HIERARCHICAL MULTI-SUBSPACE LOW-RANK COMPRESSION FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-rank model compression represents a weight matrix in a global low-rank subspace, which can limit its ability to preserve distinct local low-rank structures existing in the pretrained weight. We propose Hierarchical Multi-subspace low-rank Compression (HMC), a post-training method that partition the weight rows into different groups, where each group can separately approximate their local subspace using low-rank factorization. We further construct a hierarchical representation by repartitioning and approximating successive residuals to capture local structure beyond a single partition. To optimize the local factor matrices, HMC adaptively allocates ranks and jointly fits the local factors across different groups. Under subspace and partition assumptions, our analysis establishes when local factors reduce reconstruction error relative to a global factorization and when two levels reduce the factor cost of exact reconstruction relative to a single level. Experimental results across five LLMs, HMC generally lowers WikiText-2 and C4 perplexity and improves accuracy on six down-stream tasks over the low-rank compression baselines. Controlled ablations further support the benefits of multi-subspace partition, joint fitting, and hierarchical representation.

1 INTRODUCTION

The rapid growth of large language models (LLMs) imposes prohibitive deployment costs in both storage and computation, particularly on resource-limited edge devices (Alizadeh et al., 2024; Liu et al., 2024). Post-training compression reduces these costs through weight quantization (Zhou et al., 2024; Frantar et al., 2023; Lin et al., 2024), pruning (Guo et al., 2025; Frantar & Alistarh, 2023), and low-rank approximation (Hsu et al., 2022; Yuan et al., 2023; Wang et al., 2025c;a; Chen et al., 2026; Qi et al., 2026). Among these, low-rank factorization approaches approximate a dense weight matrix with two smaller factors, but the resulting errors in layer outputs can propagate through subsequent Transformer layers and degrade end-to-end model performance.

A prominent line of work addresses this challenge by incorporating calibration data into vanilla low-rank approximation to preserve model's output behavior. For example, weighted low-rank factorization uses Fisher information to weight reconstruction errors (Hsu et al., 2022), while activation-aware singular-value decomposition (SVD) rescales weight channels (Yuan et al., 2023). Building on this principle, Wang et al. (2025c) aligns singular-value truncation with an activation-aware objective through whitening. More recently, differentiable SVD formulations optimize truncation choices together with the factorization (Wang et al., 2025a), and Qi et al. (2026) derives an optimal fixed-rank solution for a layer-wise activation-aware objective. These methods improve reconstruction accuracy and maintain outputs's behavior through data-aware fitting, and achieve strong language modeling and downstream task performance.

However, directly applying low-rank factorization to the original or whitened weight matrix may struggle to capture local low-rank structures. As illustrated in Figure 1, the local similarities among pretrained weight rows are difficult to discern in their original order, yet reordering their row indices uncovers clear cluster structures. This pattern suggests that the weights consist of multiple *distinct* local subspaces, where each group of rows shares its own low-dimensional structure. Consequently, a global factorization forces these distinct groups into a single shared subspace, inevitably leads to poor approximation of these local structures. This motivates our central question: *how can we identify and exploit the multi-subspace structure of pretrained weights for LLM compression?*

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.

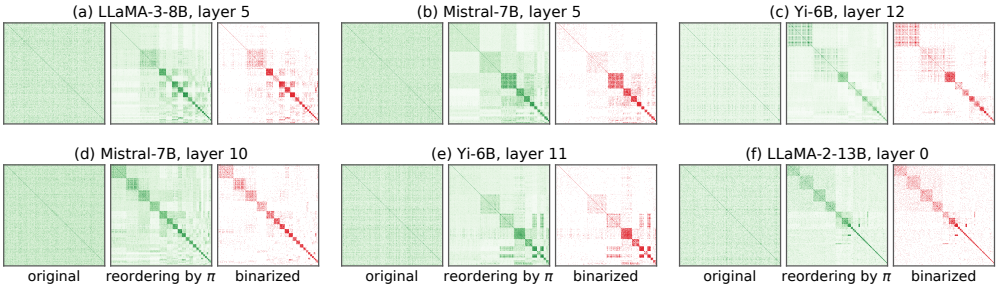


Figure 1: Empirical evidence of the Multi-subspace structure of weight matrices under row partitioning in pretrained LLMs. Each cell shows the cosine similarity $|\cos(\mathbf{w}_i, \mathbf{w}_j)|$ between row vectors of the down-projection weight matrix across six layers, shown from left to right in the original row order, after reordering by π via k-subspaces partitioning group number $K{=}64$, and finally binarized to highlight the top 5% highest similarities. The corresponding Hierarchical Multi-subspace evidence is given in Figure 8.

We address this with **H**ierarchical **M**ulti-subspace low-rank **C**ompression (**HMC**), which partitions weight rows into groups to preserve distinct local subspaces. To optimize this representation, HMC adaptively allocates ranks and jointly fits local factors across groups to account for activation correlations. However, a single partition is often insufficient; the resulting approximation residuals frequently contain latent low-rank structures under different group assignments (Figure 8). Thus, we construct a hierarchical representation that recursively repartitions and approximates successive residuals, capturing latent low-rank structures a single level would miss. Our contributions are:

1. We propose a multi-subspace low-rank factorization for LLM compression that identifies local low-rank structure in pretrained weight matrices and adaptively allocates ranks and jointly fits factors across subspaces. We show that it theoretically achieves lower approximation error than global low-rank factorization at the same factor budget in an orthogonal subspace model.
2. We develop a hierarchical representation that captures additional local low-rank structure by regrouping and approximating successive residuals. Our analysis establishes that this two-level approach reduces the factor cost of exact reconstruction relative to a single-level partition, effectively extracting latent patterns that a single level fails to resolve.
3. Experiments on 5 decoder-only LLMs show that HMC generally achieves lower perplexity and higher zero-shot accuracy than the evaluated low-rank compression baselines. Controlled ablations further demonstrate the benefits of subspace partitioning, activation-aware joint fitting, and hierarchical representation.

Section 3 defines the multi-subspace representation, its activation-aware fitting objective, and the hierarchical representation. Section 4 evaluates the method and its components. Appendix A reviews related work; Appendix B documents the configurations, complete results, and additional analyses; and Appendices C and D provide the theoretical analysis and complete proofs.

2 Preliminaries

Let $\mathbf{W} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ be the weight matrix, and $\hat{\mathbf{W}}$ be a compression of it. To measure how faithfully $\hat{\mathbf{W}}$ preserves the input-output mapping (Wang et al., 2025c; Yuan et al., 2023), we adopt its empirical output reconstruction error on a calibration set $\mathbf{X} = [\mathbf{x}_1^\top; \mathbf{x}_2^\top; \ldots; \mathbf{x}_N^\top] \in \mathbb{R}^{N \times d_{\text{in}}}$ as

$$J_0(\hat{\mathbf{W}}) := \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{x}_i^\top (\mathbf{W} - \hat{\mathbf{W}}) \right\|_2^2 = \operatorname{tr}\left[(\mathbf{W} - \hat{\mathbf{W}})^\top \mathbf{C}_0 (\mathbf{W} - \hat{\mathbf{W}}) \right], \tag{1}$$

where $\mathbf{C}_0 = \mathbf{X}^\top \mathbf{X}/N$ is the empirical second-moment matrix of the input $\mathbf{x}$. In case $\mathbf{C}_0$ may be singular, we safely regularize it with a fixed parameter $\gamma > 0$ to have

$$\mathbf{C} := \mathbf{C}_0 + \gamma \mathbf{I} \succ 0, \qquad J(\hat{\mathbf{W}}) := J_0(\hat{\mathbf{W}}) + \gamma \|\mathbf{W} - \hat{\mathbf{W}}\|_{\text{F}}^2. \tag{2}$$

At a retention ratio ρ, we have a factor budget $\mathcal{B} = \rho\, d_{\text{in}} d_{\text{out}}$ to compress the current layer, whereas a standard global rank-r factorization consumes $r(d_{\text{in}} + d_{\text{out}})$ parameters.

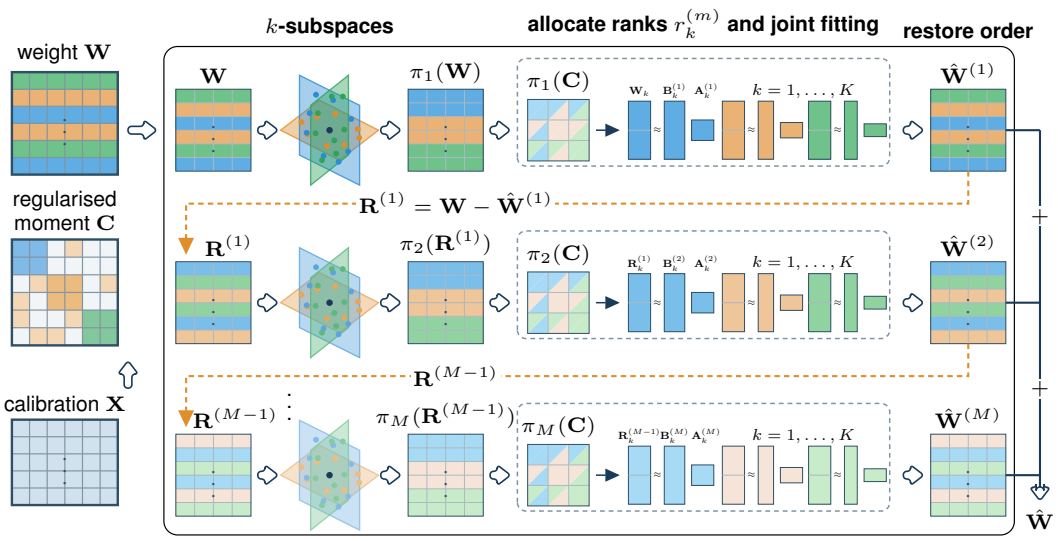


Figure 2: Overview of Hirarchical Multi-subspace low-rank Compression (HMC). Calibration activations define the regularized second-moment matrix $\mathbf{C}$. At level m, k-subspaces partitions the rows of the current construction residual $\mathbf{R}^{(m-1)}$. A level-specific budget adaptively allocates the matrix ranks and joint activation-aware updates then fit all local factors while retaining cross-cluster interactions.

3 Multi-Subspace Low-Rank Compression

Standard low-rank factorization represents all weight rows using a single shared subspace, which may limit its ability to capture local low-rank structure. We first introduce a multi-subspace representation in §3.1, then describe activation-aware joint fitting for a single level in §3.2. We extend this construction to successive residuals in §3.3 and present adaptive rank allocation under each level's budget in §3.4. Figure 2 summarizes the overall framework.

3.1 Multi-Subspace Representation

To construct local low-rank approximations, we group the rows of $\mathbf{W}$ according to their subspace structure. Let $\pi = \{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$ denote a partition of the d_{in} row indices, with each group defining a block $\mathbf{W}_k = \mathbf{W}_{\mathcal{I}_k,:}$ with $n_k = |\mathcal{I}_k|$ rows. We obtain π using k-subspaces clustering (Wang et al., 2022), the clustering procedure is detailed in Appendix B.1.

To assemble the local blocks into a full weight matrix, we define the row-assembly operator Merge_π, which places each block back in its original row positions:

$$\mathbf{W} = \text{Merge}_\pi\big(\{\mathbf{W}_k\}_{k=1}^K\big), \quad \mathbf{W}_{\mathcal{I}_k,:} = \mathbf{W}_k \text{ for every } k.$$

We then approximate each block $\mathbf{W}_k$ with a local rank-r_k factorization:

$$\hat{\mathbf{W}}_{\mathcal{I}_k,:} = \hat{\mathbf{W}}_k = \mathbf{B}_k\mathbf{A}_k, \qquad \mathbf{B}_k \in \mathbb{R}^{n_k \times r_k}, \quad \mathbf{A}_k \in \mathbb{R}^{r_k \times d_{\text{out}}}. \tag{3}$$

Consequently, the approximated rows in $\mathcal{I}_k$ lie in the row space of $\mathbf{A}_k$, allowing each group to use its own subspace with a group-specific rank.

In Proposition 2, we show that under an orthogonal multi-subspace model, this local formulation achieves strictly lower reconstruction error than any global low-rank factorization at the same factor budget. We present the detailed analysis and proofs in Appendix C

3.2 Single-Level Multi-Subspace Compression

Given the local representation, we now fit its factor matrices under the activation-aware objective J in Eq. (2), with the partition π and ranks $\{r_k\}_{k=1}^K$ fixed. To express this objective in terms of local

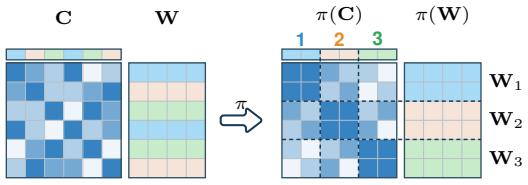


Figure 3: Applying row partition to the weight matrix. Left: $\mathbf{W}$ and $\mathbf{C}$ in the original input-channel order. Right: π groups the rows of $\mathbf{W}$ and applies the same permutation to both axes of $\mathbf{C}$, yielding blocks $\mathbf{W}_k$ and $\mathbf{C}_{kl}$.

blocks, we group the rows of $\mathbf{W}$ and both axes of $\mathbf{C}$ according to the same input-channel partition π (Figure 3). Writing $\pi(\cdot)$ for the corresponding reordering, we obtain

$$\pi(\mathbf{W}) = \begin{bmatrix}\mathbf{W}_1^\top & \cdots & \mathbf{W}_K^\top\end{bmatrix}^\top, \qquad \pi(\mathbf{C}) = [\mathbf{C}_{kl}]_{k,l=1}^K, \qquad \mathbf{C}_{kl} = \mathbf{C}_{\mathcal{I}_k,\mathcal{I}_l}, \tag{4}$$

where $\mathbf{C}_{kl} \in \mathbb{R}^{n_k \times n_l}$ and $\mathbf{C}_{lk} = \mathbf{C}_{kl}^\top$ by symmetry. With residual blocks $\mathbf{D}_k = \mathbf{W}_k - \hat{\mathbf{W}}_k$, the objective decomposes as

$$J(\hat{\mathbf{W}}) = \sum_{k=1}^{K} \operatorname{tr}(\mathbf{D}_k^\top \mathbf{C}_{kk} \mathbf{D}_k) + \sum_{1 \le k,l \le K, k \ne l} \operatorname{tr}(\mathbf{D}_k^\top \mathbf{C}_{kl} \mathbf{D}_l). \tag{5}$$

The first sum measures within-cluster reconstruction errors, while the second captures cross-cluster interactions that independent fitting neglects. To account for both, we minimize the full objective under the local rank constraints in Eq. (3), updating one weight block at a time with the others fixed.

For block k, we collect the contributions from the other residual blocks in $\mathbf{G}_k$ and use the Cholesky factor $\mathbf{L}_k$ of $\mathbf{C}_{kk}$ (Golub & Van Loan, 2013; Wang et al., 2025c) to form the adjusted target $\mathbf{T}_k$:

$$\mathbf{G}_k = \sum_{l \ne k} \mathbf{C}_{kl} \mathbf{D}_l, \quad \mathbf{C}_{kk} = \mathbf{L}_k \mathbf{L}_k^\top, \quad \mathbf{T}_k = \mathbf{L}_k^\top \mathbf{W}_k + \mathbf{L}_k^{-1} \mathbf{G}_k, \tag{6}$$

The correction term $\mathbf{L}_k^{-1}\mathbf{G}_k$ incorporates cross-cluster interactions into the local target $\mathbf{T}_k$. With the other blocks fixed, the optimal update is obtained from the best rank-r_k approximation of $\mathbf{T}_k$, as stated below.

Proposition 1 (Exact joint block update). *Fix $\mathbf{C} \succ 0$, a partition π, all blocks $\{\hat{\mathbf{W}}_l : l \ne k\}$, and an integer $0 \le r_k \le \min(n_k, d_{\text{out}})$. A global minimiser of J over $\operatorname{rank}(\hat{\mathbf{W}}_k) \le r_k$ is*

$$\hat{\mathbf{W}}_k^\star = \mathbf{L}_k^{-\top} \mathcal{T}_{r_k}(\mathbf{T}_k), \tag{7}$$

where $\mathcal{T}_{r_k}(\cdot)$ is the r_k truncated SVD.

For the SVD $\mathbf{T}_k = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, the stored factors are $\mathbf{B}_k = \mathbf{L}_k^{-\top}\mathbf{U}_{:r_k}\mathbf{\Sigma}_{:r_k}$ and $\mathbf{A}_k = \mathbf{V}_{:r_k}^\top$. Sequential exact updates monotonically decrease J. When $K = 1$, this reduces to the global whitening solution of SVD-LLM (Wang et al., 2025c). Appendix D.1 gives the detailed derivation and proofs for Proposition 1.

3.3 Hierarchical Multi-subspace Compression

A single-level multi-subspace approximation captures local low-rank structure within each group, but its residual $\mathbf{R}^{(1)} = \mathbf{W} - \hat{\mathbf{W}}^{(1)}$ may retain structure under a different grouping. As shown in Figure 8, regrouping the residual rows reveals distinct blocks in their pairwise cosine similarity matrix, suggesting that further local low-rank approximation may be effective. We therefore construct a hierarchical representation by repartitioning and approximating successive residuals. Repeating this procedure over M levels gives

$$\hat{\mathbf{W}} = \sum_{m=1}^{M} \hat{\mathbf{W}}^{(m)}, \quad \hat{\mathbf{W}}^{(m)} \in \mathcal{H}(\pi_m, \mathbf{r}^{(m)}), \tag{8}$$

where $\mathcal{H}(\pi, \mathbf{r}) = \{\mathbf{H} : \operatorname{rank}(\mathbf{H}_{\mathcal{I}_k,:}) \le r_k, \forall k\}$ denotes the class of multi-subspace matrices under partition π and rank profile $\mathbf{r}$. In Proposition 3, we shows that under the stated subspace assumptions and partition constraints, two levels can admit exact reconstruction at a lower factor cost than a single level. We present more details in Appendix C.

Algorithm 1 Hierarchical Multi-Subspace Compression (HMC)

Input: Weight matrix $\mathbf{W}$, regularized second-moment matrix $\mathbf{C}$, level budgets $\{\mathcal{B}_m\}_{m=1}^{M}$, cluster counts $\{K_m\}$, sweeps S, refinement rounds T

Output: Compressed factors $\{(\mathbf{B}_k^{(m)}, \mathbf{A}_k^{(m)})\}_{m,k}$ and partitions $\{\pi_m\}_{m=1}^{M}$

$\mathbf{R}^{(0)} \leftarrow \mathbf{W}$ ▷ Initialize residual

// Phase 1: Construct the Hierarchical Representation

for $m = 1, \dots, M$ **do**

  $\pi_m \leftarrow \text{kSUBSPACES}(\mathbf{R}^{(m-1)}, K_m)$ ▷ Subspace clustering (§3.1)

  $\mathbf{L}_k^{(m)} \leftarrow \text{chol}(\mathbf{C}_{kk}^{(m)}), \quad \forall k \in [K_m]$ ▷ Cholesky preconditioning

  $\mathbf{r}^{(m)} \leftarrow \text{ALLOCATE}(\{\sigma((\mathbf{L}_k^{(m)})^\top \mathbf{R}_k^{(m-1)})\}_k, \mathcal{B}_m)$ ▷ Adaptive rank allocation (§3.4)

  $\hat{\mathbf{W}}^{(m)} \leftarrow \text{SOLVELEVEL}(\mathbf{R}^{(m-1)}, \pi_m, \mathbf{r}^{(m)}, \{\mathbf{L}_k^{(m)}\}, S, \varnothing)$ ▷ Exact joint block update (§3.2)

  $\mathbf{R}^{(m)} \leftarrow \mathbf{R}^{(m-1)} - \hat{\mathbf{W}}^{(m)}$ ▷ Update residual

end for

// Phase 2: Refine Each Level

for $t = 1, \dots, T$ **do**

  for $m = 1, \dots, M$ **do**

    $\mathbf{R}_{-m} \leftarrow \mathbf{W} - \sum_{j \neq m} \hat{\mathbf{W}}^{(j)}$ ▷ Leave-one-out partial residual

    $\hat{\mathbf{W}}^{(m)} \leftarrow \text{SOLVELEVEL}(\mathbf{R}_{-m}, \pi_m, \mathbf{r}^{(m)}, \{\mathbf{L}_k^{(m)}\}, S, \hat{\mathbf{W}}^{(m)})$

  end for

  if $\Delta J < \epsilon$ **then break** ▷ Stop if objective descent $< \epsilon$

  end if

end for

return $\{(\mathbf{B}_k^{(m)}, \mathbf{A}_k^{(m)})\}_{m,k}$ and $\{\pi_m\}_{m=1}^{M}$

At each level m, we approximate the remaining residual $\mathbf{R}^{(m-1)}$, starting with $\mathbf{R}^{(0)} = \mathbf{W}$. We obtain a new partition π_m by clustering its rows and select the mth level budget $\mathcal{B}_m = \beta_m \mathcal{B}$. Here, $\beta_m > 0$ and $\sum_{m=1}^{M} \beta_m = 1$. With this partition and these ranks fixed, we fit the multi-subspace component by solving

$$\min_{\mathbf{H} \in \mathcal{H}(\pi_m, \mathbf{r}^{(m)})} \text{tr}\Big[(\mathbf{R}^{(m-1)} - \mathbf{H})^\top \mathbf{C} (\mathbf{R}^{(m-1)} - \mathbf{H})\Big], \tag{9}$$

where the closed-form solution of mth level and the kth block $\mathbf{R}_k^{(m)}$ can be solved by Proposition 1, thus guarantees a monotonic decrease in the objective.

We note that the forward pass consists of selecting input channels, applying the local factors, and summing their outputs. All levels receive the same input $\mathbf{x}$ and can be evaluated in parallel:

$$\hat{\mathbf{y}} = \sum_{m=1}^{M} \sum_{k=1}^{K_m} (\mathbf{A}_k^{(m)})^\top (\mathbf{B}_k^{(m)})^\top \mathbf{x}_{\mathcal{I}_k^{(m)}}, \tag{10}$$

where $\mathcal{I}_k^{(m)}$ is the row index set of cluster k in partition π_m. This computation requires neither residual formation nor reconstruction of the dense weight matrix.

3.4 Adaptive Budget-consistent Rank Allocation

Within each level, we allocate ranks to maximize captured spectral energy under the factor budget $\mathcal{B}_{\text{lev}}$. Omitting the level index, let $\mathbf{W}_k$ denote a target residual block and $\sigma_{k,i}$ the ith singular value of $\mathbf{L}_k^\top \mathbf{W}_k$. Each rank increment costs $c_k = n_k + d_{\text{out}}$ parameters, with rank ceiling $m_k = \min(n_k, d_{\text{out}})$. Writing the captured energy as $V(\mathbf{r}) = \sum_{k=1}^{K} \sum_{i=1}^{r_k} \sigma_{k,i}^2$ gives the integer knapsack problem (Kellerer et al., 2004)

$$\max_{\mathbf{r} \in \mathbb{Z}^K} V(\mathbf{r}) \quad \text{s.t.} \quad 1 \leq r_k \leq m_k, \qquad \sum_{k=1}^{K} c_k r_k \leq \mathcal{B}_{\text{lev}}. \tag{11}$$

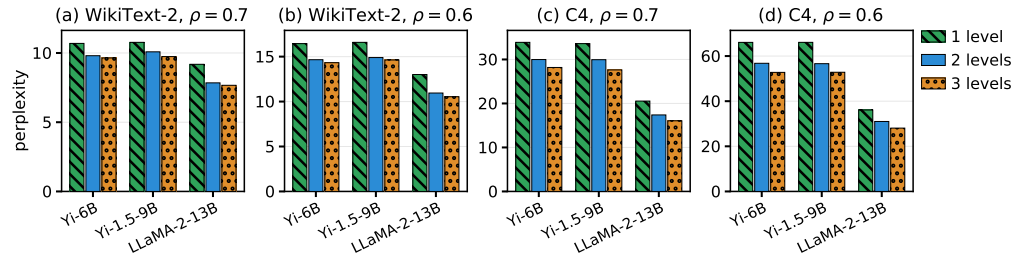


Figure 4: Hierarchical representation effect on WikiText-2 and C4 perplexity. We vary the number of hierarchical levels $M \in \{1, 2, 3\}$, while fixing the target retention ratios and all other settings. The complete results for 5 models are given in Figure 9.

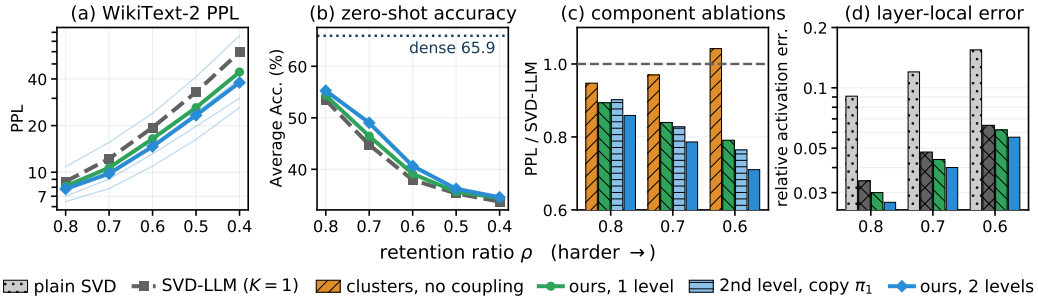


Figure 5: Quality with component ablations. (a) average WikiText-2 perplexity over five models and (b) average six-task mean accuracy over five models. (c) Component ablation measured by the average WikiText-2 perplexity ratio to SVD-LLM. (d) Relative activation error $\mathcal{E}_0 = J_0(\hat{\mathbf{W}})/J_0(\mathbf{0})$ on Yi-6B, averaged over all decoder linear layers.

We use a greedy allocation based on spectral gain per parameter. Starting from rank one per cluster, we add components in decreasing order of $\sigma_{k,i}^2/c_k$, breaking ties by smaller i, until the next component exceeds the remaining budget or all components are selected. Provided $\sum_k c_k \le \mathcal{B}_{\text{lev}}$, the resulting ranks $\mathbf{r}^{\text{G}}$ respect the budget and satisfy

$$0 \le V^\star - V(\mathbf{r}^{\text{G}}) \le \max_{k,i} \sigma_{k,i}^2,$$

where $V^\star$ is the optimum of Eq. (11). In Lemma 1, we give the detailed derivations and proofs for this error bound.

Algorithm 1 summarizes the full compression procedure.

4 Experiments

Our experiments evaluate the effectiveness of HMC and examine the contributions of its key designs. Specifically, §4.1–§4.5 investigate four aspects: ❶ end-to-end compression performance against global low-rank baselines across 5 LLMs; ❷ the effect of the number of hierarchical levels under a fixed factor budget; ❸ the effects of subspace count and grouping strategy; ❹ the contributions of clustering, activation-aware joint fitting, and hierarchical representation through controlled ablations; and ❺ the optimization behavior and inference efficiency. Figure 5 provides an overview of the results, including average WikiText-2 perplexity and zero-shot accuracy across 5 models, component ablations, and activation-weighted reconstruction error on Yi-6B.

Experimental setup. We evaluate 5 decoder-only language models: Yi-6B (Young et al., 2024), Mistral-7B (Jiang et al., 2023), LLaMA-3-8B (Team, 2024), Yi-1.5-9B (Young et al., 2024), and LLaMA-2-13B (Touvron et al., 2023). We compress the decoder linear weights and leave the remaining model weights unchanged. For each compressed matrix, the target retention ratio ρ refers to the factor-parameter budget relative to its dense parameter count. We report WikiText-2 and C4 perplexity, and accuracy over six zero-shot tasks, including ARC-easy, HellaSwag, MathQA, OpenBookQA, PIQA, and WinoGrande. We compare with the activation-aware global low-rank baselines SVD-LLM (Wang et al., 2025c), Swift-SVD (Qi et al., 2026), and ASVD (Yuan et al., 2023) and adopt their official repository.

Table 1: Performance on LLaMA-3-8B at three retention ratios ρ: perplexity (↓), six zero-shot accuracies (↑), and their average (Avg.). Bold marks the best compressed result. Green percentages mark the better HMC variant per metric and budget, relative to the best of baseline. GB values next to ρ denote the shared theoretical FP16 weight budgets (10^9 bytes).

		Perplexity (↓)		Zero-Shot Accuracy (↑)						
ρ	Method	WikiText-2	C4	ARC-e	HellaS.	MathQA	OBQA	PIQA	WinoG.	**Avg.**
1.00 (16.06 GB)	Original	6.14	9.47	0.78	0.79	0.40	0.45	0.81	0.73	0.66
0.8 (13.27 GB)	ASVD	1615.16	1240.19	0.33	0.32	0.22	0.26	0.58	0.55	0.37
	SVD-LLM	15.83	80.86	0.56	0.53	0.25	0.36	0.66	0.64	0.50
	Swift-SVD	14.99	84.44	0.58	0.54	0.25	0.34	0.67	0.64	0.50
	HMC (1-level)	11.70	31.03	**0.61**	0.57	0.28	0.37	0.68	0.67	0.53
	HMC (2-level)	**10.86** (↓27.6%)	**30.34** (↓62.5%)	0.59	**0.60**	**0.29**	**0.38**	**0.69**	**0.69**	**0.54** (↑7.1%)
0.7 (11.87 GB)	ASVD	2.3e+04	1.5e+04	0.27	0.26	0.21	0.28	0.51	0.51	0.34
	SVD-LLM	25.50	169.66	0.44	0.39	0.23	0.30	0.58	0.58	0.42
	Swift-SVD	30.60	279.18	0.43	0.39	0.23	0.30	0.59	0.60	0.42
	HMC (1-level)	17.54	**58.63** (↓65.4%)	0.46	0.42	**0.25**	0.31	0.59	0.61	0.44
	HMC (2-level)	**15.60** (↓38.8%)	58.68	**0.47**	**0.46**	**0.25**	**0.32**	**0.62**	**0.63**	**0.46** (↑7.7%)
0.6 (10.48 GB)	ASVD	3.5e+04	3.1e+04	0.26	0.26	0.19	**0.28**	0.52	0.48	0.33
	SVD-LLM	42.75	337.82	0.32	0.31	0.22	0.26	0.52	0.52	0.36
	Swift-SVD	42.07	328.67	0.32	0.30	0.21	0.28	0.54	0.55	0.37
	HMC (1-level)	27.89	**110.57** (↓66.4%)	0.35	0.33	**0.23**	**0.28**	0.55	0.55	0.38
	HMC (2-level)	**23.98** (↓43.0%)	134.14	**0.36**	**0.35**	**0.23**	0.27	**0.56**	**0.56**	**0.39** (↑6.4%)

Table 2: Performance on Mistral-7B. Metrics and percentage annotations follow Table 1.

		Perplexity (↓)		Zero-Shot Accuracy (↑)						
ρ	Method	WikiText-2	C4	ARC-e	HellaS.	MathQA	OBQA	PIQA	WinoG.	**Avg.**
1.00 (14.48 GB)	Original	5.25	8.11	0.80	0.81	0.36	0.44	0.82	0.74	0.66
0.8 (11.69 GB)	ASVD	19.48	31.14	0.50	0.48	0.24	0.31	0.67	0.55	0.46
	SVD-LLM	8.51	26.73	0.67	0.55	0.28	0.35	0.69	0.67	0.53
	Swift-SVD	8.02	24.02	0.70	0.57	0.28	0.36	0.70	0.66	0.55
	HMC (1-level)	7.26	14.93	0.68	0.61	0.29	0.37	0.72	0.70	0.56
	HMC (2-level)	**6.99** (↓12.8%)	**14.50** (↓39.6%)	**0.72**	**0.65**	**0.31**	**0.40**	**0.74**	**0.70**	**0.58** (↑7.3%)
0.7 (10.30 GB)	ASVD	314.71	400.87	0.29	0.28	0.23	0.24	0.53	0.49	0.34
	SVD-LLM	12.23	50.20	0.52	0.42	0.25	0.33	0.61	0.61	0.46
	Swift-SVD	10.79	41.88	0.56	0.46	0.24	0.34	0.63	0.61	0.47
	HMC (1-level)	10.09	25.23	0.56	0.47	0.25	**0.34**	0.64	0.63	0.48
	HMC (2-level)	**9.05** (↓16.2%)	**24.42** (↓41.7%)	**0.62**	**0.50**	**0.26**	**0.34**	**0.66**	**0.64**	**0.50** (↑6.4%)
0.6 (8.90 GB)	ASVD	8087.24	7056.97	0.27	0.26	0.21	0.27	0.50	0.49	0.33
	SVD-LLM	19.15	94.45	0.37	0.34	0.22	0.30	0.56	0.55	0.39
	Swift-SVD	16.92	92.26	0.42	0.36	0.22	**0.32**	0.58	0.57	0.41
	HMC (1-level)	14.80	44.87	0.43	0.37	0.23	0.28	0.58	0.56	0.41
	HMC (2-level)	**13.13** (↓22.4%)	**43.50** (↓52.9%)	**0.46**	**0.39**	**0.23**	0.31	**0.59**	**0.59**	**0.43** (↑4.2%)

4.1 End-to-End LLM Compression Performance

Tables 1 and 2 report detailed results for LLaMA-3-8B and Mistral-7B, while Appendix B.3 reports the remaining three models (Tables 5-7). All comparisons and analysis below refer to the complete five-model results at the budgets reported in the table.

Language modeling performance. Across all five models and the reported budgets, the proposed HMC achieves lower WikiText-2 and C4 perplexity than SVD-LLM and ASV. Compared with Swift-SVD, it achieves lower C4 perplexity in every setting and lower WikiText-2 perplexity except on Yi-1.5-9B at $\rho = 0.6$. The relative reduction in WikiText-2 perplexity over SVD-LLM grows as the parameter budget decreases on every evaluated model. This trend is consistent with our motivation in Section 1, i.e., a tighter budget reduces the rank of a global factorization, leaving fewer shared directions to represent different groups of weight. By assigning local base to these groups, HMC can preserve distinct local structures within the same parameter budget.

Downstream task performance. The benefits also extend to zero-shot task accuracy. HMC achieves higher six-task mean accuracy than SVD-LLM and ASVD in every reported setting and outperforms Swift-SVD in most settings. Note that the gains vary across tasks, so lower perplexity does not exactly imply higher accuracy on individual benchmark. Overall, these results show that the hierar-

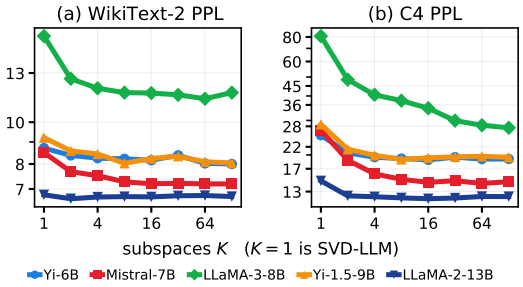


Figure 6: Multi-subspace effect on WikiText-2 and C4 perplexity. We vary the number of subspaces $K \in \{1, 2, 4, 8, 16, 32, 64, 128\}$ on five LLMs.

Table 3: Effect on subspace grouping strategies for multi-subspace partition on WikiText-2 perplexity (↓). We compare k-subspaces clustering with k-means and random partition. All results on 5 models are given in Table 9.

Model	ρ	random	k-means	k-subspaces
LLaMA-3-8B	0.8	16.40	13.71	**11.70**
	0.7	25.30	21.03	**17.54**
	0.6	41.65	33.53	**27.89**
Mistral-7B	0.8	8.20	7.81	**7.26**
	0.7	12.33	10.98	**10.09**
	0.6	19.16	16.46	**14.80**

chical multi-subspace representation preserves downstream performance across the tested budgets, in addition to improving language modeling perplexity.

4.2 Effect of Hierarchical Levels on Compression Performance

In this part, we investigate how hierarchical representation of HMC affects performance under a fixed budget. Figures 4 and 9 compare $M \in \{1, 2, 3\}$ using the prescribed budget splits for each level.

Benefits of hierarchical representation. From the experimental results, we observe that three-level HMC achieves lower WikiText-2 and C4 perplexity than the two-level variant in most cases. These results support our motivation that successive residuals retain multi-subspace structure that additional levels can capture. However, the third level generally brings smaller improvements than the second. This may reflect a budget trade-off since allocating parameters to additional levels leaves fewer for earlier components, potentially limiting the overall improvement.

4.3 Effect of the Number of Subspaces and Subspace Partitioning

In this part, we investigate how the subspace grouping strategy and the number of subspaces affect LLM compression performance under a fixed factor budget.

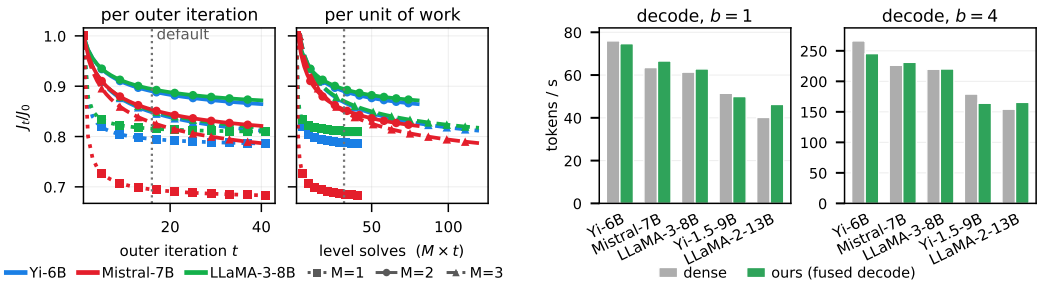
Effect of the number of subspaces. Figure 6 varies the number of subspaces K within a single level, with $K = 1$ corresponding to the SVD-LLM approximation. Increasing K initially lowers perplexity across all models, supporting the benefit of using local subspaces for LLM compression. Further increases generally bring smaller gains, although C4 perplexity on LLaMA-3-8B continues to decrease over the tested range. This is mainly because the the budget trade-off, since finer partitions capture more local structure, but storing more local bases can limit the rank available within each group.

Effect of the grouping strategy. Table 3 compares random partitioning, k-means, and k-subspaces using the same cluster count and compression budget. Note that for fairness, the random partition preserves the same cluster sizes of the k-subspaces partition. From the experimental results, we observe that k-subspaces consistently achieves the lowest WikiText-2 perplexity. The consistent conclusion holds across all five models reported in Table 9. These results demonstrate that grouping weight vectors by subspace similarity achieves better compression performance than the alternative grouping strategies.

4.4 Effect of Key Components

In this part, we evaluate the contributions of key components, particularly joint fitting with cross-cluster activation correlations using WikiText-2 perplexity. Table 4 presents the main results, with complete results in Table 8.

Effect of using only within-cluster activation correlations. The $\mathbf{C}_{kk}$ only variant groups weight vectors into local subspaces and fits each group independently using only $\mathbf{C}_{kk}$. This retains the first term in Eq. (5) and forcing the cross-cluster terms $\mathbf{C}_{kl} = 0$. Compared with the global approxima-



(a) (b)

Figure 7: Optimization and inference at $\rho = 0.6$. *(a)* Mean J_t/J_0 over projection traces for $M \in \{1, 2, 3\}$, with J_0 the first recorded objective of each trace. Vertical lines mark the default two-level iteration budget. *(b)* Dense and single-level decode throughput at batch sizes 1 and 4 on an A100-40GB with fp16 and CUDA-graph capture.

Table 4: Component ablations on two LLMs, with complete results in Table 8. We report WikiText-2 perplexity (↓).

		Global SVD (K=1)			Multi-Subspace Model (Ours)		
Model	ρ	SVD	+ diag.	+ whitening	$\mathbf{C}_{kk}$ only	full $\mathbf{C}$	+ 2 levels
LLaMA-3-8B	0.8	6.3e+05	2154.81	15.83	13.05	11.70	**10.86**
	0.7	1.6e+05	6826.83	25.50	22.09	17.54	**15.60**
	0.6	6.4e+04	5.9e+04	42.75	39.57	27.89	**23.98**
Mistral-7B	0.8	1.6e+05	16.32	8.51	8.06	7.26	**6.99**
	0.7	5.7e+04	57.59	12.23	11.87	10.09	**9.05**
	0.6	8.2e+04	596.89	19.15	19.95	14.80	**13.13**

tion in the + whitening column, this variant lowers perplexity in most settings in Tables 4 and 8, although some settings worsen. These improvements show that clustering can benefit compression even when fitting uses only within-cluster activation correlations.

Effect of joint cross cluster activation. The full $\mathbf{C}$ variant corresponds to single-level HMC and optimizes the complete objective in Eq. (5). It achieves lower perplexity than the $\mathbf{C}_{kk}$ only variant, demonstrating the benefit of activation correlations across clusters when fitting the local factors.

Effect of hierarchical representation. Two-level HMC further reduces perplexity over full $\mathbf{C}$ in every setting and achieves the best results in Table 4, demonstrating the benefit of hierarchical representation.

4.5 Optimization Behavior and Inference Efficiency

Figure 7(a) shows monotonic decreases in the displayed mean objectives, with smaller gains beyond 16 iterations, consistent with Proposition 1. Appendix B.4.6 presents the full experiment details.

Figure 7(b) shows faster single-level decoding than the dense model on Mistral-7B, LLaMA-3-8B, and LLaMA-2-13B. LLaMA-2-13B achieves the largest speedups, approximately 1.15× and 1.07× at the two cases, while both Yi models are a little bit slower. The throughput benefit thus depends on the model and batch size; Appendix B.5.2 gives implementation details.

5 Conclusion

In this work, we introduced HMC, a post-training LLM compression method designed to capture the local low-rank structures of pretrained weights via a hierarchical multi-subspace representation. By partitioning weights into local subspaces, adaptively allocating ranks, and jointly fitting factors using activation correlations, HMC overcomes the representational bottlenecks of standard global factorization. Under the subspace assumptions, our analysis establishes the accuracy and parameter benefits enabled by HMC. Extensive experiments across 5 LLMs demonstrate that HMC generally achieves lower perplexity and higher zero-shot accuracy than standard global low-rank baselines, with ablations confirming the efficacy of its core methodological components.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with experimental design, implementing methods, checking mathematical proofs, and interpreting experimental results. We have not used generative AI tools for formulating the initial research idea or core method, which were developed by the authors, and qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for manuscript drafting and language editing. We have reviewed all AI-assisted work. Specifically, we checked the manuscript text against the experimental results, tested the AI-assisted code, and independently checked the mathematical proofs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Keivan Alizadeh, Seyed Iman Mirzadeh, Dmitry Belenko, S. Khatamifard, Minsik Cho, Carlo C Del Mundo, Mohammad Rastegari, and Mehrdad Farajtabar. LLM in a flash: Efficient large language model inference with limited memory. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12562–12584. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.678. URL https://aclanthology.org/2024.acl-long.678/.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/316648eb8b4ffb6010f531b07848c300-Abstract-Conference.html.

Patrick H. Chen, Si Si, Yang Li, Ciprian Chelba, and Cho-Jui Hsieh. GroupReduce: Block-wise low-rank approximation for neural language model shrinking. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/a2b8a85a29b2d64ad6f47275bf1360c6-Abstract.html.

Zhuo Chen, Minghao Li, Xiaoqian Ma, Siqi Fan, Xiusheng Huang, Zhang Liujie, and Weihang Chen. HiSVD: Principled low-rank approximation of LLMs via hierarchical modeling of information capacity and spectral structure. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20659–20677. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.946. URL https://aclanthology.org/2026.acl-long.946/.

George B. Dantzig. Discrete-variable extremum problems. *Operations Research*, 5(2):266–288, 1957. doi: 10.1287/opre.5.2.266.

Emily Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems*, volume 27, 2014. URL https://papers.neurips.cc/paper_files/paper/2014/hash/1adaeb993eba95859121a43ea61bd858-Abstract.html.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://papers.neurips.cc/paper_files/paper/2022/file/c3ba4962c05c49636d4c6206a97e9c8a-Paper-Conference.pdf.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. doi: 10.1007/BF02288367.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 10323–10337. PMLR, 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4 edition, 2013. doi: 10.56021/9781421407944. URL https://www.press.jhu.edu/books/title/10678/matrix-computations.

Jialong Guo, Xinghao Chen, Yehui Tang, and Yunhe Wang. Slimllm: Accurate structured pruning for large language models. *ArXiv*, abs/2505.22689, 2025. URL https://api.semanticscholar.org/CorpusID:278996555.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf.

Yen-Chang Hsu, Ting Hua, Sung-En Chang, Qian Lou, Yilin Shen, and Hongxia Jin. Language model compression with weighted low-rank factorization. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=uPv9Y3gmAI5.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023. URL https://api.semanticscholar.org/CorpusID:263830494.

Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer Berlin Heidelberg, 2004. doi: 10.1007/978-3-540-24777-7. URL https://link.springer.com/book/10.1007/978-3-540-24777-7.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.html.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. MobileLLM: Optimizing sub-billion parameter language models for on-device use cases. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32431–32454. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24ce.html.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/44956951349095f74492a5471128a7e0-Abstract-Conference.html.

L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960. doi: 10.1093/qmath/11.1.50. URL https://academic.oup.com/qjmath/article-abstract/11/1/50/1525786.

Mohanad Odema, Gabrielle De Micheli, Dayin Gou, Nilesh Malpeddi, Prathamesh Vaste, and Jacob Song. Understanding calibration and truncation error propagation in training-free low-rank compression for LLMs. arXiv preprint arXiv:2608.08506, 2026. URL https://arxiv.org/abs/2608.08506.

Tetiana Parshakova, Trevor Hastie, Eric Darve, and Stephen Boyd. Factor fitting, rank allocation, and partitioning in multilevel low rank matrices. In Ashkan Nikeghbali, Panos M. Pardalos, and Michael Th. Rassias (eds.), *Optimization, Discrete Mathematics and Applications to Data Sciences*, volume 220 of *Springer Optimization and Its Applications*, pp. 135–173. Springer, Cham, 2025. doi: 10.1007/978-3-031-78369-2_9. URL https://link.springer.com/chapter/10.1007/978-3-031-78369-2_9.

Ruoling Qi, Yirui Liu, Xuaner Wu, Xiangyu Wang, Ming Li, Chen Chen, Jian Chen, Yin Chen, and Qizhen Weng. Swift-SVD: Theoretical optimality meets practical efficiency in low-rank LLM compression. In *International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=nAQ4h8FpdM.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=8Wuvhh0LYW.

Shiva Sreeram, Alaa Maalouf, Pratyusha Sharma, and Daniela Rus. Compress to impress: Efficient LLM adaptation using a single gradient step on 100 samples. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/a029fd88d3b4e5480e93f79f15a9036c-Abstract-Conference.html.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=PxoFut3dWW.

Llama Team. The llama 3 herd of models. *ArXiv*, abs/2407.21783, 2024. URL https://api.semanticscholar.org/CorpusID:282614980.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Peng Wang, Huikang Liu, Anthony Man-Cho So, and Laura Balzano. Convergence and recovery guarantees of the k-subspaces method for subspace clustering. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 22884–22918. PMLR, 2022. URL https://proceedings.mlr.press/v162/wang22r.html.

Qinsi Wang, Jinghan Ke, Masayoshi Tomizuka, Kurt Keutzer, and Chenfeng Xu. Dobi-SVD: Differentiable SVD for LLM compression and some new perspectives. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/218ca0d92e6ed8f9db00621e103dc70c-Abstract-Conference.html.

Xin Wang, Samiul Alam, Zhongwei Wan, Hui Shen, and Mi Zhang. SVD-LLM V2: Optimizing singular value truncation for large language model compression. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4287–4296. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.naacl-long.217. URL https://aclanthology.org/2025.naacl-long.217/.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *International Conference on Learning Representations (ICLR)*, 2025c. URL https://openreview.net/forum?id=LNYIUouhdt.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 38087–38099. PMLR, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. ASVD: Activation-aware singular value decomposition for compressing large language models. arXiv preprint arXiv:2312.05821, 2023. URL https://arxiv.org/abs/2312.05821v5. Revised August 2025.

Jingjing Zheng, Wanglong Lu, Yiming Dong, Chaojie Ji, Yankai Cao, and Zhouchen Lin. Adamss: Adaptive multi-subspace approach for parameter-efficient fine-tuning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/1c85c302ece39939c1b334c78f7ee1b8-Abstract-Conference.html.

Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, et al. A survey on efficient inference for large language models. *arXiv preprint arXiv:2404.14294*, 2024.

A Related Work

A.1 Activation-aware Global Low-rank Compression

Post-training low-rank compression increasingly uses data-aware objectives to preserve model behavior. FWSVD (Hsu et al., 2022) weights reconstruction errors using Fisher information, while ASVD (Yuan et al., 2023) rescales weight channels using activation statistics. SVD-LLM (Wang et al., 2025c) uses whitening to align singular-value truncation with activation-weighted reconstruction error and sequentially updates low-rank factors to account for earlier compression errors. Swift-SVD (Qi et al., 2026) obtains an optimal approximation for a fixed rank under an activation-aware objective through a single eigendecomposition of the output-activation second-moment matrix. SVD-LLM V2 (Wang et al., 2025b) allocates compression ratios based on truncation loss, while Dobi-SVD (Wang et al., 2025a) learns truncation positions through differentiable optimization. HiSVD (Chen et al., 2026) uses a two-stage allocation strategy that first considers information capacity and then refines ranks based on spectral-tail redundancy. Odema et al. (2026) further revisit rank allocation after correcting calibration activations affected by earlier compression. These methods improve compression while retaining a global low-rank representation for each weight matrix. HMC instead represents groups of weight rows with distinct local subspaces and builds a hierarchical representation by regrouping and approximating successive residuals.

A.2 Multi-subspace Representations in Neural Networks

Local low-rank representations exploit differences among groups of neural network weights. Denton et al. (2014) cluster input and output channels of convolutional layers and approximate the resulting weight blocks with local low-rank factors to accelerate CNN inference. GroupReduce (Chen et al., 2018) groups vocabulary rows of embedding and softmax matrices, fits frequency-weighted low-rank approximations, and iteratively refines the groups. For LLM adaptation, Sreeram et al. (2025) partition rows of selected weight matrices into groups and approximate each group separately, considering both block splitting and subspace clustering. AdaMSS (Zheng et al., 2025) identifies multiple subspaces in pretrained weights and selectively updates their parameters for parameter-efficient fine-tuning (PEFT). Unlike this methods, HMC uses weight subspace partitioning for compression and further introduces a hierarchical representation by repartitioning and approximating successive residuals.

A.3 Quantization and Pruning for LLM Compression

Quantization reduces the numerical precision of weights and activations. LLM.int8() (Dettmers et al., 2022) and SmoothQuant (Xiao et al., 2023) handle activation outliers through mixed-precision computation and equivalent channel scaling, respectively. GPTQ (Frantar et al., 2023), AWQ (Lin et al., 2024), and OmniQuant (Shao et al., 2024) improve low-bit quantization through second-order updates, activation-guided scaling, and learned calibration parameters, respectively. Pruning removes weights or network structures, with early work combining magnitude-based pruning and retraining (Han et al., 2015). SparseGPT (Frantar & Alistarh, 2023) and Wanda (Sun et al., 2024) enable LLM pruning without retraining using second-order reconstruction and activation-weighted importance scores, respectively. LLM-Pruner (Ma et al., 2023) removes coupled structures based on gradient information, while SliceGPT (Ashkboos et al., 2024) removes rows and columns after orthogonal transformations. Different from these methods, HMC instead reduces the number of parameters through local low-rank factors and a hierarchical residual representation.

B Extensive Experimental Results

B.1 Row Partitioning via Subspace Clustering

We obtain the partition π by applying k-subspaces clustering (Wang et al., 2022) to the rows of $\mathbf{W}$. Let $\mathbf{w}_i \in \mathbb{R}^{d_{\text{out}}}$ denote row i as a column vector. For a maximum subspace dimension q, the

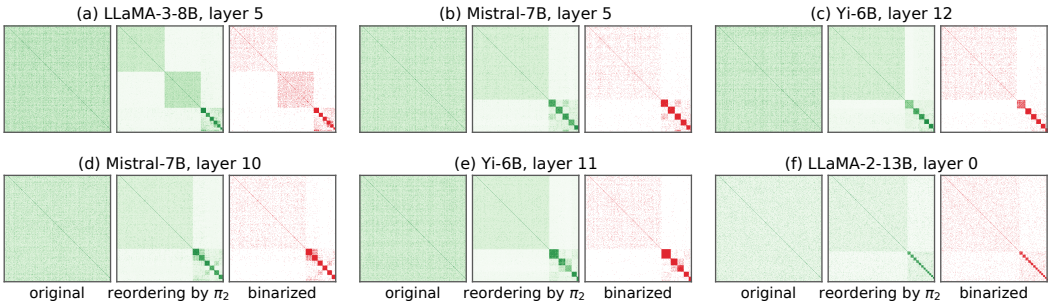


Figure 8: Empirical evidence of Hierarchical Multi-subspace structure. We compute the cosine similarites $|\cos(\mathbf{r}_i^{(1)}, \mathbf{r}_j^{(1)})|$ between the residual rows of $\mathbf{R}^{(1)} = \mathbf{W} - \hat{\mathbf{W}}^{(1)}$. The left original is the similarites in the original row order, and after reordering by π_2, it demonstrates clear block-diagonal structure. The right binarized one is the top 5% binarized version of the middle one.

clustering approximately solves

$$\min_{z_i \in [K],\ \mathbf{V}_k^\top \mathbf{V}_k = \mathbf{I},\ \mathrm{cols}(\mathbf{V}_k) \le q} \sum_{i=1}^{d_{\text{in}}} \left\| \mathbf{w}_i - \mathbf{V}_{z_i} \mathbf{V}_{z_i}^\top \mathbf{w}_i \right\|_2^2, \tag{12}$$

where z_i denotes the cluster assignment of row i, and $\mathbf{V}_k \in \mathbb{R}^{d_{\text{out}} \times q}$ spans the local subspace of cluster k. We alternate between row assignment and basis updates:

$$z_i \leftarrow \arg\max_{k \in [K]} \|\mathbf{V}_k^\top \mathbf{w}_i\|_2^2, \qquad \mathbf{V}_k \leftarrow \mathrm{TopRightSV}_{\min(q, n_k, d_{\text{out}})}(\mathbf{W}_{\mathcal{I}_k,:}). \tag{13}$$

These updates iteratively assign each row to its closest subspace and update the basis using the leading right singular vectors of the assigned block.

B.2 Experimental Configuration

We use the same configuration across all models and retention ratios, except when varying the settings in Appendix B.4. The first level uses $K = 64$ row clusters per weight matrix, with a per-cluster rank cap of 32 in equation 12. For the two-level representation, we use $K_2 = 16$ residual clusters and allocate 0.75 and 0.25 of the factor budget to the first and second levels, respectively. We perform 16 refinement rounds across the two levels and set the damping coefficient to $\lambda = 0.01$, as detailed below. Calibration uses 256 sequences of 2048 tokens from the WikiText-2 training split. We fix the random seed to 0 for calibration and clustering, with no model-specific tuning.

Following GPTQ (Frantar et al., 2023), we instantiate the damped metric in Eq. (2) using

$$\gamma = \lambda \max\left\{ \frac{\mathrm{tr}(\mathbf{C}_0)}{d_{\text{in}}}, 10^{-8} \right\}, \qquad \lambda = 0.01, \tag{14}$$

which sets the damping scale to one percent of the mean diagonal of $\mathbf{C}_0$. The numerical floor guarantees $\gamma > 0$. For each layer, we compute $\mathbf{C}$ once and share it across all clusters and levels. Thus, every block update and monotonicity check uses the same objective J, while reported reconstruction errors use the undamped objective J_0.

B.3 Complete Main Results

Tables 5, 6, and 7 extend the main results to Yi-1.5-9B, Yi-6B, and LLaMA-2-13B, respectively. We report WikiText-2 and C4 perplexity and six-task zero-shot accuracy at target retention ratios $\rho \in \{0.8, 0.7, 0.6\}$ for the Yi models and $\rho \in \{0.7, 0.6, 0.5\}$ for LLaMA-2-13B. Perplexity is evaluated in FP16 after each compressed model is reloaded. All reported perplexities are finite measurements; very large values indicate severe performance degradation after compression. Additional budget sweeps and ablations use the retention ratios specified in their respective figures and tables.

Table 5: End-to-end compression performance on Yi-1.5-9B at target retention ratios $\rho \in \{0.8, 0.7, 0.6\}$. We report WikiText-2 and C4 perplexity, six zero-shot task accuracies, and their average (Avg.). Metric definitions and formatting follow Table 1.

		Perplexity (↓)		Zero-Shot Accuracy (↑)						
ρ	Method	WikiText-2	C4	ARC-e	HellaS.	MathQA	OBQA	PIQA	WinoG.	**Avg.**
1.00 (17.66 GB)	Original	5.48	9.46	0.79	0.78	0.44	0.45	0.81	0.73	0.67
0.8 (14.34 GB)	ASVD	15.51	25.72	0.55	0.61	0.28	0.35	**0.73**	0.61	0.52
	SVD-LLM	9.21	28.66	0.61	0.58	0.29	**0.39**	0.68	0.66	0.53
	Swift-SVD	9.15	27.52	0.62	0.59	0.29	0.37	0.69	0.66	0.54
	HMC (1-level)	8.23	20.50	0.63	0.59	0.30	0.37	0.71	0.66	0.54
	HMC (2-level)	**7.91**	**19.33**	**0.64**	**0.62**	**0.30**	0.37	0.71	**0.68**	**0.55**
0.7 (12.68 GB)	ASVD	336.04	820.40	0.29	0.31	0.20	0.27	0.53	0.48	0.35
	SVD-LLM	12.83	55.58	0.45	0.45	0.25	0.32	0.62	0.60	0.45
	Swift-SVD	12.12	57.88	0.49	0.46	0.25	0.31	0.63	0.61	0.46
	HMC (1-level)	10.77	33.60	0.48	0.47	0.25	0.33	0.65	0.61	0.46
	HMC (2-level)	**10.08**	**29.95**	**0.55**	**0.51**	**0.27**	**0.35**	**0.65**	**0.62**	**0.49**
0.6 (11.02 GB)	ASVD	1.0e+04	2.3e+04	0.26	0.25	0.19	0.27	0.50	0.51	0.33
	SVD-LLM	20.97	117.65	0.35	0.34	0.23	0.29	0.54	0.54	0.38
	Swift-SVD	**14.12**	83.75	0.38	0.35	0.23	**0.30**	0.57	**0.56**	0.40
	HMC (1-level)	16.59	66.08	0.37	0.36	0.23	0.27	0.57	0.54	0.39
	HMC (2-level)	14.90	**56.62**	**0.39**	**0.38**	**0.23**	0.29	**0.58**	**0.56**	**0.41**

Table 6: End-to-end compression performance on Yi-6B at target retention ratios $\rho \in \{0.8, 0.7, 0.6\}$. We report WikiText-2 and C4 perplexity, six zero-shot task accuracies, and their average (Avg.). Metric definitions and formatting follow Table 1.

		Perplexity (↓)		Zero-Shot Accuracy (↑)						
ρ	Method	WikiText-2	C4	ARC-e	HellaS.	MathQA	OBQA	PIQA	WinoG.	**Avg.**
1.00 (12.12 GB)	Original	5.84	9.79	0.77	0.75	0.34	0.41	0.79	0.71	0.63
0.8 (9.91 GB)	ASVD	12.36	23.19	0.53	0.55	0.25	0.32	0.68	0.59	0.49
	SVD-LLM	8.71	25.21	0.58	0.55	**0.27**	**0.37**	0.67	0.62	0.51
	Swift-SVD	8.57	24.61	0.61	0.56	**0.27**	0.36	0.68	0.62	0.52
	HMC (1-level)	8.13	19.83	0.59	0.56	0.26	**0.37**	0.69	0.63	0.52
	HMC (2-level)	**7.81**	**18.48**	**0.64**	**0.58**	**0.27**	**0.37**	**0.71**	**0.64**	**0.53**
0.7 (8.80 GB)	ASVD	153.74	261.19	0.30	0.29	0.22	0.26	0.53	0.52	0.35
	SVD-LLM	12.04	48.69	0.49	0.44	**0.25**	0.31	0.60	0.59	0.45
	Swift-SVD	11.19	46.01	**0.52**	0.46	0.25	0.33	0.62	**0.59**	**0.46**
	HMC (1-level)	10.69	33.89	0.49	0.44	0.24	**0.33**	0.62	0.58	0.45
	HMC (2-level)	**9.80**	**29.98**	0.51	**0.47**	0.24	**0.33**	**0.63**	0.58	**0.46**
0.6 (7.69 GB)	ASVD	2505.85	6386.28	0.27	0.25	0.20	0.26	0.52	0.49	0.33
	SVD-LLM	19.44	101.79	0.35	0.33	0.22	0.26	0.54	0.55	0.38
	Swift-SVD	16.79	111.08	**0.39**	0.36	0.22	**0.31**	0.57	**0.56**	**0.40**
	HMC (1-level)	16.44	66.07	0.36	0.34	0.22	0.26	0.56	0.51	0.38
	HMC (2-level)	**14.66**	**56.79**	0.39	**0.37**	**0.23**	0.27	**0.58**	0.53	**0.40**

B.4 Ablation Studies

Each ablation varies a component or setting described in Appendix B.2 and holds the total factor budget and the calibration data fixed. Tables 8, 9, and 10 use $\rho \in \{0.7, 0.6, 0.5\}$ for LLaMA-2-13B and $\rho \in \{0.8, 0.7, 0.6\}$ for the other models.

B.4.1 Effect of Key Components Across All Models

Table 8 extends the component ablations in Section 4.4 to all 5 models, reporting WikiText-2 perplexity at fixed factor budgets. The first three columns compare global SVD with diagonal activation scaling and full whitening ($K = 1$). The multi-subspace variants then use $K = 64$ row groups, fitting either with only the within-cluster blocks $\mathbf{C}_{kk}$ or with the full matrix $\mathbf{C}$. The final column adds a second residual level while keeping the total factor budget fixed.

Whitening substantially reduces perplexity compared with plain SVD and diagonal scaling. The $\mathbf{C}_{kk}$ `only` variant further improves on whitening in most settings, showing that local subspaces can benefit compression even with independent fitting, although some settings worsen. Using `full` $\mathbf{C}$ lowers perplexity relative to $\mathbf{C}_{kk}$ `only`, demonstrating the benefit of including cross-cluster terms

Table 7: End-to-end compression performance on LLaMA-2-13B at target retention ratios $\rho \in \{0.7, 0.6, 0.5\}$. We report WikiText-2 and C4 perplexity, six zero-shot task accuracies, and their average (Avg.). Metric definitions and formatting follow Table 1.

		Perplexity (↓)		Zero-Shot Accuracy (↑)						
ρ	Method	WikiText-2	C4	ARC-e	HellaS.	MathQA	OBQA	PIQA	WinoG.	**Avg.**
1.00 (26.03 GB)	Original	4.88	6.74	0.77	0.79	0.32	0.45	0.81	0.72	0.64
0.7 (18.42 GB)	ASVD	24.23	30.94	0.44	0.44	0.22	0.32	0.65	0.58	0.44
	SVD-LLM	8.37	24.03	0.52	0.48	0.25	0.37	0.66	0.65	0.49
	Swift-SVD	8.33	25.03	0.52	0.51	0.24	0.37	**0.68**	0.64	0.50
	HMC (1-level)	9.18	20.54	0.52	0.47	0.24	0.36	0.64	0.62	0.48
	HMC (2-level)	**7.84**	**17.39**	**0.58**	**0.54**	**0.25**	**0.38**	0.67	**0.65**	**0.51**
0.6 (15.88 GB)	ASVD	212.12	335.27	0.31	0.29	0.21	0.28	0.55	0.52	0.36
	SVD-LLM	11.77	42.91	0.42	0.37	0.23	0.32	0.59	0.58	0.42
	Swift-SVD	11.41	48.31	0.43	**0.42**	0.23	**0.34**	0.61	**0.61**	0.44
	HMC (1-level)	13.01	36.10	0.41	0.38	0.23	0.31	0.60	0.57	0.42
	HMC (2-level)	**10.95**	**31.01**	**0.45**	0.41	**0.24**	**0.34**	**0.61**	0.60	**0.44**
0.5 (13.34 GB)	ASVD	2212.35	2131.04	0.27	0.27	0.21	0.27	0.50	0.50	0.34
	SVD-LLM	18.75	80.21	0.34	0.32	0.23	0.27	0.55	0.54	0.37
	Swift-SVD	19.58	109.87	0.36	**0.34**	0.22	**0.30**	**0.55**	**0.56**	**0.39**
	HMC (1-level)	20.08	69.05	0.34	0.33	0.22	0.28	0.54	0.54	0.37
	HMC (2-level)	**16.30**	**53.90**	**0.36**	**0.34**	**0.23**	0.29	**0.55**	**0.55**	**0.39**

Table 8: Component ablations on all 5 LLMs, extending Table 4. We report WikiText-2 perplexity (↓) at fixed factor budgets. The $\mathbf{C}_{kk}$ only variant uses within-cluster activation second moments; full $\mathbf{C}$ also includes cross-cluster terms. The + 2 levels variant adds a residual level at the same total budget. Bold marks the lowest reported perplexity in each row.

		Global SVD (K=1)			Multi-Subspace Model (Ours)		
Model	ρ	SVD	+ diag.	+ whitening	$\mathbf{C}_{kk}$ only	full $\mathbf{C}$	+ 2 levels
LLaMA-3-8B	0.8	6.3e+05	2154.81	15.83	13.05	11.70	**10.86**
	0.7	1.6e+05	6826.83	25.50	22.09	17.54	**15.60**
	0.6	6.4e+04	5.9e+04	42.75	39.57	27.89	**23.98**
Mistral-7B	0.8	1.6e+05	16.32	8.51	8.06	7.26	**6.99**
	0.7	5.7e+04	57.59	12.23	11.87	10.09	**9.05**
	0.6	8.2e+04	596.89	19.15	19.95	14.80	**13.13**
Yi-1.5-9B	0.8	2.3e+05	1871.60	9.21	8.71	8.23	**7.91**
	0.7	3.2e+05	2.1e+04	12.83	12.41	10.77	**10.08**
	0.6	4.4e+05	4.6e+04	20.97	21.11	16.59	**14.90**
Yi-6B	0.8	1.4e+05	81.71	8.71	8.57	8.13	**7.81**
	0.7	2.6e+05	367.52	12.04	13.09	10.69	**9.80**
	0.6	4.3e+05	1.2e+04	19.44	23.41	16.44	**14.66**
LLaMA-2-13B	0.7	3.8e+04	36.90	8.37	10.98	9.18	**7.84**
	0.6	2.4e+04	157.88	11.77	19.22	13.01	**10.95**
	0.5	4.8e+04	2371.17	18.75	24.81	20.08	**16.30**

in Eq. (5). Two-level HMC further improves on this single-level variant and achieves the lowest reported perplexity in every row.

B.4.2 Effect of Subspace Partitioning Across All Models

Table 9 extends the grouping comparison in Section 4.3 to all 5 models at $\rho \in \{0.7, 0.6, 0.5\}$ for LLaMA-2-13B and $\rho \in \{0.8, 0.7, 0.6\}$ for the other models. We compare random partitioning, k-means, and k-subspaces using the same cluster count, factor budget and rank allocation procedure. The random partition preserves the cluster sizes of the k-subspaces partition, allowing us to assess the benefit of grouping rows by their subspace structure.

Across all models and retention ratios, k-subspaces achieves the lowest WikiText-2 perplexity, followed by k-means and random partitioning. The perplexity gap between k-subspaces and both al-

Table 9: Partition quality over every model evaluated; conventions as in Table 3.

Model	ρ	random	k-means	k-subspaces
Yi-6B	0.8	9.10	8.31	**8.13**
	0.7	14.03	11.18	**10.69**
	0.6	23.58	17.35	**16.44**
Mistral-7B	0.8	8.20	7.81	**7.26**
	0.7	12.33	10.98	**10.09**
	0.6	19.16	16.46	**14.80**
LLaMA-3-8B	0.8	16.40	13.71	**11.70**
	0.7	25.30	21.03	**17.54**
	0.6	41.65	33.53	**27.89**
Yi-1.5-9B	0.8	9.16	8.65	**8.23**
	0.7	13.92	11.71	**10.77**
	0.6	23.44	18.04	**16.59**
LLaMA-2-13B	0.7	11.42	9.48	**9.18**
	0.6	17.05	13.70	**13.01**
	0.5	28.11	21.37	**20.08**

ternatives widens as the retention ratio decreases. These results show that grouping weight rows by subspace similarity provides more effective local low-rank approximations than random assignment or proximity-based clustering, with a larger perplexity advantage under tighter budgets.

B.4.3 Effect of Residual Repartitioning

Table 10 evaluates whether the second level benefits from a partition adapted to the residual. We compare residual reclustering with a copied-partition control that derives the second-level groups from the first-level partition. Both variants fit the residual using the same total factor budget, budget split, and second-level cluster count.

Residual reclustering improves on a single level in every reported setting. It improves on a single level in most settings but consistently performs worse than reclustering. These results demonstrate the additional benefit of adapting the partition to the residual when constructing the second level. Figure 8 provides supporting evidence by showing that reordering residual rows according to the second-level partition reveals groups with high within-group cosine similarity. This pattern suggests that the new partition can help to capture the local subspace retained in residual matrix.

B.4.4 Hierarchical Levels Across All Models

Figure 9 extends Figure 4 to all 5 models, comparing $M \in \{1, 2, 3\}$ hierarchical levels on WikiText-2 and C4 at $\rho \in \{0.7, 0.6\}$. We fix the total factor budget and all other settings, using budget splits $\beta = (0.75, 0.25)$ for two levels and $\beta = (0.5, 0.3, 0.2)$ for three.

Overall, perplexity tends to decrease as the number of levels increases. Moving from one to two levels lowers WikiText-2 perplexity in every tested setting, and a third level yields further gains in most cases on both datasets. Some curves are nonmonotonic, most visibly for LLaMA-3-8B on C4. A possible reason is that allocating a fixed budget becomes more challenging as the number of levels grows.

B.4.5 Effect of Budget Allocation Across Levels

We study how the allocation of a fixed factor budget between two levels affects perplexity. We vary the first-level share β_1 from 0.10 to 0.90, with $\beta_2 = 1 - \beta_1$, across all 5 models at $\rho = 0.8$. Figure 10 shows the both WikiText-2 and C4 perplexity results.

We can observe that perplexity is relatively insensitive to $\beta_1 \in [0.5, 0.75]$ for most models, and the default achieves perplexity close to the lowest observed value for each model on both evaluation

Table 10: Effect of residual repartitioning on WikiText-2 perplexity (↓) across all 5 models. Both two-level variants fit the residual at the same total factor budget, budget split, and second-level cluster count. The copied-partition control merges first-level clusters by a fixed rule, while the re-partition variant reclusters the residual. Bold marks the lowest reported perplexity in each row.

Model	ρ	1 level	2 levels, copied partition	2 levels, re-partition
Yi-6B	0.8	8.13	8.10	**7.81**
	0.7	10.69	10.29	**9.80**
	0.6	16.44	15.31	**14.66**
Mistral-7B	0.8	7.26	7.21	**6.99**
	0.7	10.09	9.55	**9.05**
	0.6	14.80	13.77	**13.13**
LLaMA-3-8B	0.8	11.70	11.55	**10.86**
	0.7	17.54	16.57	**15.60**
	0.6	27.89	25.76	**23.98**
Yi-1.5-9B	0.8	8.23	8.31	**7.91**
	0.7	10.77	10.62	**10.08**
	0.6	16.59	16.03	**14.90**
LLaMA-2-13B	0.7	9.18	8.23	**7.84**
	0.6	13.01	11.81	**10.95**
	0.5	20.08	18.25	**16.30**

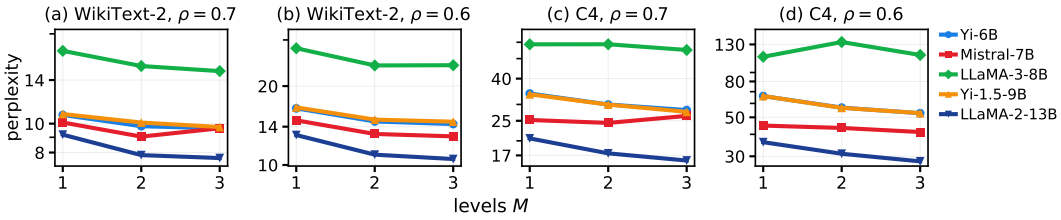


Figure 9: Hierarchical Effect on WikiText-2 and C4 perplexity. The number of hierarchical levels $M \in \{1, 2, 3\}$ for all 5 models at $\rho \in \{0.7, 0.6\}$.

sets. Allocating too little budget to the first level degrades perplexity, particularly at tighter total budgets. This suggests that the first level needs sufficient capacity to capture the dominant structure before the second level approximates the residual. These results support the default split used across models and retention ratios in the main experiments.

B.4.6 Optimization Trace Details

Figure 7(a) uses recorded fitting objectives from one decoder block of Yi-6B, Mistral-7B, and LLaMA-3-8B at $\rho = 0.6$. For each model and level count, we normalize each projection trace by its first recorded objective and take an unweighted mean. The horizontal coordinate Mt accounts for the number of levels updated per round.

Across the curves, the mean normalized objective decreases monotonically, with a rapid initial drop followed by gradual stabilization. For the default two-level setting, most of the reduction occurs within the first 16 rounds, with smaller gains thereafter. These observations are consistent with the descent property in Proposition 1 and support our default iteration budget.

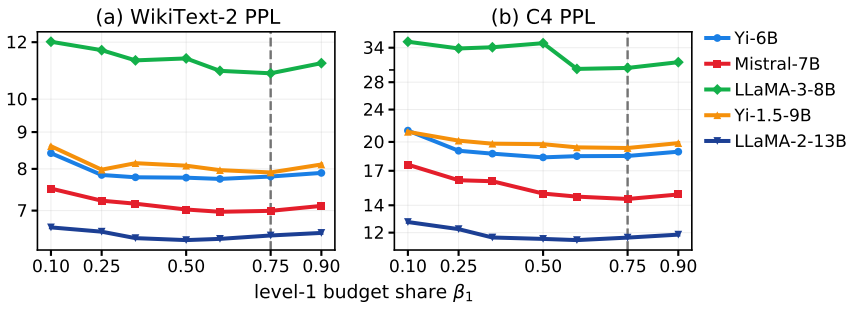


Figure 10: Effect of budget allocation on WikiText-2 and C4 perplexity for all 5 models at $\rho = 0.8$. We vary the first-level budget share β_1 while keeping the total factor budget fixed. Perplexity is shown on logarithmic axes; the dashed line marks the default $\beta_1 = 0.75$.

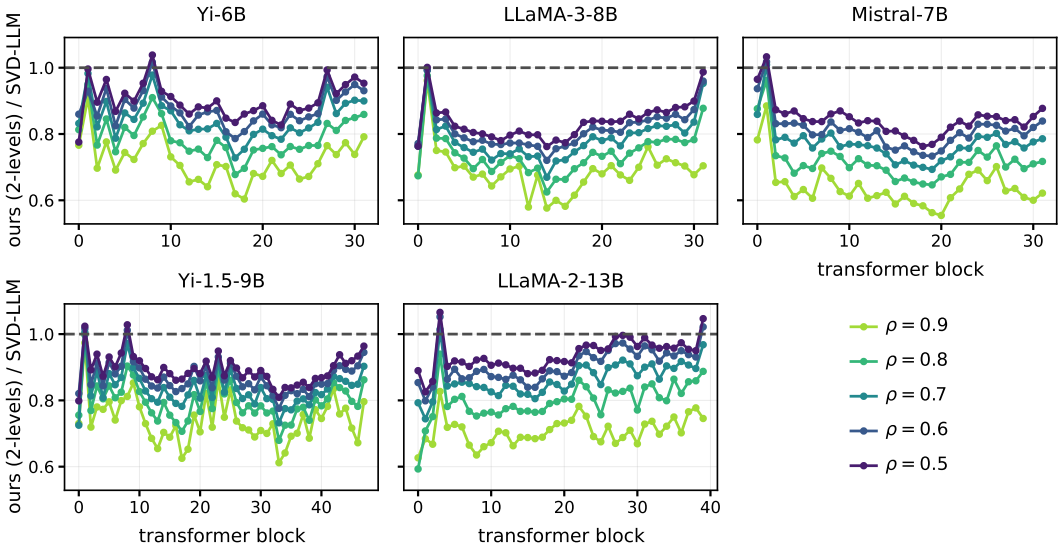


Figure 11: Blockwise local reconstruction error of the representation with two levels, normalized by that of SVD-LLM. Errors are aggregated over all linear layers in each decoder block using their dense parameter counts as weights. Results are shown for all 5 models at $\rho \in \{0.9, 0.8, 0.7, 0.6, 0.5\}$. The horizontal dashed line denotes SVD-LLM, and values below it favor our method. The representation with two levels reduces local reconstruction error for nearly all evaluated blocks, while the few exceptions remain close to the baseline.

B.5 Additional Reconstruction and Inference Efficiency

B.5.1 Local reconstruction error

We compare the local reconstruction quality of our method and SVD-LLM under matched factor budgets using the undamped relative error $\mathcal{E}_0(\hat{\mathbf{W}}) = J_0(\hat{\mathbf{W}})/J_0(\mathbf{0})$.

Figure 5(d) averages this error over the decoder linear layers of Yi-6B, weighted by their dense parameter counts. Figure 11 reports the corresponding averages within each decoder block for all 5 models at $\rho \in \{0.9, 0.8, 0.7, 0.6, 0.5\}$, normalized by those of SVD-LLM. Figures 12, 13, 14, and 15 further break down the error by projection and decoder block at $\rho = 0.7$.

The two-level representation achieves lower aggregate error than SVD-LLM at every reported budget and lower blockwise error in nearly all cases. The few exceptions occur at lower retention ratios, where errors remain close to the baseline. At the projection level, improvements are most consistent for the attention output, MLP down, and value projections, whereas query and key projections show mixed results.

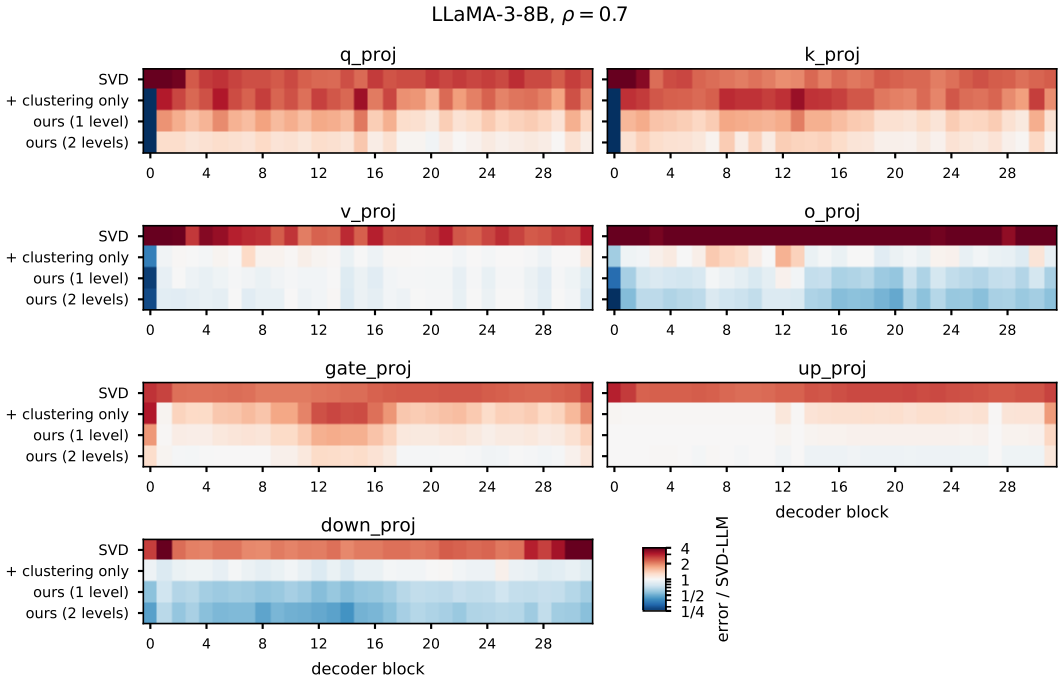


Figure 12: Layer-wise activation-weighted reconstruction error on LLaMA-3-8B at $\rho = 0.7$, relative to SVD-LLM. Each cell is the error ratio for one projection in one decoder block; blue indicates lower error and red higher error. Rows add clustering, joint fitting, and a residual level to the plain-SVD baseline.

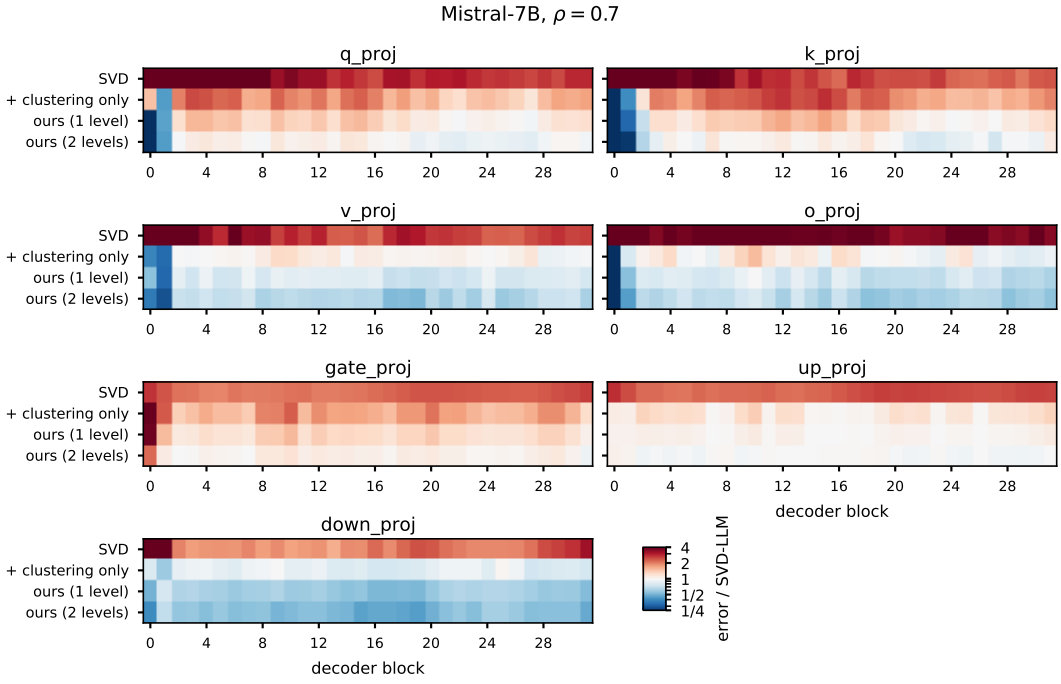


Figure 13: As Figure 12, for Mistral-7B at $\rho = 0.7$.

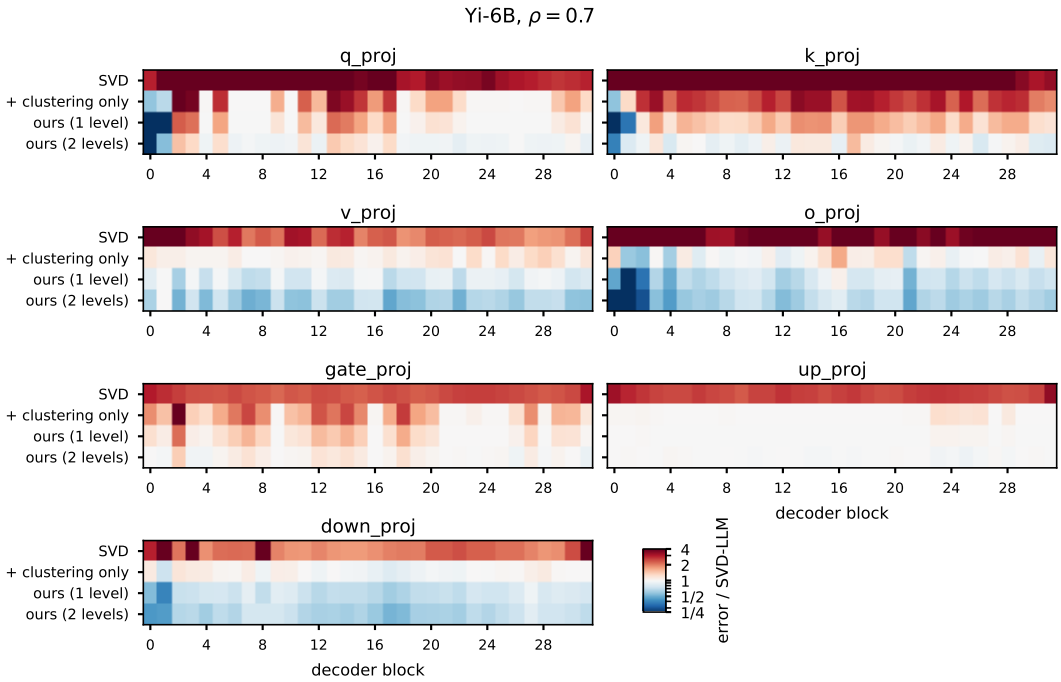


Figure 14: As Figure 12, for Yi-6B at $\rho = 0.7$.

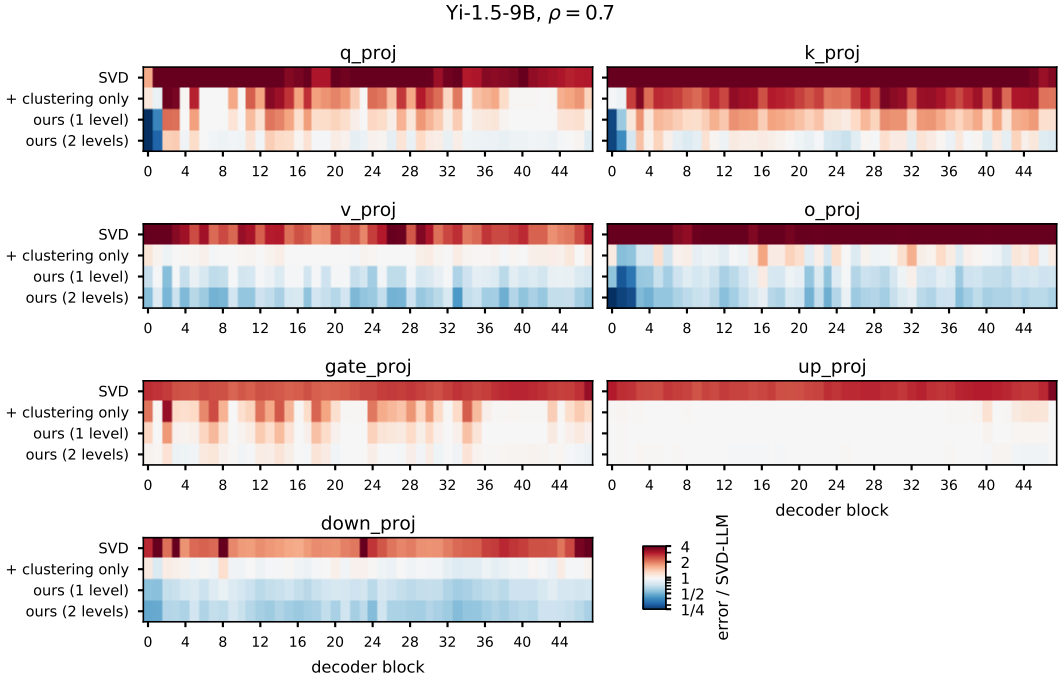


Figure 15: As Figure 12, for Yi-1.5-9B at $\rho = 0.7$.

Algorithm 2 Inference through one compressed layer

Input: $\{(\mathbf{B}_k^{(m)}, \mathbf{A}_k^{(m)})\}_{m,k}$, $\{\pi_m\}_{m=1}^{M}$, $p_{\max} \geq 1$ (max_pad), S streams

procedure PLAN ▷ once, at load

$\mathcal{J} \leftarrow \{j = (m,k)\}, \quad n_j = |\mathcal{I}_k^{(m)}|, \quad r_j = \mathrm{cols}(\mathbf{B}_k^{(m)}), \quad c_0 = \sum_j r_j(n_j + d_{\mathrm{out}})$

$\mathcal{G} \leftarrow \big(\{j\}\big)_{j\in\mathcal{J}}$ ordered by (r_j, n_j)

$\bar{n}_g = \max_{j\in g} n_j, \quad \bar{r}_g = \max_{j\in g} r_j, \quad c(g) = |g|\,\bar{r}_g(\bar{n}_g + d_{\mathrm{out}})$ ▷ padded cost of a group

while $\min_{g,g' \text{ adjacent}} \big[c(g\cup g') - c(g) - c(g')\big] \leq p_{\max}c_0 - \sum_{h\in\mathcal{G}} c(h)$ **do**

$\mathcal{G} \leftarrow \mathcal{G} \setminus \{g, g'\} \cup \{g \cup g'\}$ for the minimising pair

end while

for $g \in \mathcal{G},\ t = 1, \ldots, |g|,\ j = g_t = (m,k)$ **do**

$[\mathbf{I}_g]_{t,:} \leftarrow \big[\pi_m(\mathcal{I}_k^{(m)})\ \ \mathbf{0}_{\bar{n}_g - n_j}\big] \in \mathbb{N}^{\bar{n}_g}$

$[\mathbf{B}_g]_t \leftarrow \begin{bmatrix} \mathbf{B}_j & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{\bar{n}_g \times \bar{r}_g}, \quad [\mathbf{A}_g]_t \leftarrow \begin{bmatrix} \mathbf{A}_j \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{\bar{r}_g \times d_{\mathrm{out}}}$

end for

$\mathbf{i} \leftarrow \big[\mathrm{vec}\,\mathbf{I}_{g_1}; \ldots; \mathrm{vec}\,\mathbf{I}_{g_G}\big], \quad R_0 = \sum_g |g|\,\bar{r}_g, \quad R = 64\lceil R_0/64 \rceil$

$\mathbf{A} \leftarrow \big[\mathbf{A}_{g_1}; \ldots; \mathbf{A}_{g_G}; \mathbf{0}_{(R-R_0)\times d_{\mathrm{out}}}\big] \in \mathbb{R}^{R\times d_{\mathrm{out}}}$

end procedure

procedure FORWARD($\mathbf{X} \in \mathbb{R}^{T\times d_{\mathrm{in}}}$)

$\tilde{\mathbf{X}} \leftarrow \mathbf{X}_{:,\mathbf{i}}$ ▷ 1 kernel

$\mathbf{Z}_{:,R_0+1:R} \leftarrow \mathbf{0}, \quad \mathbf{Z} \in \mathbb{R}^{T\times R}$

for $g \in \mathcal{G}$ on stream $g \bmod S,\ o_g = \sum_{h<g} |h|\bar{r}_h,\ o'_g = \sum_{h<g} |h|\bar{n}_h$ **do**

$\mathbf{Z}_{:,\,o_g+(t-1)\bar{r}_g+1\,:\,o_g+t\bar{r}_g} \leftarrow \tilde{\mathbf{X}}_{:,\,o'_g+(t-1)\bar{n}_g+1\,:\,o'_g+t\bar{n}_g}\,[\mathbf{B}_g]_t, \quad t = 1, \ldots, |g|$ ▷ 1 bmm

end for

return $\mathbf{Y} = \mathbf{Z}\mathbf{A} + \mathbf{1}\mathbf{b}^\top$ ▷ 1 dense GEMM

end procedure

kernels $= 1 + G + 1$; MACs/token $= \sum_g |g|\,\bar{r}_g\bar{n}_g + R\,d_{\mathrm{out}} \leq p_{\max}c_0 + 63\,d_{\mathrm{out}}$

$\mathbf{Y} = \mathbf{X}\hat{\mathbf{W}}$ exactly: $[\mathbf{I}_g]_{t,s} = 0 \Rightarrow [\mathbf{B}_g]_{t,s,:} = \mathbf{0}$, and $\mathbf{Z}$ writes are disjoint across g

B.5.2 INFERENCE EFFICIENCY

Reducing factor storage is useful only if the factors can be evaluated without materializing a dense weight. Algorithm 2 constructs a grouped execution plan at load time. Each forward gathers the assigned input channels, evaluates the grouped first-stage products across all levels, and applies one dense output projection. It therefore forms neither a reconstructed weight nor a dense residual during inference.

Figure 7(b) reports greedy decode throughput at $\rho = 0.6$ on one A100-40GB in fp16 with CUDA-graph capture. At batch 1, the single-level implementation is faster than the dense model on LLaMA-2-13B, Mistral-7B, and LLaMA-3-8B, with the largest gain on LLaMA-2-13B, but is slightly slower on the two Yi models. The same ordering holds at batch 4.

C THEORETICAL ANALYSIS

We compare the factor cost of local and global representations, then show when distinct residual levels avoid duplicating bases in a flat partition. Budgets count dense factor entries and exclude partition metadata.

C.1 LOCAL MULTI-SUBSPACES VERSUS ONE GLOBAL LOW-RANK SUBSPACE

Exact global reconstruction requires a basis spanning all group subspaces. Local factors need only represent their own groups, as in grouped low-rank compression (Chen et al., 2018). The following result shows that this distinction can yield a strict error gap at the same budget.

Proposition 2. *Let π partition the rows of $\mathbf{W}$ into $K \geq 2$ blocks with pairwise orthogonal row spaces of dimension $q \in \mathbb{Z}_{\geq 1}$, and let $\mathbf{C} = c\mathbf{I}$ with $c > 0$. At factor budget $\mathcal{B}^\star = q(d_{\mathrm{in}} + Kd_{\mathrm{out}})$,*

rank-q factors on this partition attain $J_{\text{local}} = 0$. Every global factorization within this budget satisfies

$$J_{\text{global}} \geq c\, q(K-1)\frac{d_{\text{in}}}{d_{\text{in}} + d_{\text{out}}}\sigma_{\min}^2 > 0, \tag{15}$$

where $\sigma_{\min}$ is the smallest nonzero singular value of $\mathbf{W}$.

Each rank-q block has exact local factors, with total cost $\sum_k q(n_k + d_{\text{out}}) = \mathcal{B}^\star$. The orthogonal group subspaces give $\operatorname{rank}(\mathbf{W}) = Kq$, whereas a global factorization with inner dimension r_{global} must satisfy

$$r_{\text{global}} \leq \frac{\mathcal{B}^\star}{d_{\text{in}} + d_{\text{out}}} = Kq - q(K-1)\frac{d_{\text{in}}}{d_{\text{in}} + d_{\text{out}}} < Kq. \tag{16}$$

It therefore leaves nonzero singular directions unrepresented, even at the optimal fit (Eckart & Young, 1936; Mirsky, 1960). Appendix D.3 derives the error bound.

C.2 Hierarchical versus Single-Level Representations

Sums of structured low-rank components have also been studied in multilevel matrix approximation (Parshakova et al., 2025). Different row groupings can share different low-dimensional components. A single-level representation organized by their joint groups stores a separate basis for each group. Two levels can instead reuse the component bases across those groups. We model this structure by

$$\mathbf{w}_i = \mathbf{u}_i + \mathbf{v}_i, \qquad \mathbf{u}_i \in U_{a(i)}, \quad \mathbf{v}_i \in V_{b(i)}. \tag{17}$$

Let $\mathcal{I}_{ab} = \{i : a(i) = a,\ b(i) = b\}$ be a joint row group. The rank condition needed for the comparison is

$$\operatorname{span}\{\mathbf{w}_i : i \in \mathcal{I}_{ab}\} = U_a \oplus V_b \quad \text{for every } (a,b) \in [K] \times [K_2]. \tag{18}$$

This requires the rows of each joint group to span its component subspaces.

Proposition 3 (Two-level representation under crossed structure)**.** *Let $U_1, \ldots, U_K, V_1, \ldots, V_{K_2}$ be mutually orthogonal subspaces of $\mathbb{R}^{d_{\text{out}}}$ with dimensions $q_1, q_2 \in \mathbb{Z}_{\geq 1}$, respectively, and $K, K_2 \geq 2$. Assume Eqs. (17) and (18). Two levels partitioned by a and b, with local ranks q_1 and q_2, admit exact reconstruction at factor cost*

$$\mathcal{B}_H = (q_1 + q_2)d_{\text{in}} + (Kq_1 + K_2 q_2)d_{\text{out}}. \tag{19}$$

Every exact single-level representation whose clusters are unions of complete $\mathcal{I}_{ab}$ groups has cost $F = \sum_g r_g(n_g + d_{\text{out}}) > \mathcal{B}_H$, where $n_g > 0$ and $r_g \geq \operatorname{rank}(\mathbf{W}_g)$.

The proposition compares exact representations and does not guarantee recovery by sequential fitting. Its strict comparison applies to single-level partitions preserving complete joint groups. Appendix D.4 gives the proof.

D Proofs

This appendix proves the propositions in Section 3 and Appendix C and states and proves the rank-allocation lemma. We first establish the block update and allocation guarantees, then prove the representation comparisons and their assumptions. The remaining results quantify the effect of omitting cross-cluster interactions.

D.1 Proof of Proposition 1

The proposition states the minimizer of one block subproblem. The proof is a completion of squares followed by the Eckart-Young-Mirsky Theorem (Eckart & Young, 1936; Mirsky, 1960). The last step in Eq. (21) records why a sweep of exact block updates cannot increase J, which Section 3 uses.

Proof. Let $\mathbf{D}_j = \mathbf{W}_j - \hat{\mathbf{W}}_j$ for every block j, and fix $\mathbf{D}_l$ for $l \neq k$. The symmetry of $\mathbf{C}$ implies $\mathbf{C}_{lk} = \mathbf{C}_{kl}^\top$, and therefore we have

$$\operatorname{tr}(\mathbf{D}_l^\top \mathbf{C}_{lk} \mathbf{D}_k) = \operatorname{tr}(\mathbf{D}_k^\top \mathbf{C}_{kl} \mathbf{D}_l).$$

Note that all terms of J that depend on $\mathbf{D}_k$ are consequently

$$q_k(\mathbf{D}_k) = \operatorname{tr}(\mathbf{D}_k^\top \mathbf{C}_{kk} \mathbf{D}_k) + 2 \operatorname{tr}(\mathbf{D}_k^\top \mathbf{G}_k), \qquad \mathbf{G}_k = \sum_{l \neq k} \mathbf{C}_{kl} \mathbf{D}_l.$$

Every principal block of $\mathbf{C} \succ 0$ is positive definite, so $\mathbf{C}_{kk} = \mathbf{L}_k \mathbf{L}_k^\top$ with $\mathbf{L}_k$ invertible. Expanding the square gives

$$\begin{aligned} q_k(\mathbf{D}_k) &= \|\mathbf{L}_k^\top \mathbf{D}_k + \mathbf{L}_k^{-1} \mathbf{G}_k\|_F^2 - \|\mathbf{L}_k^{-1} \mathbf{G}_k\|_F^2 \\ &= \|\mathbf{T}_k - \mathbf{L}_k^\top \hat{\mathbf{W}}_k\|_F^2 - \|\mathbf{L}_k^{-1} \mathbf{G}_k\|_F^2, \end{aligned} \tag{20}$$

where

$$\mathbf{T}_k = \mathbf{L}_k^\top \mathbf{W}_k + \mathbf{L}_k^{-1} \mathbf{G}_k.$$

The subtracted term in Eq. (20) is independent of $\hat{\mathbf{W}}_k$.

Set $\mathbf{Z}_k = \mathbf{L}_k^\top \hat{\mathbf{W}}_k$. Invertibility of $\mathbf{L}_k$ gives

$$\operatorname{rank}(\mathbf{Z}_k) = \operatorname{rank}(\hat{\mathbf{W}}_k)$$

and reduces the block update to

$$\min_{\operatorname{rank}(\mathbf{Z}_k) \leq r_k} \|\mathbf{T}_k - \mathbf{Z}_k\|_F^2.$$

The Eckart-Young-Mirsky theorem (Eckart & Young, 1936; Mirsky, 1960) yields

$$\mathbf{Z}_k^\star = \mathcal{T}_{r_k}(\mathbf{T}_k), \qquad \hat{\mathbf{W}}_k^\star = \mathbf{L}_k^{-\top} \mathcal{T}_{r_k}(\mathbf{T}_k).$$

The displayed update is a global minimizer of the current block subproblem. Replacing blocks sequentially therefore produces

$$J(\hat{\mathbf{W}}^{(t,k)}) \leq J(\hat{\mathbf{W}}^{(t,k-1)}) \tag{21}$$

at every coordinate step. A fixed-partition, fixed-rank sweep is thus non-increasing in J. □

Algorithm 3 implements the block solver, including the warm start and a numerical monotonicity check.

D.2 Adaptive Rank Allocation: Guarantee and Proof

We use the notation and rank-allocation problem in Section 3.4, Eq. (11). The following lemma bounds the spectral energy lost by greedy allocation relative to the integer optimum.

Lemma 1 (Adaptive greedy allocation)**.** *Assume $\sum_k c_k \leq \mathcal{B}_{\text{lev}}$, and let $V^\star$ be the optimal value of Eq. (11). Starting from $r_k = 1$, add the remaining singular components in non-increasing order of $\sigma_{k,i}^2 / c_k$ (taking smaller i if equal), stopping when the next component would exceed the remaining budget. The resulting greedy ranks $\mathbf{r}^{\text{G}}$ satisfy*

$$\sum_{k=1}^{K} c_k r_k^{\text{G}} \leq \mathcal{B}_{\text{lev}}, \qquad 0 \leq V^\star - V(\mathbf{r}^{\text{G}}) \leq \max_{k,i} \sigma_{k,i}^2.$$

The gains $\sigma_{k,i}^2$ are those of the uncoupled warm start (Section 3.4), so the lemma is a statement about Eq. (11) and not about the coupled objective. The proof is the fractional-knapsack argument with the prefix constraints checked explicitly.

Proof. After reserving rank one per cluster, the residual budget is

$$\mathcal{B}' = \mathcal{B}_{\text{lev}} - \sum_k c_k \geq 0.$$

For every candidate $i \geq 2$, define its gain and density by

$$m_k = \min(n_k, d_{\text{out}}), \qquad g_{k,i} = \sigma_{k,i}^2, \qquad d_{k,i} = \frac{g_{k,i}}{c_k}, \qquad 2 \leq i \leq m_k.$$

Algorithm 3 Joint activation-aware solver for one multi-subspace level

function SOLVELEVEL($\mathbf{R}, \pi, \mathbf{r}, \{\mathbf{L}_k\}, S, \hat{\mathbf{R}}^{\text{in}}$)
  $J_{\mathbf{R}}(\mathbf{H}) \equiv \text{tr}[(\mathbf{R}-\mathbf{H})^\top \mathbf{C}(\mathbf{R}-\mathbf{H})]$
  if $\hat{\mathbf{R}}^{\text{in}} = \varnothing$ **then**
    $\hat{\mathbf{R}}_k \leftarrow \mathbf{L}_k^{-\top} \mathcal{T}_{r_k}(\mathbf{L}_k^\top \mathbf{R}_k)$ for all k
  else
    $\hat{\mathbf{R}} \leftarrow \hat{\mathbf{R}}^{\text{in}}$
  end if
  save $\hat{\mathbf{R}}$ and $J_{\mathbf{R}}(\hat{\mathbf{R}})$ as the best feasible state
  for $s = 1, \ldots, S$ **do**
    $\mathbf{D} \leftarrow \mathbf{R} - \hat{\mathbf{R}}$; compute $\mathbf{CD}$
    for $k = 1, \ldots, |\pi|$ **do**
      $\mathbf{G}_k \leftarrow (\mathbf{CD})_k - \mathbf{C}_{kk}\mathbf{D}_k$
      $\mathbf{T}_k \leftarrow \mathbf{L}_k^\top \mathbf{R}_k + \mathbf{L}_k^{-1}\mathbf{G}_k$
      $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \leftarrow \text{svd}(\mathbf{T}_k)$
      $\mathbf{B}_k \leftarrow \mathbf{L}_k^{-\top}\mathbf{U}_{:r_k}\mathbf{\Sigma}_{:r_k}$; $\mathbf{A}_k \leftarrow \mathbf{V}_{:r_k}^\top$
      $\hat{\mathbf{R}}_k \leftarrow \mathbf{B}_k\mathbf{A}_k$; update $\mathbf{D}_k$ and $(\mathbf{CD})$
    end for
    save the current state; **break if** its change is below tolerance
  end for
  return $\{(\mathbf{B}_k, \mathbf{A}_k)\}$ and $\hat{\mathbf{R}}$
end function

Set $\bar{g} = \max(\{0\} \cup \{g_{k,i} : 2 \le i \le m_k\})$. If there are no remaining items, all residual gains below are zero and the conclusion holds directly. With $x_{k,i} \in [0,1]$, the fractional prefix relaxation is

$$V_F = \max_{\{x_{k,i}\}} \quad \sum_k \sum_{i=2}^{m_k} g_{k,i} x_{k,i} \tag{22}$$

$$\text{subject to} \quad \sum_k \sum_{i=2}^{m_k} c_k x_{k,i} \le \mathcal{B}', \qquad 0 \le x_{k,i} \le 1, \tag{23}$$

$$x_{k,i+1} \le x_{k,i} \quad \text{for every } k, i. \tag{24}$$

Temporarily omit the prefix constraints. Suppose a feasible point assigns positive cost to an item of density d_2 while an item of density $d_1 > d_2$ is not full. Moving a cost amount $\eta > 0$ from the former to the latter changes the objective by

$$\Delta V = \eta(d_1 - d_2) > 0.$$

Thus an optimum exists that fills items in non-increasing density order. Fractional greedy constructs such an optimum and has at most one fractional item (Dantzig, 1957).

For fixed k,

$$\sigma_{k,i+1}^2 \le \sigma_{k,i}^2 \quad \Longrightarrow \quad d_{k,i+1} \le d_{k,i},$$

because all items in cluster k have the same cost c_k. Breaking density ties by increasing rank index therefore gives

$$x_{k,i+1} \le x_{k,i}.$$

The density solution is consequently feasible and optimal for Eq. (24), including its prefix constraints.

Set the unique fractional item, if present, to zero. Let V_G be the rounded gain and V_I the integer optimum. Prefix feasibility is preserved, and

$$0 \le V_F - V_G \le \bar{g} \le \max_{\substack{k \in [K] \\ 1 \le i \le m_k}} \sigma_{k,i}^2.$$

Since the integer feasible set is contained in the fractional one,

$$V_G \le V_I \le V_F \quad \Longrightarrow \quad 0 \le V_I - V_G \le \bar{g}.$$

Adding any later prefix-feasible item that fits can only increase V_G, so the same bound holds for the implementation's integer scan.

The reserved rank-one items contribute the common value $V_0 = \sum_k \sigma_{k,1}^2$. Therefore, the notation in Lemma 1 satisfies

$$V^\star = V_0 + V_I, \qquad \sum_{k=1}^{K} \sum_{i=1}^{r_k^{\mathrm{G}}} \sigma_{k,i}^2 = V_0 + V_G,$$

which proves the stated additive bound for the full allocation.

Finally,

$$\sum_k c_k = \sum_k (n_k + d_{\mathrm{out}}) = d_{\mathrm{in}} + K d_{\mathrm{out}} \le \mathcal{B}_{\mathrm{lev}}$$

is exactly the feasibility condition enforced by the cap on K. □

D.3 Proof of Proposition 2

The comparison imposes the same factor budget on both classes. This budget is smaller than dense storage when $q(d_{\mathrm{in}} + K d_{\mathrm{out}}) < d_{\mathrm{in}} d_{\mathrm{out}}$.

Proof. Since $\mathrm{rank}(\mathbf{W}_k) = q$, each block has exact factors $\mathbf{W}_k = \mathbf{B}_k \mathbf{A}_k$ of inner dimension q. Their total cost is

$$\sum_{k=1}^{K} q(n_k + d_{\mathrm{out}}) = q(d_{\mathrm{in}} + K d_{\mathrm{out}}) = \mathcal{B}^\star. \tag{25}$$

Assembling these blocks gives $\hat{\mathbf{W}} = \mathbf{W}$ and hence $J_{\mathrm{local}} = 0$.

The row space of $\mathbf{W}$ is the sum of its block row spaces. Their pairwise orthogonality makes this a direct sum, so $\mathrm{rank}(\mathbf{W}) = Kq$. Let r be the inner dimension of any global factorization within the budget. Then $\mathrm{rank}(\hat{\mathbf{W}}) \le r$ and

$$Kq - r \ge Kq - \frac{\mathcal{B}^\star}{d_{\mathrm{in}} + d_{\mathrm{out}}} = q(K-1)\frac{d_{\mathrm{in}}}{d_{\mathrm{in}} + d_{\mathrm{out}}} > 0. \tag{26}$$

According to the Eckart–Young–Mirsky theorem (Eckart & Young, 1936; Mirsky, 1960) therefore gives

$$\begin{aligned} J(\hat{\mathbf{W}}) &= c\|\mathbf{W} - \hat{\mathbf{W}}\|_F^2 \ge c \sum_{i>r} \sigma_i^2(\mathbf{W}) \\ &\ge c(Kq - r)\sigma_{\min}^2 \ge c\, q(K-1)\frac{d_{\mathrm{in}}}{d_{\mathrm{in}} + d_{\mathrm{out}}}\sigma_{\min}^2. \end{aligned} \tag{27}$$

The middle inequality counts the $Kq - r$ nonzero singular values beyond rank r. This proves Eq. (15). □

D.4 Proof of Proposition 3

Proof. Define matrices $\mathbf{W}_U$ and $\mathbf{W}_V$ whose ith rows are $\mathbf{u}_i^\top$ and $\mathbf{v}_i^\top$, respectively. Then $\mathbf{W} = \mathbf{W}_U + \mathbf{W}_V$. Grouping $\mathbf{W}_U$ by $a(i)$ gives blocks of rank at most q_1, and grouping $\mathbf{W}_V$ by $b(i)$ gives blocks of rank at most q_2. The corresponding exact factors have total cost

$$q_1(d_{\mathrm{in}} + K d_{\mathrm{out}}) + q_2(d_{\mathrm{in}} + K_2 d_{\mathrm{out}}) = \mathcal{B}_H.$$

Taking $\hat{\mathbf{W}}^{(1)} = \mathbf{W}_U$ leaves residual $\mathbf{W}_V$, which the second level represents exactly.

Now consider a single-level cluster g that is a union of complete joint groups. Let E_g be its set of pair labels, and let $A_g = \{a : (a,b) \in E_g\}$ and $B_g = \{b : (a,b) \in E_g\}$. The spanning assumption gives

$$\mathrm{rowspan}(\mathbf{W}_g) = \sum_{(a,b)\in E_g} (U_a \oplus V_b) = \left(\bigoplus_{a\in A_g} U_a\right) \oplus \left(\bigoplus_{b\in B_g} V_b\right).$$

Mutual orthogonality therefore implies

$$r_g \geq \operatorname{rank}(\mathbf{W}_g) = q_1|A_g| + q_2|B_g| \geq q_1 + q_2. \tag{28}$$

Every cluster is nonempty, and every U_a and V_b index occurs in at least one cluster. Hence

$$\sum_g r_g n_g \geq (q_1 + q_2)d_{\text{in}}, \qquad \sum_g r_g \geq q_1 \sum_g |A_g| + q_2 \sum_g |B_g| \geq Kq_1 + K_2 q_2. \tag{29}$$

Adding the first bound and d_{out} times the second yields $F \geq \mathcal{B}_H$.

Suppose equality holds. Both bounds in Eq. (29) must then be equalities. Because every $n_g > 0$, equality in the first forces $r_g = q_1 + q_2$ for every cluster. Eq. (28) and $q_1, q_2 > 0$ then give $|A_g| = |B_g| = 1$. As complete joint groups cannot be split, the clusters must be exactly the KK_2 joint groups. Their total rank satisfies

$$\sum_g r_g - (Kq_1 + K_2 q_2) = Kq_1(K_2 - 1) + K_2 q_2(K - 1) > 0,$$

contradicting equality in the second bound. Thus $F > \mathcal{B}_H$. □