p^{01} = 0$, $p^{10} = 0$, $p^{00} = 0$.
for training steps $t = 1, 2, \ldots$ **do**
  Receive gradient G_t.
  $M \leftarrow \beta_1 M + (1-\beta_1) G_t$.
  $S \leftarrow U^\top G_t V$.
  $C \leftarrow \beta_2 C + (1-\beta_2) S^{\odot 2}$.
  Apply Adam normalization to $U^\top M V$ using C and lift it back as N^{11} by (12).
  Form (A^{01}, B^{01}), (A^{10}, B^{10}), and (A^{00}, B^{00}) by Eq. (13).
  Update residual scales p^{01}, p^{10}, p^{00} by Eq. (14).
  Compute compensation blocks N^{01}, N^{10}, N^{00} by Eq. (17).
  $N \leftarrow N^{11} + \alpha_c(N^{01} + N^{10} + N^{00})$.
  $W \leftarrow W - \eta N$.
  Update Y^U, Y^V by Eq. (19); every K steps update U, V by QR as in Eq. (20) and transport the basis-dependent state as in Appendix B.
end for

Proof. See Appendix A for the proof. □

The proposition above states that formulas defined in (14) is an EMA estimate of the optimal averaged residual Fisher scales in (16).

Let $D(p) = \mathrm{Diag}((\sqrt{p} + \epsilon)^{-1})$. The residual compensation updates are

$$\begin{aligned} N_t^{01} &= A_t^{01} D(p_t^{01}) V_t^\top, \\ N_t^{10} &= U_t D(p_t^{10}) (A_t^{10})^\top, \\ N_t^{00} &= A_t^{00} / (\sqrt{p_t^{00}} + \epsilon). \end{aligned} \tag{17}$$

The p variables estimate per-coordinate second moments within the corresponding residual subspaces, so each residual component is normalized in the same Adam-like form as the core. During a basis update, motivated by LDAdam's projection-aware state update (Robert et al., 2025), we transport the basis-dependent state: Y^U, Y^V are rotated by the overlaps $U_{t+1}^\top U_t$ and $V_{t+1}^\top V_t$, while second-moment mass is redistributed across $C, p^{01}, p^{10}, p^{00}$. Appendix B gives the transport formulas. Each compensation block can also be passed through a mild norm-growth limiter after basis updates. The full update has the form

$$N_t = N_t^{11} + \alpha_c N_t^{01} + \alpha_c N_t^{10} + \alpha_c N_t^{00}, \tag{18}$$

where α_c controls the strength of the residual compensation. With $\alpha_c = 0$, ALSO reduces to a pure two-sided core optimizer. With nonzero α_c, it remains low-memory but can recover part of the gradient energy that would otherwise be lost by the rank-r truncation. The choice $\alpha_c = 1$ most

Table 2. Dominant optimizer-state memory for one $m \times n$ matrix, excluding parameters and gradients. One-sided variants use a row subspace; a column version swaps m and n.

Method	State size
Adam	$2mn$
SOAP	$2mn + 2m^2 + 2n^2$
OSOAP	$2mn + 2m^2$
Alice	$mr + 2rn + r^2 + n + O(1)$
Alice-M	$mn + 2mr + rn + n + O(1)$
ALSO	$mn + 2mr + 2nr + r^2 + 2r + O(1)$

closely matches the SOAP update, which does not attenuate the residual directions, and is our recommended practical default.

Subspace update from Shampoo statistics. The bases U_t and V_t are meant to track the dominant left and right gradient directions used by SOAP, that is, the leading eigenvectors of the row and column covariance factors L_t, R_t of Eq. (2). SOAP obtains its bases Q_L, Q_R by eigendecomposing L_t and R_t; Shampoo maintains the same factors but preconditions by their matrix roots rather than extracting an explicit eigenbasis. Storing the full L_t and R_t would reintroduce the $m^2 + n^2$ memory cost that ALSO is designed to avoid. We therefore maintain only the action of these factors on the current low-rank bases:

$$\begin{aligned} Y_t^U &= \rho Y_{t-1}^U + (1-\rho) G_t G_t^\top U_t, \\ Y_t^V &= \rho Y_{t-1}^V + (1-\rho) G_t^\top G_t V_t. \end{aligned} \tag{19}$$

Equivalently, $Y_t^U \approx \mathbb{E}[GG^\top U]$ and $Y_t^V \approx \mathbb{E}[G^\top G V]$. A basis update applies one QR subspace-iteration step,

$$U_{t+1} = \mathrm{qf}(Y_t^U), \qquad V_{t+1} = \mathrm{qf}(Y_t^V), \tag{20}$$

where $\mathrm{qf}(\cdot)$ denotes the thin Q factor from QR decomposition. This is a low-rank analogue of SOAP's eigenspace update: the same row and column covariance geometry is queried, but only through its products with r basis vectors.

As in GaLore (Zhao et al., 2024), the bases are initialized from the first gradient: U_1 and V_1 are the top-r left and right singular vectors of the truncated SVD of G_1.

Memory. Table 2 counts the dominant matrix-shaped optimizer states used by each method. SOAP stores a full first moment, a full rotated second moment, and square Shampoo statistics and bases on both matrix axes. OSOAP removes the column-side square terms, but still keeps full-size adaptive moments. Here Alice refers to the memory-minimal original state without a full ambient first moment; Alice-M keeps this moment and explicit thin row-basis/action statistics. Relative to Alice-M, ALSO adds the second thin axis and an $r \times r$ core, so the extra cost is linear in the matrix dimensions plus r^2, with no m^2 or n^2 state when $r \ll m, n$.

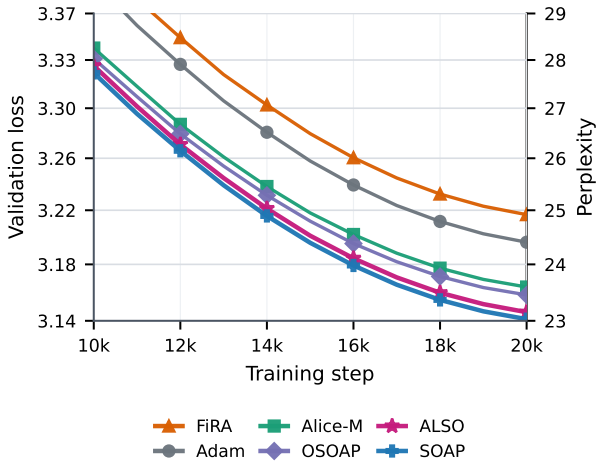

Figure 1. Validation loss for LLaMA-130M on C4, steps 10k–20k. The right axis shows perplexity.

Table 3. Validation perplexity on C4. Lower is better.

Model	Adam	FiRA	Alice-M	OSOAP	SOAP	ALSO
LLaMA-60M	32.31	34.22	31.41	31.14	**30.79**	31.22
LLaMA-130M	24.40	24.92	23.59	23.45	**23.03**	23.15

4. Experiments

We evaluate ALSO by pre-training LLaMA-based causal language models on C4 (Raffel et al., 2020; Touvron et al., 2023). All methods update the full parameter tensors; the low-rank structure affects optimizer states and projected adaptive statistics, not the parameterization of the model.

4.1. C4 Pre-Training

The C4 experiments compare Adam, FiRA, Alice-M, OSOAP, full SOAP, and ALSO. We use sequence length 256 and effective batch size 256. The LLaMA-60M runs use 10k training steps, 1k warmup steps, and rank $r = 128$ for low-rank methods. The LLaMA-130M runs use 20k training steps, 2k warmup steps, and rank $r = 256$. For subspace methods, the basis update interval is $K = 10$. The optimized matrices are the linear layers in attention and MLP blocks; remaining parameters use Adam.

The final perplexities in Table 3 show that ALSO consistently improves over Alice-M and remains close to full SOAP. For the longer LLaMA-130M run, the late-stage validation curves in Figure 1 make this trend clearer: Alice-M follows OSOAP closely, while ALSO tracks full SOAP through the late part of training and separates from Adam and FiRA.

Table 4. Modded-NanoGPT validation loss and matrix optimizer-state memory. The memory column reports FP32 optimizer-state tensors for attention and MLP matrix weights, excluding parameters, gradients, activations, and the tied embedding/head AdamW state shared by all methods. Lower is better.

Method	Val. loss	Opt. state (GB)
Alice-M	3.285	0.58
OSOAP	3.272	2.21
ALSO	3.273	0.65
SOAP	3.256	3.28

4.2. Modded-NanoGPT

We also include a small Modded-NanoGPT comparison, using the older SOAP fork of the benchmark (Jordan et al., 2024; Vyas, 2024). Although the benchmark has since seen many architectural and training-recipe changes, this version remains a useful sanity check for directly transferring SOAP hyperparameters to ALSO; we use rank $r = 256$. Table 4 reports loss and state memory for the 12-layer model.

5. Conclusion

ALSO extends Alice from a one-sided subspace optimizer to a two-sided approximation of SOAP. From the structured-Fisher viewpoint, it keeps SOAP-like row and column geometry in a low-rank core and assigns averaged Fisher-scale compensation to the three residual blocks outside that core. On C4 pre-training, ALSO moves closer to SOAP than Alice-M while avoiding the square optimizer states required by OSOAP and full SOAP.

6. Limitations and Future Work

Our evaluation is limited by the available computational resources: we test LLaMA-style models up to 130M parameters on a single C4 pre-training setup. Larger models, longer training runs, and additional datasets are needed to establish how reliably the observed relationship between Alice, OSOAP, ALSO, and SOAP scales. Future work could also include more detailed ablations over rank, compensation strength, and the basis update interval. In particular, less frequent basis updates could reduce overhead without changing the optimizer state structure. Another direction is to move closer to the original Alice memory target by storing the auxiliary statistics themselves in lower-dimensional form, rather than keeping the more explicit Shampoo-action statistics used in this work.

7. Acknowledgments

The work was supported by the grant for research centers in the field of AI provided by the Ministry of Economic Development of the Russian Federation in accordance with the agreement 000000C313925P4E0002 and the agreement with HSE University № 139-15-2025-009. This research was supported in part through computational resources of HPC facilities at HSE University (Kostenetskiy et al., 2021).

References

Chen, X., Feng, K., Li, C., Lai, X., Yue, X., Yuan, Y., and Wang, G. Fira: Can we achieve full-rank training of llms under low-rank constraint? *Advances in Neural Information Processing Systems*, 38:120680–120712, 2026.

Gong, W., Scetbon, M., Ma, C., and Meeds, E. Towards efficient optimizer design for LLM via structured Fisher approximation with a low-rank extension. In *International Conference on Learning Representations*, 2026. arXiv:2502.07752.

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

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Jordan, K., Bernstein, J., Rappazzo, B., @fernbear.bsky.social, Vlado, B., Jiacheng, Y., Cesista, F., Koszarsky, B., and @Grad62304977. modded-nanogpt: Speedrunning the nanogpt baseline, 2024. URL https://github.com/KellerJordan/modded-nanogpt.

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

Kostenetskiy, P., Chulkevich, R., and Kozyrev, V. Hpc resources of the higher school of economics. In *Journal of Physics: Conference Series*, volume 1740, pp. 012050. IOP Publishing, 2021.

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

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Robert, T., Safaryan, M., Modoranu, I.-V., and Alistarh, D. LDAdam: Adaptive optimization from low-dimensional

gradient statistics. In *International Conference on Learning Representations*, volume 2025, pp. 4778–4814, 2025.

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

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Vyas, N. Modded-NanoGPT: SOAP optimizer experiments, 2024. URL https://github.com/nikhilvyas/modded-nanogpt-SOAP.

Vyas, N., Morwani, D., Zhao, R., Shapira, I., Brandfonbrener, D., Janson, L., and Kakade, S. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *International Conference on Learning Representations*, volume 2025, pp. 93423–93444, 2025.

Zhao, J., Zhang, Z., Chen, B., Wang, Z., Anandkumar, A., and Tian, Y. GaLore: Memory-efficient LLM training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

A. Proof of Proposition 3.1

Proof. In this proof we use the column-stacking convention for vec, so that

$$\mathrm{vec}(ab^\top) = b \otimes a, \qquad (b \otimes a)^\top \mathrm{vec}(G) = a^\top G b. \tag{21}$$

Fix the bases and denote $U_t, V_t, G_t, \mathcal{F}_t$ as U, V, G and $\mathcal{F}$ for simplicity. Let

$$Q = [\, Q_{11} \;\; Q_{01} \;\; Q_{10} \;\; Q_{00} \,].$$

This matrix is a column permutation of $[V, V^\perp] \otimes [U, U^\perp]$, hence it is orthogonal. Therefore, setting $\Phi = Q^\top \mathcal{F} Q$, orthogonal invariance gives

$$\|\widehat{\mathcal{F}}(C,p) - \mathcal{F}\|_{\mathrm{F}}^2 = \|Q^\top \widehat{\mathcal{F}}(C,p) Q - \Phi\|_{\mathrm{F}}^2. \tag{22}$$

In this basis,

$$Q^\top \widehat{\mathcal{F}}(C,p) Q = \mathrm{blkdiag}\big(\mathrm{Diag}(\mathrm{vec}(C)), \mathrm{Diag}(p^{01}) \otimes I_{m-r}, I_{n-r} \otimes \mathrm{Diag}(p^{10}), p^{00} I_{(m-r)(n-r)}\big)\,,$$

where $\mathrm{blkdiag}(X_1, \ldots, X_k)$ is block-diagonal matrix whose blocks are $X_1, \ldots X_k$. It can be seen that the off-diagonal blocks of Φ contribute only a constant to the Frobenius norm. Thus, writing $\Phi^{ab} = Q_{ab}^\top \mathcal{F} Q_{ab}$ for $ab \in \{11, 01, 10, 00\}$, we may, up to the additive constant, rewrite the square Frobenius norm as follows:

$$\begin{aligned}\|\widehat{\mathcal{F}}(C,p) - \mathcal{F}\|_{\mathrm{F}}^2 = {} & \|\,\mathrm{Diag}(\mathrm{vec}(C)) - \Phi^{11}\|_{\mathrm{F}}^2 + \|\,\mathrm{Diag}(p^{01}) \otimes I_{m-r} - \Phi^{01}\|_{\mathrm{F}}^2 \\ & + \|I_{n-r} \otimes \mathrm{Diag}(p^{10}) - \Phi^{10}\|_{\mathrm{F}}^2 + \|p^{00} I - \Phi^{00}\|_{\mathrm{F}}^2 + \mathrm{const}.\end{aligned} \tag{23}$$

The four terms depend on disjoint variables, so the minimization separates.

For the core block, the best diagonal Frobenius approximation keeps the diagonal of Φ^{11}. Indexing the columns of $Q_{11} = V \otimes U$ by a pair of indices (i, j), we obtain

$$C_{ij}^\star = (v_j \otimes u_i)^\top \mathcal{F} (v_j \otimes u_i) = \mathbb{E}(u_i^\top G v_j)^2, \tag{24}$$

where the last equality uses Eq. (21).

As $\mathrm{Diag}(p^{01}) \otimes I_{m-r}$ is diagonal and its diagonal can be divided into r blocks with equal diagonal value, we may recursively represent the second term: if Φ_{jj}^{01} is the j-th $(m-r) \times (m-r)$ diagonal submatrix of Φ^{01} corresponding to the coordinates $U_\perp^\top G v_j$, then

$$\sum_{j=1}^{r} \|\Phi_{jj}^{01} - p_j^{01} I_{m-r}\|_{\mathrm{F}}^2.$$

For each j, minimization of the functional above is equivalent to the minimization of the following quadratic functional:

$$\|\Phi_{jj}^{01}\|_{\mathrm{F}}^2 - 2 p_j^{01}\, \mathrm{tr}(\Phi_{jj}^{01}) + (m-r)(p_j^{01})^2,$$

Finally, for 01 block we obtain $p_j^{01,\star} = \mathrm{tr}(\Phi_{jj}^{01})/(m-r)$. Since Φ_{jj}^{01} is the covariance of $(U^\perp)^\top G v_j$, we have

$$\begin{aligned} p_j^{01,\star} &= \frac{1}{m-r} \sum_{k=1}^{m-r} (v_j \otimes u_{\perp,k})^\top \mathcal{F}_t (v_j \otimes u_{\perp,k}) \\ &= \frac{1}{m-r} \mathbb{E} \sum_{k=1}^{m-r} (u_{\perp,k}^\top G v_j)^2 = \frac{\mathbb{E}\|(U^\perp)^\top G v_j\|_2^2}{m-r} = \frac{\mathbb{E}\|P_U^\perp G v_j\|_2^2}{m-r}. \end{aligned} \tag{25}$$

Similarly, let Φ_{ii}^{10} be the principal submatrix corresponding to the coordinates $u_i^\top G V^\perp$. The terms depending on p^{10} are, again up to a constant,

$$\sum_{i=1}^{r} \|\Phi_{ii}^{10} - p_i^{10} I_{n-r}\|_{\mathrm{F}}^2,$$

so $p_i^{10,\star} = \mathrm{tr}(\Phi_{ii}^{10})/(n-r)$. Hence

$$\begin{aligned} p_i^{10,\star} &= \frac{1}{n-r}\sum_{\ell=1}^{n-r}(v_{\perp,\ell}\otimes u_i)^\top \mathcal{F}(v_{\perp,\ell}\otimes u_i) \\ &= \frac{1}{n-r}\mathbb{E}\sum_{\ell=1}^{n-r}(u_i^\top G v_{\perp,\ell})^2 = \frac{\mathbb{E}\|u_i^\top G V_\perp\|_2^2}{n-r} = \frac{\mathbb{E}\|u_i^\top G P_V^\perp\|_2^2}{n-r}. \end{aligned} \tag{26}$$

Finally, for the fully residual block, set $d = (m-r)(n-r)$. The scalar quadratic

$$\|\Phi^{00} - p^{00} I_d\|_{\mathrm{F}}^2 = \|\Phi^{00}\|_{\mathrm{F}}^2 - 2p^{00}\,\mathrm{tr}(\Phi^{00}) + d(p^{00})^2$$

has minimizer $p^{00,\star} = \mathrm{tr}(\Phi^{00})/d$. Moreover,

$$\begin{aligned} \mathrm{tr}(\Phi^{00}) &= \sum_{k=1}^{m-r}\sum_{\ell=1}^{n-r}(v_{\perp,\ell}\otimes u_{\perp,k})^\top \mathcal{F}(v_{\perp,\ell}\otimes u_{\perp,k}) \\ &= \mathbb{E}\sum_{k=1}^{m-r}\sum_{\ell=1}^{n-r}(u_{\perp,k}^\top G v_{\perp,\ell})^2 = \mathbb{E}\|U_\perp^\top G V_\perp\|_{\mathrm{F}}^2 = \mathbb{E}\|P_U^\perp G P_V^\perp\|_{\mathrm{F}}^2. \end{aligned} \tag{27}$$

Combining (24)–(27) yields (16). The expressions are nonnegative and depend on $U^\perp, V^\perp$ only through the projectors $P_U^\perp, P_V^\perp$. □

B. Projection-Aware Transport

When bases are updated, the coordinates used by the second-moment state change. Resetting $C, p^{01}, p^{10}, p^{00}$ would discard accumulated curvature information, while keeping them unchanged would attach the old scales to the wrong subspaces. We therefore transport the state by interpreting it as the structured Fisher approximation in (15), then projecting the same matrix onto the new bases. Although the motivation is similar to LDAdam's projection-aware state update, we write the transport directly in terms of the structured Fisher approximation. This gives a Fisher-matrix interpretation to the redistribution of curvature mass among the core and residual blocks.

We use this transport in our experiments because it makes basis changes smoother and gives a modest empirical improvement. At the same time, it should be viewed as a practical state-continuity refinement rather than a defining part of the optimizer: ALSO can also be implemented by keeping the basis-dependent second-moment state unchanged after a basis update, with the same memory structure and a slightly less aligned approximation.

Let U, V be the old bases and let U^+, V^+ be the new bases. Define

$$P_U = UU^\top, \quad P_U^+ = U^+(U^+)^\top, \qquad P_V = VV^\top, \quad P_V^+ = V^+(V^+)^\top.$$

Let $P_U^\perp = I - P_U$ and $P_V^\perp = I - P_V$. Define the overlap matrices

$$S_U = (U^+)^\top U, \qquad S_V = (V^+)^\top V, \qquad A_U = S_U^{\odot 2}, \qquad A_V = S_V^{\odot 2}.$$

Let

$$\begin{aligned} d_U^+ &= \mathrm{diag}(I_r - S_U S_U^\top), \quad d_U^- = \mathrm{diag}(I_r - S_U^\top S_U), \\ d_V^+ &= \mathrm{diag}(I_r - S_V S_V^\top), \quad d_V^- = \mathrm{diag}(I_r - S_V^\top S_V) \end{aligned}$$

denote the diagonal residuals. They have the concrete interpretations

$$(d_U^+)_i = \|(I - P_U)u_i^+\|_2^2, \qquad (d_U^-)_i = \|(I - P_U^+)u_i\|_2^2,$$

and analogously for V. Thus d^+ measures old-residual mass seen by a new core direction, while d^- measures old-core mass that has moved into the new residual complement. The residual-residual overlap traces are

$$\gamma_U = \mathrm{tr}\big((I - P_U^+)(I - P_U)\big) = m - 2r + \|S_U\|_{\mathrm{F}}^2,$$

$$\gamma_V = \operatorname{tr}\big((I - P_V^+)(I - P_V)\big) = n - 2r + \|S_V\|_{\mathrm{F}}^2.$$

Let $\widehat{\mathcal{F}}_{\mathrm{old}}$ denote the structured Fisher matrix represented by the old state. For any row direction $a \in \mathbb{R}^m$ and column direction $b \in \mathbb{R}^n$, (15) gives

$$(b \otimes a)^\top \widehat{\mathcal{F}}_{\mathrm{old}}(b \otimes a) = \sum_{\alpha=1}^{r}\sum_{\beta=1}^{r} C_{\alpha\beta}(u_\alpha^\top a)^2 (v_\beta^\top b)^2 + \|P_U^\perp a\|_2^2 \sum_{\beta=1}^{r} p_\beta^{01}(v_\beta^\top b)^2 + \|P_V^\perp b\|_2^2 \sum_{\alpha=1}^{r} p_\alpha^{10}(u_\alpha^\top a)^2 + p^{00}\|P_U^\perp a\|_2^2 \|P_V^\perp b\|_2^2. \tag{28}$$

The transported state is obtained by applying the projection formulas from Proposition 3.1 to $\widehat{\mathcal{F}}_{\mathrm{old}}$ and the new bases U^+, V^+.

For the new core entry (i, j), take $a = u_i^+$ and $b = v_j^+$ in (28). Since $(u_\alpha^\top u_i^+)^2 = (A_U)_{i\alpha}$, $(v_\beta^\top v_j^+)^2 = (A_V)_{j\beta}$, $\|P_U^\perp u_i^+\|_2^2 = (d_U^+)_i$, and $\|P_V^\perp v_j^+\|_2^2 = (d_V^+)_j$, we get

$$C_{ij}^+ = \sum_{\alpha,\beta}(A_U)_{i\alpha} C_{\alpha\beta}(A_V)_{j\beta} + (d_U^+)_i \sum_\beta p_\beta^{01}(A_V)_{j\beta} + (d_V^+)_j \sum_\alpha p_\alpha^{10}(A_U)_{i\alpha} + p^{00}(d_U^+)_i (d_V^+)_j. \tag{29}$$

In matrix form it can be rewritten as follows:

$$C^+ = A_U C A_V^\top + d_U^+ (A_V p^{01})^\top + (A_U p^{10})(d_V^+)^\top + p^{00} d_U^+ (d_V^+)^\top. \tag{30}$$

For the 01 residual scale, Proposition 3.1 gives

$$(p_j^{01})^+ = \frac{1}{m-r}\sum_{k=1}^{m-r}((v_j^+ \otimes u_{\perp,k}^+)^\top \widehat{\mathcal{F}}_{\mathrm{old}}(v_j^+ \otimes u_{\perp,k}^+)),$$

where $\{u_{\perp,k}^+\}$ is any orthonormal basis of the new row residual space. Summing (28) over these residual directions uses the identities

$$\sum_k (u_\alpha^\top u_{\perp,k}^+)^2 = (d_U^-)_\alpha, \qquad \sum_k \|P_U^\perp u_{\perp,k}^+\|_2^2 = \gamma_U.$$

Therefore

$$(m-r)(p_j^{01})^+ = \sum_\beta (A_V)_{j\beta} \sum_\alpha C_{\alpha\beta}(d_U^-)_\alpha + \gamma_U \sum_\beta (A_V)_{j\beta} p_\beta^{01} + (d_V^+)_j \sum_\alpha (d_U^-)_\alpha p_\alpha^{10} + \gamma_U p^{00}(d_V^+)_j. \tag{31}$$

Collecting the j coordinates gives

$$(p^{01})^+ = \frac{A_V(C^\top d_U^-) + \gamma_U A_V p^{01} + d_V^+\big((d_U^-)^\top p^{10}\big) + \gamma_U p^{00} d_V^+}{m-r}. \tag{32}$$

The 10 block is symmetric. Summing over an orthonormal basis $\{v_{\perp,\ell}^+\}$ of the new column residual space gives

$$\sum_\ell (v_\beta^\top v_{\perp,\ell}^+)^2 = (d_V^-)_\beta, \qquad \sum_\ell \|P_V^\perp v_{\perp,\ell}^+\|_2^2 = \gamma_V.$$

Hence

$$(n-r)(p_i^{10})^+ = \sum_\alpha (A_U)_{i\alpha} \sum_\beta C_{\alpha\beta}(d_V^-)_\beta + \gamma_V \sum_\alpha (A_U)_{i\alpha} p_\alpha^{10} + (d_U^+)_i \sum_\beta (d_V^-)_\beta p_\beta^{01} + \gamma_V p^{00}(d_U^+)_i, \tag{33}$$

or, in vector form,

$$(p^{10})^+ = \frac{A_U(C d_V^-) + \gamma_V A_U p^{10} + d_U^+\big((d_V^-)^\top p^{01}\big) + \gamma_V p^{00} d_U^+}{n-r}. \tag{34}$$

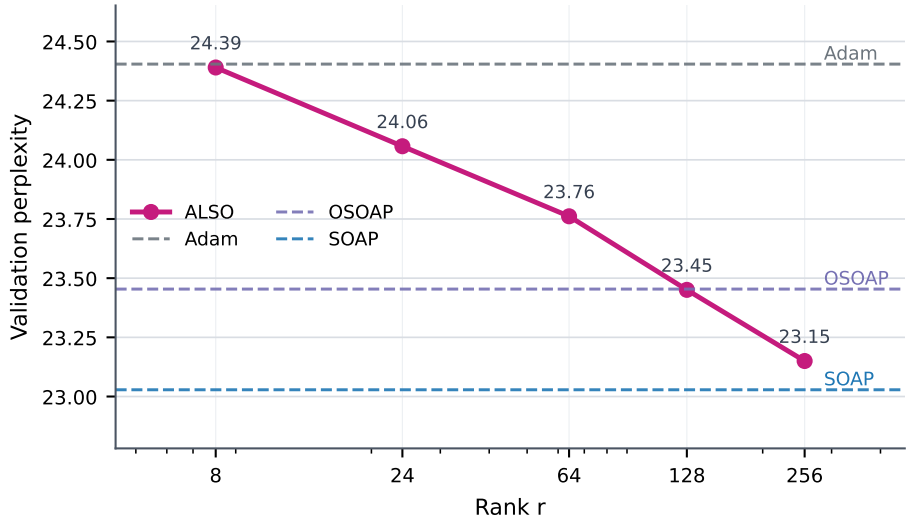

Figure 2. Rank ablation for ALSO on LLaMA-130M pre-training on C4. Dashed lines show Adam, OSOAP, and full SOAP baselines from the same experimental setup. Lower validation perplexity is better.

Finally, the 00 residual scale averages the old quadratic form over both new residual complements:

$$(p^{00})^+ = \frac{1}{(m-r)(n-r)} \sum_{k,\ell} (v^+_{\perp,\ell} \otimes u^+_{\perp,k})^\top \widehat{\mathcal{F}}_{\text{old}} (v^+_{\perp,\ell} \otimes u^+_{\perp,k}).$$

Using the same summed identities on both axes yields

$$(p^{00})^+ = \frac{(d_U^-)^\top C d_V^- + \gamma_U (d_V^-)^\top p^{01} + \gamma_V (d_U^-)^\top p^{10} + \gamma_U \gamma_V p^{00}}{(m-r)(n-r)}. \tag{35}$$

Thus (30), (32), (34), and (35) preserve the interpretation of $C, p^{01}, p^{10}, p^{00}$ as the Frobenius projection of a single structured Fisher estimate onto the current bases. The first moment M_t is stored in ambient coordinates, so it does not require an analogous transport; the action statistics can be rotated as $Y^{U,+} = Y^U S_U^\top$ and $Y^{V,+} = Y^V S_V^\top$.

C. Rank Ablation

We additionally evaluate the sensitivity of ALSO to the rank r of the two-sided core on the LLaMA-130M C4 setup. In this sweep, only r is varied; the training setup, basis update interval, and other optimizer hyperparameters are kept fixed. Figure 2 shows that increasing the rank monotonically lowers the final validation perplexity in this setting, moving the method from the Adam baseline toward full SOAP.