CARE: COVARIANCE-AWARE AND RANK-ENHANCED DECOMPOSITION FOR ENABLING MULTI-HEAD LATENT ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Converting pretrained attention modules such as *grouped-query attention* (GQA) into *multi-head latent attention* (MLA) can improve expressivity without increasing KV-cache cost, making it attractive for efficient inference. However, existing conversion methods typically apply naïve singular value decomposition (SVD). They focus on minimizing the difference between weight matrices rather than on how those weights affect input activations, ignore the covariance structure of activations, and enforce uniform rank across layers—causing activation drift and degraded attention fidelity. To address these issues, we propose CARE, a ***C**ovariance-**A**ware, **R**ank-**E**nhanced* MLA conversion pipeline under a fixed KV width. CARE introduces three key steps: (i) *activation-preserving factorization*, which aligns the approximation with the actual input activations rather than just the weights; (ii) *adjusted-rank allocation*, which spreads a fixed KV budget across layers by giving more capacity to layers that need it most; and (iii) *KV-parity mapping*, which reparameterizes the converted K and V to fit the MLA format while keeping the KV-cache size unchanged. Under a matched KV-cache budget, our method consistently outperforms a uniform-rank SVD baseline on Llama-3-8B, delivering up to 331% relative gains in one-shot evaluation (higher accuracy, lower perplexity). With a brief post-SVD "healing" fine-tune, we fully recover original model's accuracy.

1 INTRODUCTION

Large Language Models (LLMs) deliver impressive capabilities but at high inference cost, with the key–value (KV) cache in self-attention emerging as a primary memory and bandwidth bottleneck (Vaswani et al., 2017; Kwon et al., 2023). In the standard multi-head attention (MHA) formulation, each head materializes and caches its own keys and values at every decoding step, causing the KV footprint to grow linearly with sequence length and head count. To alleviate this, architecture variants such as multi-query attention (MQA), which shares a single K, V across all heads, and grouped-query attention (GQA), which shares K, V within head groups, have been adopted at scale to shrink KV cache size (Shazeer, 2019; Ainslie et al., 2023; Touvron et al., 2023; Jiang et al., 2023; Chowdhery et al., 2022; Shoeybi et al., 2019). While effective, these variants reduce the number of distinct key/value projections, which can limit attention expressivity and introduce quality regressions when compression is pushed aggressively.

A more recent line of work reframes the KV-cache problem as one of *learned low-rank representation* (Wang et al., 2020; Xiong et al., 2021). Multi-Head Latent Attention (MLA) compresses keys and values into low-dimensional *latent* vectors, caches only these latents, and restores expressivity with lightweight up-down projections at compute time (DeepSeek-AI Team, 2024). In practice, MLA can dramatically reduce KV size while preserving or even improving task accuracy by trading memory and communication for modest extra floating-point-operations (FLOPs) in the projections (DeepSeek-AI Team, 2024; Guo et al., 2025; Liu et al., 2024a; Geens & Verhelst, 2025). Despite these advantages, the ecosystem is dominated by pretrained MHA/GQA checkpoints (Touvron et al., 2023; Jiang et al., 2023; Yang et al., 2024). Retraining large models from scratch under MLA is expensive, so a natural question arises:

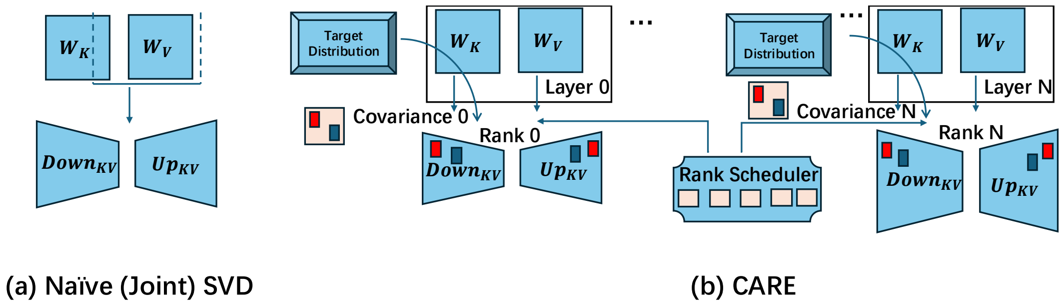


Figure 1: **(a) Naïve (joint) SVD**: directly factorizes concatenated $W_K^{(g)}, W_V^{(g)}$ and truncates to a *uniform* per-layer rank, optimizing parameter error $\|W - \hat{W}\|_F$ and neglecting layer-wise anisotropy—often yielding activation drift. **(b) CARE**: for each layer, estimate input covariance C, compute a covariance-weighted SVD on CW (equivalently, SVD on the whitened operator CW), then *unwhiten* via C^{-1} to initialize $(\mathrm{Down}KV, \mathrm{Up}_{KV})$; use the eigen/singular spectrum of CW to construct an energy curve and select rank r by constrained energy maximization. A global *rank scheduler* enforces KV parity ($r = g_h d_h$) and the total KV-cache budget, after which MLA factors (W^a, W^b) are fine-tuned ("healing") to close residual gaps—preserving activation geometry while improving one-shot accuracy/perplexity over uniform SVD.

Can we *convert* strong, pretrained MHA/GQA models into MLA *post-hoc*, without increasing the KV budget and without incurring large performance loss?

Recent work has explored converting traditional attention (MHA/GQA) into multi-head latent attention (MLA) under fixed KV width. *TransMLA* (Meng et al., 2025) demonstrates that every GQA layer admits an equivalent MLA parameterization and proposes a practical post-training mapping followed by light finetuning. *MHA2MLA* (Ji et al., 2025) generalizes to MHA→MLA by addressing positional encoding mismatches (e.g., partial RoPE adjustments) and initializing W_K, W_V with low-rank joint SVD before efficient recovery (Su et al., 2021b). Together, these works establish MLA as a promising post-hoc target and highlight *low-rank factorization* as central to preserving pretrained knowledge under KV-constrained reparameterizations (Hu et al., 2022; Denil et al., 2013; Denton et al., 2014; Sainath et al., 2013; Eckart & Young, 1936).

However, direct SVD initialization has two key shortcomings. First, it minimizes error in weight space ($\|W - \hat{W}\|$) rather than activation space ($\|XW - X\hat{W}\|$), ignoring how the projection actually operates during decoding (Hassibi et al., 1993; Wang et al., 2024; Yuan et al., 2023). This mismatch induces attention-logit drift even when the weight approximation is accurate. Second, it enforces a uniform rank across layers, neglecting differences in spectral structure. Layers with fast spectral decay are over-compressed, while those with slower decay are under-compressed, leading to fidelity loss and heavier reliance on post-conversion finetuning.

To address the above two shortcomings, we propose CARE, a **C**ovariance-**A**ware, **R**ank-**E**nhanced conversion pipeline, as shown in Fig. 1. First, CARE makes the decomposition *activation-aware*: rather than applying vanilla SVD to W, we solve a whitened approximation problem by applying SVD to CW and then unwhitening to obtain $\hat{W}$, where C summarizes input activation covariance estimated from a modest calibration set. This ensures that dominant activation directions are preserved and substantially reduces attention-logit error before any finetuning. Second, CARE is *rank-adaptive*: it distributes a fixed KV budget across layers and heads based on their singular spectra, allocating higher rank to spectrally complex matrices and lower rank to intrinsically low-rank ones, akin in spirit to budgeted, importance-aware adapter methods (Zhang et al., 2023; Valipour et al., 2023; Hu et al., 2022; Wang et al., 2025a). This budgeted, importance-aware scheduling maintains fidelity under the KV constraint while reducing reliance on post-conversion finetuning.

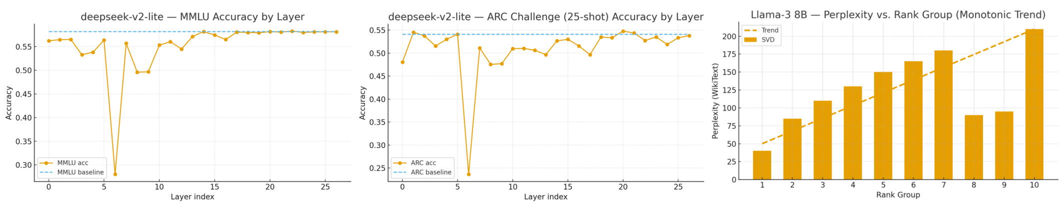
Contributions.

- **Activation-aware decomposition.** We propose a covariance-aware factorization that minimizes activation error $\|XW - X\hat{W}\|$ (rather than weight error), implemented via SVD on a whitened operator and subsequent unwhitening. This preserves attention logits more faithfully at equal KV budget.

- **Rank-adaptive scheduling under fixed KV width.** We introduce a singular-value-guided allocation that distributes rank unevenly across layers/heads and the $\{K, V\}$ matrices, matching spectral difficulty and improving zero-shot fidelity compared with uniform ranks.
- **KV-parity mapping and practical pipeline.** We derive a KV-parity reparameterization for MLA conversion and integrate the above techniques into a practical conversion pipeline CARE-converted models exhibit lower activation error and improved task quality over naive (joint) SVD baselines at equal KV cost, while requiring less data to recover residual gaps.

2 Naïve (Joint) SVD is not enough for MLA Transfer

Multi-head latent attention (MLA) transfer is often initialized with singular value decomposition (SVD), either per matrix or via a joint factorization across related matrices (e.g., W_K, W_V). While convenient, this practice implicitly optimizes weight-space error $\|W - \hat{W}\|_F$ and assumes that the spectrum alone reveals task importance. In this section we show both assumptions break down in practice and, consequently, naïve (joint) SVD is an unreliable recipe for high-fidelity MLA transfer under a fixed KV budget.



(a) (b) (c)

Figure 2: (a)(b): Layer-wise robustness to rank reduction. We randomly reduce the rank of each layer by 50% (ratio-rate = 0.5) in deepseek-v2-lite (Liu et al., 2024a) and plot accuracy on ARC Challenge (25-shot) and MMLU versus layer index, with dashed lines marking the full-rank baselines (ARC 54.09%, MMLU 58.16%). The magnitude of accuracy degradation varies markedly across layers—some layers suffer large drops while others remain near baseline—demonstrating heterogeneous sensitivity to rank reduction. (c): Grouped truncation of MLA attention (layers 30–32) of Llama-3-8B and its effect on Wiki PPL. We partition singular values into 10 magnitude groups sorted descending, and—for each group—truncate it and average the resulting perplexity across the 3 layers. The dashed line marks the ideal monotonic trend expected if "smaller" singular-value groups always had a smaller impact. In practice (bars), groups 8–9, although relatively small in energy, do not yield the worst PPL increase, while later groups can. This non-monotonic sensitivity suggests that SVD's singular value is imperfect proxy for MLA conversion.

Observation 1: Accuracy-preserving rank is *not* uniform across layers. Fig. 2 (a)(b) exhibits pronounced layer-wise heterogeneity when we halve the rank of every layer: some layers tolerate aggressive reduction with negligible loss, whereas others incur sharp drops on ARC and MMLU. Hence, the "safe" rank that preserves accuracy is layer-dependent. A one-size-fits-all policy (uniform pruning or a fixed ratio per layer) either over-compresses fragile layers—degrading task performance—or under-compresses robust layers—wasting KV budget. These results call for rank-aware scheduling that allocates higher rank to sensitive layers while reclaiming capacity from robust ones.

Observation 2: Singular values are poor proxies for accuracy importance. A common heuristic treats singular values as importance scores, expecting that truncating smaller values should least affect accuracy. We directly test this with a brute-force ablation: given $W = U\Sigma V^\top$, we set the i-th singular value to zero and reconstruct $\bar{W}^{(i)} = U\,\mathrm{diag}(\sigma_1, \ldots, 0, \ldots, \sigma_r)\,V^\top$, treating V^T as compress of MLA and U as expand of MLA. As shown in Fig. 2 (c), the link between singular-value magnitude and downstream accuracy is non-monotonic. We conjecture the root cause is mismatch of objectives and statistics: vanilla SVD minimizes weight error, not activation-space error $\|XW - X\hat{W}\|$ under the true (anisotropic) input distribution.

Thus, naïve (joint) SVD isn't enough: (i) the rank needed to keep accuracy varies by layer, and (ii) singular values alone don't reflect accuracy importance.

3 CARE: Covariance-Aware and Rank-Enhanced MLA Conversion

We propose **CARE**, a post-hoc conversion pipeline that maps a pretrained MHA or GQA layer to an MLA layer *at the same KV budget*, while explicitly minimizing *activation* error and *adapting ranks* to spectral difficulty. CARE revisits low-rank factorization through the lens of activation statistics and integrates several components from our prior work (covariance-weighted SVD, non-uniform rank allocation, and low-rank compensation) into a single, practical procedure compatible with LLaMA backbones.

Notation. Given a fixed layer with input activations $X \in \mathbb{R}^{T\times D}$ with length T and embedding dimension D. The multi-head attention in our setup contains n_h heads of size d_h, where we assume the output space corresponds with input space, formally $n_h d_h = D$. We denote g_h to be the number of GQA groups, where for this layer: $Q = XW_Q,\ K = XW_K^{(g)} \in \mathbb{R}^{T\times(g_h d_h)},\ V = XW_V^{(g)} \in \mathbb{R}^{T\times(g_h d_h)}$ with $g_h < n_h$, with $W_{\{\cdot\}}^{(g)}$ represents the weight matrix under the GQA setup. By contrast, an MLA layer with latent rank r uses $K = (XW_K^a)W_K^b$, $V = (XW_V^a)W_V^b$, where $W_{\{\cdot\}}^a \in \mathbb{R}^{D\times r}$, $W_{\{\cdot\}}^b \in \mathbb{R}^{r\times(n_h d_h)}$. Only the latent $XW_{\{\cdot\}}^a \in \mathbb{R}^{T\times r}$ is cached, with $W_{\{\cdot\}}^b \in \mathbb{R}^{r\times(n_h d_h)}$ used to recover KV matrix.

3.1 Recall: KV-Parity Mapping

Grouped-Query Attention (GQA) reduces KV-cache memory by letting multiple heads share the same key–value projection. To convert GQA to Multi-Head Latent Attention (MLA) *without* increasing the KV budget, we enforce **KV parity**. Consider a GQA layer with n_h heads of size d_h with total multi-head hidden size $D = n_h d_h$, we split n_h heads split into g_h groups, where each group contains $\frac{n_h}{g_h}$ heads. The layer l uses $W_Q^{(l)} \in \mathbb{R}^{D\times(n_h d_h)}$ and grouped $W_K^{(l)}, W_V^{(l)} \in \mathbb{R}^{D\times(g_h d_h)}$. We conceptually *replicate* each group's $W_K^{(l)}$ and $W_V^{(l)}$ across its $\frac{n_h}{g_h}$ members to form the full-size $\widetilde{W}_K^{(l)}, \widetilde{W}_V^{(l)} \in \mathbb{R}^{D\times(n_h d_h)}$ (no need to materialize in code). This demonstrates that the GQA method can be reduced to MLA by removing the repeated head blocks (Meng et al., 2025). Therefore, we set MLA's latent rank to match GQA's per-token KV width:

$$r \;=\; g_h\, d_h \tag{1}$$

3.2 Preliminary: Covaraince for input activations

Let $X_b^{(l)} \in \mathbb{R}^{T_b\times D}$ be the bth batch of the domain activations with length T_b at some layer l (note that the length of tokens remains the same across layers, but can be different across batches). These batches of domain activations are used to calculate the extent of preserved rank in Sec. 3.3 and initialize the trainable parameters later in Sec. 3.4.

We then define $C^{(l)}$, the covariance matrix over all the N batches at layer l, as follows:

$$C^{(l)} = \tfrac{1}{N}\sum_{b=1}^{N}(X_b^{(l)})^\top X_b^{(l)}.$$

3.3 Adjusted-Rank Scheduling Across Layers

Due to heterogeneous key / value spectra across different layers, the retained rank of each layer using MLA are supposed to be different. Let $\mathcal{W} = \{\widetilde{W}_K^{(l)}, \widetilde{W}_V^{(l)}\}_{l=1}^{L}$ represent the pretrained KV weights from Sec. 3.1, where L represents the number of layers. Note that $\mathcal{W}$ contains $2L$ weight matrices, each with dimension $D \times n_h d_h$ (recall $D = n_h d_h$).

Given a total rank budget R_{tot}, we *globally* maximize the retained energy:

$$\max_{\{r_K^{(l)}, r_V^{(l)}\}} \sum_{l=1}^{L} \sum_{m=1}^{r_K^{(l)}} \sigma_{K,m}^{(l)} + \sum_{l=1}^{L} \sum_{m=1}^{r_V^{(l)}} \sigma_{V,m}^{(l)} \quad \text{s.t.} \quad \sum_{l=1}^{L} (r_K^{(l)} + r_V^{(l)}) = R_{\text{tot}}, \;\; r_K^{(l)} \le R_K^{(l)}, r_V^{(l)} \le R_V^{(l)} \tag{2}$$

where $\sigma_{K,m}^{(l)}$ (same for V) represents the mth-largest singular value of matrix $C^{(l)}\widetilde{W}_K^{(l)}$, $C^{(l)}$ is the covariance matrix calculated in Sec. 3.2, $\widetilde{W}_K^{(l)} \in \mathcal{W}$ is defined above, $R_K^{(l)}$ is the rank of $C^{(l)}\widetilde{W}_K^{(l)}$ and $r_K^{(l)}$ represents its retained rank with $r_K^{(l)} \le R_K^{(l)}$.

The optimal solution can be computed by greedily water-fill: repeatedly increment rank to the matrix which will increase the most of the sum of singular values. We optionally enforce hardware-friendly multiples (e.g., 8/16) and apply mild priors (e.g., favor V when ties occur).

One can argue that the solution achieved by water-fill maximizes Eq. 2, as shown in Eq. 3 and proved in App. F (Eckart & Young, 1936). We start with the largest singular value of all the candidate matrices, and next time when a selection is made, the value must be smaller than the previous one since it is either the next rank from other matrices (smaller since we chose the largest one), or the next rank from the previously chosen matrix (smaller since singular values are sorted in decreasing order). Therefore, our algorithm selects the R_{tot}-th largest singular values from all the matrices, which results in the largest singular value sum under budget R_{tot} ranks.

3.4 Rethinking SVD with Activation Covariance

A naive rank lowering attempts to minimize Frobenius error $\|W - \widehat{W}\|_F$, where $\widehat{W}$ represents the de-ranked matrix for compressing, and the original pretrained weight matrix to compress, $W := W_{\{\cdot\}}^{(l)} \in \mathcal{W}$, lies in layer l . For inference fidelity, we propose the relevant objective to be minimizing the empirical *activation error*. Focusing on compressing the certain weight matrix W, we denote r to be the target compressed rank calculated in Sec. 3.3. Here $\{X_b := X_b^{(l)} \in \mathbb{R}^{T_b \times D}\}_{b=1}^{N}$ represents the small domain activation batches to compute the low-rank decomposition at layer l, where N is the total number of batches, T_b is the length of each batch and $C := C^{(l)}$ is the covariance matrix defined in Eq. 3.2.

Formally, we try to minimize the following:

$$\min_{\text{rank}(\widehat{W}) \le r} \frac{1}{N} \sum_{b=1}^{N} \|X_b W - X_b \widehat{W}\|_F^2. \tag{3}$$

This optimization formalizes how well $\widehat{W}$ preserves $X_b W$ on relevant inputs X_b. We can tell from the definition of C: $\frac{1}{N}\sum_{b=1}^{N} \|X_b W - X_b \widehat{W}\|_F^2 \; = \| \sqrt{C}(W - \widehat{W}) \|_F^2$.

Heuristically, we use $\|C(W - \widehat{W}) \|_F^2$ as a proxy for $\| \sqrt{C}(W - \widehat{W}) \|_F^2$ to amplify the covariance influence during rank scheduling in 3.3. Formally, this can be argued by letting $C = Q\Lambda Q^\top$ with $\Lambda = \text{diag}(\lambda_i)$, the eigenvalues of C. Then

$$\sqrt{C}\, W = Q\Lambda^{1/2}(Q^\top W), \text{while } CW = Q\Lambda(Q^\top W).$$

Both of them are left-multiplied by the *same* eigenspaces of C; the difference is a re-weighting of directions changes to λ_i^2. Thus CW emphasizes *the same* directions as $\sqrt{C}\, W$ more strongly. Therefore, this tends to preserve the ordering of dominant components and hence the top-rank subspace chosen by truncation.

One can show that the optimal rank-r solution is obtained by truncated SVD of CW, with proof in App. F (Eckart & Young, 1936). This re-weights weight-space directions by usage in the input activation, aligning with importance aware compression (LeCun et al., 1989; 1990b; Frantar et al., 2022).

We therefore compute, for K and V separately,

$$C\,W \;=\; U\,\Sigma\,V^\top, \text{then } \widehat{W} \;=\; C^{-1}\,U_r\,\Sigma_r\,V_r^\top, \tag{4}$$

with U_r, Σ_r, V_r the top-r components of U, Σ, V. In practice, we use a shrinkage $C_\lambda = (1-\alpha)C + \alpha\lambda I$ to ensure C is invertible, with $\alpha \in (0,1)$ and $\lambda > 0$.

Then we initialize the trainable parameters $W^a \in \mathbb{R}^{D\times r}$ and $W^b \in \mathbb{R}^{r\times(n_h d_h)}$ s.t. $W^a W^b$ equals the compressed matrix. We map SVD factors to MLA by

$$W^a \;\leftarrow\; C^{-1}\, U_r\, \Sigma_r^{\frac{1}{2}}, \qquad W^b \;\leftarrow\; \Sigma_r^{\frac{1}{2}}\, V_r^{\top}, \tag{5}$$

so that $W^a W^b = \widehat{W}$. The cached latent $XW^a \in \mathbb{R}^{T\times r}$ in MLA spans the principal activation subspace, where X is the actual input at layer l.

3.5 Healing CARE Converted Models

Based on our initialization of down-and-up matrices W^a, W^b and number of T tokens, we attempt to encode positional information in the attention mechanism. Given layer l, let $Q_t = X_t W_Q$ be the usual query at step t, and let $K_{C,t} = (X_t W_K^a) W_K^b$ and $V_{C,t} = (X_t W_V^a) W_V^b$ be the MLA generated keys/values. Following the decoupled RoPE design in (DeepSeek-AI Team, 2024), we add a small RoPE channel of width d_r by concatenation, where we introduce *new trainable* matrices:

$$W_Q^R \in \mathbb{R}^{D\times(n_h d_r)}, \qquad W_K^R \in \mathbb{R}^{r\times d_r},$$

where $d_r \ll d_h$. Let $\mathcal{R}_t \in \mathbb{R}^{(n_h d_r)\times(n_h d_r)}$ denote the standard block-diagonal RoPE rotation matrix transformation for step t (applied head-wise on each 2D pair). We form:

$$Q_{R,t} \;=\; (X_t W_Q^R)\, \mathcal{R}_t, \qquad K_{R,t} \;=\; \operatorname{repeat}\big((X_t W_K^a) W_K^R\, \mathcal{R}_t\big),$$

where `repeat` function replicates the shared RoPE key across all the heads. We then concatenate the RoPE channels and run standard attention on them:

$$Q_t^\star = \big[\, Q_t \,;\, Q_{R,t} \big], \quad K_t^\star = \big[\, K_{C,t} \,;\, K_{R,t} \big], \quad A_t = \operatorname{Softmax}\!\left(\frac{Q_t^\star (K_t^\star)^\top}{\sqrt{d_h + d_r}}\right), \quad O_t = A_t\, V_{C,t},$$

where A_t denotes the weight of each value vector and O_t denotes the layer output. We only cache latents $X_t W_K^a$ and $X_t W_V^a$, along with RoPE latents $(X_t W_K^a) W_K^R \mathcal{R}_t$ and $Q_{R,t}$, preserving cache efficiency while letting RoPE act directly in attention as in Li et al. (2025b).

We penalize the low-rank decomposition by its cross-entropy classification error and KL-divergence imitation error, namely the loss functions:

$$\mathcal{L}_{\text{CE}} = \; -\tfrac{1}{T} \sum_{t=1}^{T} \log p^{\mathbb{S}}(x_{t+1} \mid x_{\le t}), \tag{6}$$

$$\mathcal{L}_{\text{KD}} = \; \tfrac{1}{T} \sum_{t=1}^{T} \mathrm{KL}\big(\operatorname{softmax}(z_t^{\mathbb{T}}/\tau) \,\big\|\, \operatorname{softmax}(z_t^{\mathbb{S}}/\tau)\big)\,, \tag{7}$$

$$\mathcal{L} = \; \mathcal{L}_{\text{CE}} + \beta\, \tau^2\, \mathcal{L}_{\text{KD}}. \tag{8}$$

Here $p^{\mathbb{S}}(x_{t+1} \mid x_{\le t})$ represents the probability of making the correct prediction of the next token at the first layer based on our student (compressed) model, where each process in the transformer uses down and up samples W^a and W^b instead of the original W. Also, $z_t^{\mathbb{T}}$ and $z_t^{\mathbb{S}}$ represent the teacher's (original pretrained) and student's (compressed) last layer outputs used to predict the next token, with $p^{\mathbb{S}}(x_{t+1} \mid x_{\le t}) = \operatorname{softmax}(\frac{z_t^{\mathbb{S}}}{\tau})_{x_{t+1}}$ with τ the temperature and β balancing the terms (Hinton et al., 2015). Note that both $\mathcal{L}_{CE}$ and $\mathcal{L}_{KD}$ are related to all the W^a, W^b for $\forall W \in \mathcal{W}$, since each input token goes through all the layers to yield the output of the last layer and predict the next token.

4 Experimental Results

Overview. We assess whether **CARE**—Covariance-Aware and Rank-Enhanced decomposition—enables accurate MLA migration *under fixed KV budget* (*KV-parity*). We report perplexity, accuracy on general knowledge suites, long-context robustness, decoding throughput and measured KV footprint.

Conversion protocol. MLA reparameterization follows prior recipes with covariance integration. Post-conversion, we apply brief SFT (same budget across methods). For **CARE**, we estimate $C = \mathrm{Cov}[X]$ from a small calibration set with shrinkage $C \leftarrow (1-\lambda)C + \lambda I$ and factor CW. *Adjusted-Rank* distributes per-layer ranks via water-filling over layerwise importance derived from the weighted singular spectra subject to a global KV budget.

4.1 Original, Baselines, CARE variants and Datasets

GQA (source). Unmodified grouped-query attention. **TransMLA (uniform). (Meng et al., 2025)** Weight SVD on (W_K, W_V) with uniform per-layer rank. **Direct-SVD (energy).** Weight SVD with per-layer ranks chosen by singular-value energy. **MHA2MLA (Ji et al., 2025)** Joint factorization of (W_K, W_V) with partial-RoPE during migration. However cause it involve partial-RoPE, it cannot be applied through zero-shot experiment. **CARE(ours)-U(uniform), CARE(ours)-E(energy)** Covariance-aware factorization with/without Adjusted-Rank. We evaluate models using the LM Harness suite (Gao et al., 2024), conducting both zero-shot and finetune-heal assessments on standard language-understanding tasks. The benchmark encompasses Wikitext2 (Merity et al., 2016), ARC-Challenge (ARC) and ARC-Easy (ARE) (Clark et al., 2018), HellaSwag (HS) (Zellers et al., 2019), MMLU (MM) (Hendrycks et al., 2020), OpenBookQA (OB) (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), RACE (RA) (Lai et al., 2017), and WinoGrande (WG) (Sakaguchi et al., 2021). All hyperparameters are shown in App. D.

Table 1: Zero-shot Llama3.1-8B comparison against original, baselines on multiple tasks. Higher is better for Accuracy (%) (ACC.) (↑) and Lower is better for Perplexity (PPL.) (↓).

Rank	KV Save	Methods	Wiki (↓)	ARC (↑)	ARE (↑)	HellaSwag (↑)	PIQA (↑)	MMLU (↑)	OBQA (↑)	RA (↑)	WG (↑)	AVG (↑)
		GQA (Original)	**6.82**	**50.34**	**80.18**	**60.15**	**79.65**	**48.05**	**34.80**	**40.10**	**72.69**	**58.24**
64	**93.75**	TransMLA	1943.91	20.31	27.95	14.13	55.98	24.17	12.40	20.96	51.38	28.41
		TransMLA (Energy)	1087.57	21.42	27.27	26.34	53.97	23.80	12.60	21.44	52.25	29.89
		CARE-U (OURS)	**693.82**	**18.00**	**31.40**	**26.38**	**55.50**	**23.60**	**12.80**	**20.67**	**50.83**	**29.90**
		CARE-E (OURS)	**576.21**	**19.45**	**32.95**	**26.35**	**57.02**	**23.60**	**14.00**	**21.53**	**50.20**	**30.64**
128	**87.50**	TransMLA	1486.43	20.48	27.36	16.07	54.19	21.53	14.00	21.34	50.36	28.17
		TransMLA (Energy)	992.31	20.22	27.15	26.18	54.79	27.26	11.40	22.97	49.88	29.98
		CARE-U (OURS)	**304.66**	**19.28**	**39.56**	**26.97**	**58.98**	**23.40**	**14.40**	**23.54**	**50.43**	**32.07**
		CARE-E (OURS)	**214.17**	**20.90**	**42.30**	**28.57**	**61.70**	**23.75**	**14.40**	**24.31**	**52.41**	**33.54**
256	**75.00**	TransMLA	626.93	18.94	30.77	15.54	56.04	23.38	12.40	22.39	49.64	28.64
		TransMLA (Energy)	537.39	19.80	30.13	26.95	55.60	23.79	12.40	23.25	49.33	30.16
		CARE-U (OURS)	**63.23**	**24.49**	**55.26**	**32.89**	**67.41**	**30.96**	**16.20**	**27.46**	**55.56**	**38.78**
		CARE-E (OURS)	**39.57**	**30.29**	**60.14**	**39.05**	**71.16**	**34.61**	**19.80**	**30.53**	**61.17**	**43.34**
512	**50.00**	TransMLA	131.05	23.55	45.45	22.13	61.21	23.59	17.40	26.03	53.04	35.75
		TransMLA (Energy)	118.19	29.11	47.05	28.48	59.14	22.87	19.40	29.73	51.54	35.92
		CARE-U (OURS)	**12.15**	**41.30**	**74.03**	**51.55**	**76.66**	**38.50**	**27.80**	**39.33**	**68.67**	**52.23**
		CARE-E (OURS)	**9.45**	**42.24**	**74.16**	**54.18**	**77.53**	**44.33**	**31.20**	**38.66**	**71.03**	**54.17**
1024	**0.00**	TransMLA	6.82	50.34	80.13	59.22	79.60	47.56	33.14	42.07	71.55	57.95
		TransMLA (Energy)	6.82	50.43	80.09	60.17	79.71	48.05	34.80	40.19	72.85	58.28
		CARE-U (OURS)	**6.82**	**50.43**	**80.09**	**60.18**	**79.71**	**48.05**	**34.80**	**40.19**	**72.85**	**58.29**
		CARE-E (OURS)	**6.82**	**50.26**	**80.09**	**60.17**	**79.60**	**47.85**	**34.60**	**40.10**	**72.53**	**58.15**

Tab. 1 summarize performance and efficiency. Across both scales, CARE-U/E deliver lower PPL and higher accuracy than direct SVD methods. Improvements are most pronounced on PPL, indicating that covariance-weighted objectives better preserve attention behavior than weight-only SVD.

4.2 Ablation Studies

We examine how the "energy"—the variance explained along singular directions—concentrates across layers and heads, how this concentration differs *across models*, and how it changes once input-activation covariance is incorporated.

4.2.1 Energy Distribution vs. Covariance

Using the procedure in Sec.3 to compute covariance-aware energy and the rank required to meet a fixed energy target per layer/head, we obtain highly consistent rank profiles across calibration sources C4 (Raffel et al., 2020a), Alpaca (Taori et al., 2023), WikiText2 (Merity et al., 2016), and PTB (Marcus et al., 1993)). As illustrated in Fig. 3 of Rank 64 & 512, both W_K and W_V exhibit a depth-dependent trend: ranks are smallest in early layers, grow steadily through the middle blocks, and remain elevated thereafter. The growth is markedly stronger for W_V. In contrast, W_K increases more moderately, saturating around 550–620 with a mild mid-layer valley and a drop in the final

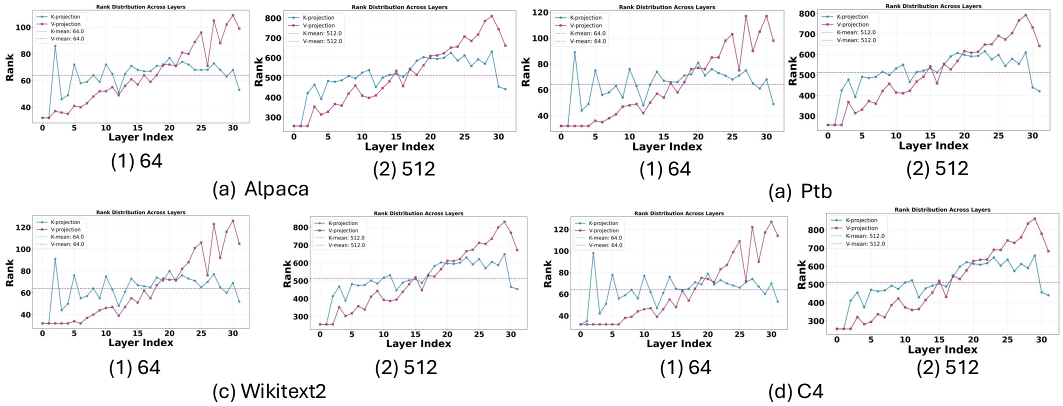


Figure 3: Covariance-aware rank profiles across calibration corpora (Alpaca, WikiText2, PTB, C4) at target ranks 64 and 512. Both W_K and W_V show a depth-dependent increase—small in early layers, rising through mid layers—with stronger late-layer growth for W_V. The consistency across corpora suggests a model-intrinsic trend.

block. The persistence of this shape across all corpora indicates that the rank distribution is largely a *model-intrinsic* property rather than a peculiarity of any single calibration dataset. This also supports results will be similar when applying different calibration dataset on Tab. 1.

4.2.2 Accuracy Impact of Covariance

Table 2: Zero-shot Llama3.1-8B comparison on different covariance. Higher is better for Accuracy (ACC.) (%) (↑) and Lower is better for Perplexity (PPL.) (↓). All methods are evaluated under the same preprocessing.

Rank	Methods	Wiki (↓)	ARC (↑)	ARE (↑)	HellaSwag (↑)	PIQA (↑)	MMLU (↑)	OBQA (↑)	RA (↑)	WG (↑)	AVG (↑)
128	**CARE-E-ALPACA**	214.17	20.90	42.30	28.57	61.70	23.75	14.40	24.31	52.41	33.54
	CARE-E-C4	126.94	18.60	38.05	30.05	63.00	23.46	14.80	24.98	52.72	33.21
	CARE-E-PTB	145.77	17.06	32.66	27.06	56.09	23.33	13.00	22.97	51.14	30.41
	CARE-E-WIKI	40.37	17.92	34.47	27.67	56.86	23.46	14.80	22.87	52.01	31.26
512	**CARE-E-ALPACA**	9.45	42.24	74.16	54.18	77.53	44.33	31.20	38.66	71.03	54.17
	CARE-E-C4	9.09	43.26	75.34	54.69	77.86	40.18	30.40	38.66	71.03	53.93
	CARE-E-PTB	9.29	39.85	72.47	52.56	75.95	39.25	27.40	34.26	71.67	51.68
	CARE-E-WIKI	7.91	41.81	73.57	52.83	76.50	36.97	29.80	35.12	71.19	52.22

We evaluate how the calibration corpus used to estimate activation covariance (C4, Alpaca, Wiki-Text2, PTB) affects *zero-shot* accuracy in Tab. 2. Despite noticeable differences in perplexity, the accuracy effects are small and consistent: per-task scores vary by roughly 1–10 % points across corpora, while the average over the full suite shifts by only $\sim$1–2 percentage points. Overall, calibration choice has a modest impact on final accuracy. We therefore adopt Alpaca as the default calibration set for subsequent SFT–budget fine-tuning.

4.3 Recovery with small SFT budgets

Table 3: Healed Llama3.1-8B comparison against orginal, baselines on multiple tasks. Higher is better for Accuracy (%) (ACC.) (↑) and Lower is better for Perplexity (PPL.) (↓). All methods are evaluated under the same preprocessing.

Rank	Methods	Wiki (↓)	ARC (↑)	ARE (↑)	HellaSwag (↑)	PIQA (↑)	MMLU (↑)	OBQA (↑)	RA (↑)	WG (↑)	AVG (↑)
–	**GQA (original)**	**6.82**	**50.34**	**80.18**	**60.15**	**79.65**	**48.05**	**34.80**	**40.10**	**72.69**	**58.24**
128	TransMLA	53.83	41.76	62.80	46.57	69.33	27.92	20.40	33.6	51.82	44.28
	CARE-E (OURS)	**11.20**	**49.42**	**75.40**	**61.41**	**81.14**	**47.31**	**36.13**	**40.97**	**72.89**	**58.08**
512	TransMLA	10.50	45.90	76.52	56.72	74.00	42.38	31.52	46.43	64.25	54.71
	CARE-E (OURS)	**7.10**	**55.14**	**86.68**	**61.73**	**78.89**	**49.05**	**39.60**	**40.29**	**74.30**	**60.71**

We sweep small-scale SFT budgets to quantify post-conversion recovery and results is shown in Tab 3. Across budgets, CARE consistently requires fewer optimization steps and less data to close the gap to the original model than transMLA, indicating a stronger initialization. Using a TINY SFT corpus of **2.5B** tokens, we recover the pre-conversion accuracy and, on several tasks, achieve additional gains of $\mathbf{3\%-5\%}$.

We provides more supporting results that complement the main paper in App. E. App. E expands our evaluation to other-series models, other system-level metrics measurement and describe "healing" details. We also provide optimization objectives, datasets, and hyperparameters in App. D.

5 Related Works

Conversion from Traditional Attention to MLA. Standard multi-head attention (MHA) underpins modern LLMs but induces a KV cache that scales linearly with sequence length and head width, creating a memory and bandwidth bottleneck at inference time (Vaswani et al., 2017; Kwon et al., 2023). Grouped-Query Attention (GQA) reduces KV heads by sharing keys/values across query groups, lowering KV memory while sacrificing expressiveness (Ainslie et al., 2023; Shazeer, 2019). Multi-Head Latent Attention (MLA) addresses KV memory by caching low-dimensional latents with lightweight up/down projections (DeepSeek-AI Team, 2024). Beyond MLA that trains from scratch, several *post-hoc* pathways demonstrate conversion feasibility: *TransMLA* gives a theory-based reduction from GQA to MLA at corresponding KV budget (Meng et al., 2025), *MHA2MLA* focuses on practical alignment via partial RoPE and joint-SVD initialization (Ji et al., 2025), and *X-EcoMLA* explores distillation-based upcycling of pretrained attention into MLA for extreme KV compression (Li et al., 2025b). Also, *Zebra-Llama* composes efficient hybrids to improve inference efficiency and can be paired with MLA-style KV reductions in deployed systems (Yang et al., 2025).

SVD inspirations. Naive SVD truncation minimizes $\|W - W_r\|_F$ and ranks by raw singular values, which need not correlate with downstream loss (Eckart & Young, 1936; LeCun et al., 1990a; Hassibi & Stork, 1993; Dong et al., 2019; Frantar et al., 2022; Krzanowski, 2000). Recent works refine this in LLMs: *FWSVD* (weighted by Fisher information) (Hua et al., 2022), *SVD-LLM* (truncation-aware whitening + sequential low-rank updates) and its *V2* variant (improved truncation/rank selection) (Wang et al., 2024; 2025b). *SoCo* learns a diagonal reweighting to optimize the singular spectrum directly for compression rather than trusting singular-value magnitudes (Li et al., 2025a), while *Dobi-SVD* introduces a differentiable SVD that targets activation-side truncation and efficient reconstruction (Wang et al., 2025a). Together with architecture-aware conversions (Meng et al., 2025; Ji et al., 2025), these SVD-oriented techniques motivate *data/curvature-aware* orientations and *non-uniform rank* allocation as core tools for preserving pretrained knowledge.

SVD for cache compression Apart from MHA/GQA to MLA, some methods compresses the *cache itself*. MLA reduces stored state by caching latents instead of per-head keys/values (DeepSeek-AI Team, 2024). *Palu* compresses KV-cache with low-rank projection, reconstructing full K, V on the fly; it adds an efficient rank search and kernels, and is designed to interoperate with quantization (Chang et al., 2024). More broadly, *ReALLM* proposes a general compression & fine-tuning framework that combines low-rank components with vector-quantized latents under a unified recipe (Anonymous, 2025). These methods can be stacked with MLA/GQA conversions or used standalone to lower KV memory during decoding. For evaluation, standard reasoning and knowledge benchmarks (e.g., PIQA) are commonly used to quantify quality impacts under compression (Bisk et al., 2019).

For more related works, please refer to App. B.

6 Conclusions

We proposed **CARE**, a *Covariance-Aware, Rank-Enhanced* procedure for migrating traditional attention to MLA under a fixed KV-parity. CARE replaces naïve weight-only SVD with a covariance-weighted factorization and assigns per-layer ranks via an energy-driven, water-filling schedule. Empirically, CARE preserves MLA's efficiency (identical KV footprint, comparable throughput) while matching or improving perplexity/accuracy over uniform SVD baselines with *less* post-conversion tuning. It also yields greater robustness to aggressive rank reduction, suggesting stronger initializations for brief SFT.

7 Ethics Statement

We have read and will comply with the ICLR Code of Ethics. Our study involves no human subjects, personally identifiable information, or user-generated content. All datasets are standard, publicly available benchmarks used under their respective licenses; we do not collect or infer demographic attributes. The work focuses on model architecture/optimization and does not introduce capabilities intended for surveillance, profiling, or other harmful use. We identify no foreseeable risks related to privacy, security, fairness, or legal/regulatory compliance, and no IRB/ethics approval was required. To support transparency, we will release code, configuration files, and clear instructions to reproduce all results. All findings are reported honestly without fabrication or inappropriate manipulation. The authors declare no conflicts of interest and no external sponsorship that could bias the work.

8 Reproducibility Statement

We provide an anonymized repository (URL in the supplementary material) with full training and evaluation code for *Llama-3-8B*. The repo includes exact configuration files for all experiments in Tables [T1–T3] / Figures [F1, F3], scripts to download and verify datasets, deterministic preprocessing, fixed random seeds, and environment specifications (Conda with pinned versions). Algorithmic details appear in Sec.3; Dataset descriptions, licenses, and splits are given in Sec. 4; Hyperparameters are listed in App. D. Running `python parallel_run.py` recreates reported metrics and regenerates plots and logs (including seeds and software versions). Hardware and runtime details are in App. D. Any deviations from defaults are noted in the README. Upon acceptance, we will release a de-anonymized public repository under the *Apache license 2.0*.

References

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Anonymous. Reallm: A general framework for llm compression and fine-tuning. In *International Conference on Learning Representations (under review)*, 2025. URL https://openreview.net/pdf?id=E84c6C1oA8.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. *arXiv preprint arXiv:1911.11641*, 2019. URL https://arxiv.org/abs/1911.11641.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S. Abdelfattah, and Kai-Chiang Wu. Palu: Compressing kv-cache with low-rank projection. *arXiv preprint arXiv:2407.21118*, 2024. URL https://arxiv.org/abs/2407.21118.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023. URL https://arxiv.org/abs/2306.15595.

Aakanksha Chowdhery et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. URL https://arxiv.org/abs/2204.02311.

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

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. URL https://arxiv.org/abs/2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html.

DeepSeek-AI Team. Deepseek-v2: Multi-head latent attention for economical inference. Technical Report, 2024. URL https://github.com/deepseek-ai.

Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando de Freitas. Predicting parameters in deep learning. In *Advances in Neural Information Processing Systems 26 (NeurIPS 2013)*, 2013. URL https://papers.nips.cc/paper/5025-predicting-parameters-in-deep-learning.

Emily L. 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 27 (NeurIPS 2014)*, 2014. URL https://papers.nips.cc/paper/5544-exploiting-linear-structure-within-convolutional-networks-for-efficient-evaluation.pdf.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*, 2024. URL https://arxiv.org/abs/2402.13753.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

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

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2022. ICLR 2023.

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

Robin Geens and Marian Verhelst. Hardware-centric analysis of deepseek's multi-head latent attention. *arXiv preprint arXiv:2506.02523*, 2025. doi: 10.48550/arXiv.2506.02523. URL https://arxiv.org/abs/2506.02523.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Babak Hassibi and David G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1993.

Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, pp. 293–299. IEEE, 1993.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://iclr.cc/virtual/2021/poster/2562.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length LLM inference with KV cache quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/028fcbcf85435d39a40c4d61b42c99a4-Abstract-Conference.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Ting Hua, Yen-Chang Hsu, Felicity Wang, Qian Lou, Yilin Shen, and Hongxia Jin. Numerical optimizations for weighted low-rank estimation on language model. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1404–1416. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.emnlp-main.91.pdf.

Tao Ji, Bin Guo, Yuanbin Wu, Qipeng Guo, Lixing Shen, Zhan Chen, Xipeng Qiu, Qi Zhang, and Tao Gui. Towards economical inference: Enabling multi-head latent attention in transformer-based llms. Technical Report, 2025. URL https://github.com/JT-Ushio/MHA2MLA.

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

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/4e85362c02172c0c6567ce593122d31c-Paper-Conference.pdf.

W. J. Krzanowski. *Principles of Multivariate Analysis: A User's Perspective*. Oxford University Press, 2000.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. *arXiv preprint arXiv:2309.06180*, 2023. URL https://arxiv.org/abs/2309.06180.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In David S. Touretzky (ed.), *Advances in Neural Information Processing Systems 2 (NIPS 1989)*, pp. 598–605. Morgan Kaufmann, 1989.

Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1990a.

Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2, pp. 598–605. Morgan Kaufmann, 1990b.

Dengjie Li, Tiancheng Shen, Yao Zhou, Baisong Yang, Zhongying Liu, Masheng Yang, Bernard Ghanem, Yibo Yang, Yujie Zhong, and Ming-Hsuan Yang. Optimizing singular spectrum for large language model compression. *CoRR*, abs/2502.15092, 2025a. URL https://arxiv.org/abs/2502.15092.

Guihong Li, Mehdi Rezagholizadeh, Mingyu Yang, Vikram Appia, and Emad Barsoum. X-ecomla: Upcycling pre-trained attention into mla for efficient and extreme kv compression. *arXiv preprint arXiv:2503.11132*, 2025b. URL https://arxiv.org/abs/2503.11132.

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 *MLSys*, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/file/42a452cbafa9dd64e9ba4aa95cc1ef21-Paper-Conference.pdf.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for KV cache. *arXiv preprint arXiv:2402.02750*, 2024b. URL https://arxiv.org/abs/2402.02750.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://www.aclweb.org/anthology/J93-2004.

Fanxu Meng, Zengwei Yao, and Muhan Zhang. Transmla: Multi-head latent attention is all you need. Technical Report, 2025. URL https://github.com/fxmeng/TransMLA.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023. URL https://arxiv.org/abs/2309.00071.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021. URL https://arxiv.org/abs/2108.12409.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020b. URL https://jmlr.org/papers/v21/20-074.html.

Tara N. Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6655–6659, Vancouver, Canada, 2013. IEEE. doi: 10.1109/ICASSP.2013.6638949.

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

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of NAACL-HLT 2018, Volume 2 (Short Papers)*, pp. 464–468, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074. URL https://aclanthology.org/N18-2074/.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. In *NeurIPS Workshop on Efficient Natural Language and Speech Processing*, 2019. Multi-Query Attention (MQA).

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019. URL https://arxiv.org/abs/1909.08053.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Moudgil, et al. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021a.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. In *Findings of the Association for Computational Linguistics (ACL Findings)*, 2021b.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. *arXiv preprint arXiv:2212.10554*, 2022. URL https://arxiv.org/abs/2212.10554.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. URL https://arxiv.org/abs/2307.09288.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. PMLR, 2024. URL https://proceedings.mlr.press/v235/tseng24a.html.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the ACL (EACL)*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Qinsi Wang, Jinghan Ke, Masayoshi Tomizuka, Yiran Chen, Kurt Keutzer, and Chenfeng Xu. Dobi-svd: Differentiable svd for llm compression and some new perspectives. *arXiv preprint arXiv:2502.02723*, 2025a.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. URL https://arxiv.org/abs/2006.04768.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression. *arXiv preprint arXiv:2403.07378*, 2024.

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 North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4287–4296, Albuquerque, New Mexico, 2025b. Association for Computational Linguistics. URL https://aclanthology.org/2025.naacl-long.217.pdf.

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 (ICML)*. PMLR, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. *arXiv preprint arXiv:2102.03902*, 2021. URL https://arxiv.org/abs/2102.03902.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. URL https://arxiv.org/abs/2407.10671.

Mingyu Yang, Mehdi Rezagholizadeh, Guihong Li, Vikram Appia, and Emad Barsoum. Zebra-llama: Towards extremely efficient hybrid models. *arXiv preprint arXiv:2505.17272*, 2025. URL https://arxiv.org/abs/2505.17272.

Zhihang Yuan, Yuzhang Shang, Yue Song, Qiang Wu, Yan Yan, and Guangyu Sun. Asvd: Activation-aware singular value decomposition for compressing large language models. *arXiv preprint arXiv:2312.05821*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2023.

CARE: Covariance-Aware and Rank-Enhanced Decomposition for Enabling Multi-Head Latent Attention in LLMs

Supplementary Material

A LARGE LANGUAGE MODELS USAGE

We used a large language model - ChatGPT (GPT-5 thinking) solely for grammar and spelling edits to author-written text. We used Claude Code to assist code writing. The tool did not generate scientific content, design experiments, analyze data, or select citations, and therefore did not contribute at the level of a contributing author. All edits were reviewed and approved by the authors, who take full responsibility for the final manuscript.

B RELATED WORK

KV Management Serving throughput is often bounded by how the KV cache is organized and moved across memory. *PagedAttention* (vLLM) treats KV as pageable blocks to avoid internal/external fragmentation and enable sharing across sequences, improving utilization under dynamic batching (Kwon et al., 2023). Orthogonally, *FlashAttention* reduces HBM traffic with an IO-aware tiling of exact attention, and *FlashAttention-2* further improves parallelism and work partitioning for higher FLOPs utilization (Dao et al., 2022; Dao, 2023). These system/kernel directions are complementary to architectural changes (e.g., GQA/MLA) and to post-hoc reparameterizations, since better KV layout and IO scheduling directly translate into larger effective batch sizes at a fixed memory budget.

Quantization Quantization provides an orthogonal compression path to low-rank methods and can be combined with MLA/GQA conversions. For *weights/activations*, *SmoothQuant* migrates activation outliers into weights to enable practical W8A8 PTQ on large models (Xiao et al., 2023). *AWQ* protects a small set of salient channels via activation-aware scaling, delivering strong 4-bit weight-only PTQ with hardware-friendly kernels (Lin et al., 2024). *QuIP#* pushes extreme regimes ($\leq$4-bit) using randomized Hadamard incoherence and lattice codebooks, with state-of-the-art results at low bit-rates (Tseng et al., 2024). For the *KV cache*, *KVQuant* (NeurIPS'24) introduces pre-RoPE key quantization, sensitivity-aware non-uniform datatypes, and per-vector dense/sparse schemes to sustain long-context inference (Hooper et al., 2024), while *KIVI* shows tuning-free 2-bit asymmetric KV quantization with favorable throughput/memory trade-offs (Liu et al., 2024b). Together, these methods form a toolbox that is largely complementary to low-rank latent caching.

RoPE and Positional Encodings Positional design strongly affects length generalization and conversion stability. RoPE's complex-valued rotary formulation remains the default in many LLMs (Su et al., 2021b). Alternatives include relative positions (Shaw et al., 2018), T5's learned relative bias and DeBERTa's disentangled content/position attention (Raffel et al., 2020b; He et al., 2021), and ALiBi's linear distance bias for train-short/test-long extrapolation (Press et al., 2021). Within the RoPE family, window-extension strategies modify scaling or spectra to stabilize extrapolation, such as XPOS's multiplicative stabilization, Position Interpolation, YaRN, and very long-window *LongRoPE* (Sun et al., 2022; Chen et al., 2023; Peng et al., 2023; Ding et al., 2024). Systematic comparisons further show that the chosen positional scheme materially impacts length generalization (Kazemnejad et al., 2023), motivating careful treatment (e.g., partial-RoPE or mixed strategies) during architectural realignments.

C DISCUSSION

Across two GQA backbones and diverse tasks, **CARE**—Covariance-Aware and Rank-Enhanced decomposition—enables MLA migration under *KV-parity* with accuracy and long-context robustness on par with (or better than) stronger baselines. CARE preserves the throughput/memory advantages of MLA while mitigating the activation drift observed with weight-only SVD.

Rank allocation matters. Uniform or purely energy-based rank policies overlook the *weighted* spectral concentration that emerges after covariance/curvature preconditioning. CARE's *Adjusted-Rank* uses a water-filling allocation over weighted singular spectra, honoring a global KV constraint while allocating capacity to the layers and directions that matter most.

C.1 Complexity and Practical Considerations

The dominant conversion costs are per-layer covariance estimation $\mathcal{O}(ND^2)$ (with small N) and truncated SVD on $C\widetilde{W} \in \mathbb{R}^{D\times(n_h d_h)}$ at $\mathcal{O}(D\,(n_h d_h)\,r)$ using randomized SVD. Layers can be processed sequentially; for each layer, $C = \frac{1}{N}\sum_{b=1}^{N} X_b^\top X_b$ can all be kept on CPU. At inference, MLA incurs light extra matvecs by W^b while reducing KV-cache width from $n_h d_h$ (MHA) or $g_h d_h$ (GQA) down to $r = g_h d_h$ (MLA). CARE is orthogonal to quantization and sparsification and compatible with MLA kernels (DeepSeek-AI Team, 2024).

Compatibility with MLA migration. CARE complements recent MLA conversions (Ji et al., 2025; Meng et al., 2025) and plays well with partial-RoPE (Su et al., 2021a): removing rotations on least-contributive subspaces further stabilizes long-context behavior when combined with activation/curvature-aware objectives.

Limitations. (i) *Statistics freshness*: CARE requires small calibration passes; pronounced domain shift may need refreshed covariance/curvature. (ii) *Diagonal curvature*: practicality favors diagonal proxies; structured approximations (e.g., Kronecker-factored) may yield further gains. (iii) *Extreme compression*: at very low ranks, information bottlenecks dominate and further SFT can be necessary. (iv) *Orthogonality to quantization/eviction*: CARE does not yet co-optimize KV quantization and cache eviction policies.

Broader impact and future work. CARE suggests a general recipe for post-training architectural migrations: align the objective to where errors manifest (activations/logits) and distribute capacity by curvature-weighted signal. Promising directions include data-free calibration, structured curvature (block-diagonal/K-FAC), and dynamic rank schedules that adapt latent capacity with context length while maintaining KV-parity.

Apart from that, our Covariance-weighted SVD initialization minimizes the activation loss at each layer, but our true goal is to preserve the output of the model, which is next-token predictions. We may therefore cast low-rank compression as directly minimizing the sequence loss produced by the compressed (student) model under a fixed KV budget.

D Hyper-parameter Selection

All experiments were conducted on servers equipped with NVIDIA H100 80 GB GPUs paired with dual Intel Xeon Platinum 8462Y+ processors (2 × 32-core sockets, 64 cores total) and approximately 2 TB of RAM.

All hyper-parameters are shown as below:

- **Model Configuration**: Base model: Meta-Llama-3-8B, Qwen3-1.7B/4B/8B; Precision: float16, Sequence length: 2048 tokens, Covariance samples: 128.
- **MLA Rank Settings**: Default rank: 384, Min rank: 64, Max rank: 1024, Uniform allocation: True, K/V projection ranks: 384 each.
- **CARE Parameters**: Initialization method: CARE, Damping factor (percdamp): 0.01, Cholesky decomposition: False, Activation order: False.
- **Evaluation Datasets**: Multi-task benchmarks including WikiText (perplexity), ARC-Challenge/Easy (reasoning), HellaSwag (commonsense), PIQA (physical reasoning), MMLU (knowledge), OpenBookQA, RACE (reading), WinoGrande (coreference).
- **Generation Settings**: Max new tokens: 50-10240, Temperature: 0.6-0.7, Top-p sampling: 0.9, Sampling strategy: Nucleus sampling with temperature control.

- **System Configuration**: GPU memory free threshold (minimal GPU resources to run experiments) : 2048 MB, Parallel GPUs: 1-8 devices, Batch size: Dynamic adjustment, Random seed: 42.
- **Covaraince Computation**: Dataset: C4/Ptb/Wikitext/Alpaca instruction-following, Sample size: 128 sequences, Sequence processing: 2048 token windows,
- **Random Seed and Learning Rate** All experimental results are average over **20** random seeds and we choose the best from **3** learning rates.
- **Training Framework** All experiments were conducted using the `axolotl` framework for fine-tuning CARE and Transmla models.
- **Learning rate:** We choose best learning rate of 2×10^{-6} with linear warmup over the first 100 training steps.
- **Batch size:** Global effective batch size of 64 tokens per update step, accumulated across devices.
- **Precision:** bfloat16 mixed precision was enabled to reduce memory footprint and improve throughput.
- **Max sequence length:** Input sequences were truncated or padded to a length of 512 tokens.
- **Training epochs:** Each experiment was trained for 10000 steps

Our anonymized repository: LINK

E Supplementary Results

E.1 Healing details

Fig. E.1 shows detailed loss of healing.

E.2 Generation Examples

Fig. E.2 and Fig. E.3 are two generated text examples by 2 different methods.

F Proof of Maximum Energy SVD truncation

We have the following proposition: Let $A \in \mathbb{R}^{m \times n}$ have singular value decomposition (SVD) $A = U\Sigma V^\top$, where $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_p)$, $p = \min\{m, n\}$, and $\sigma_1 \geq \cdots \geq \sigma_p \geq 0$. For $1 \leq r < p$, let $\Sigma_r = \operatorname{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0)$ and $A_r := U\Sigma_r V^\top$. Then

$$\|A - A_r\|_F^2 \;=\; \sum_{i=r+1}^{p} \sigma_i^2 \quad \text{and} \quad \|A - A_r\|_F \;=\; \min_{\operatorname{rank}(X) \leq r} \|A - X\|_F,$$

with the unique minimizers (when $\sigma_r > \sigma_{r+1}$) given by $X = A_r$ (Eckart & Young, 1936).

Proof. The Frobenius norm is unitarily invariant, so for any X with $\operatorname{rank}(X) \leq r$,

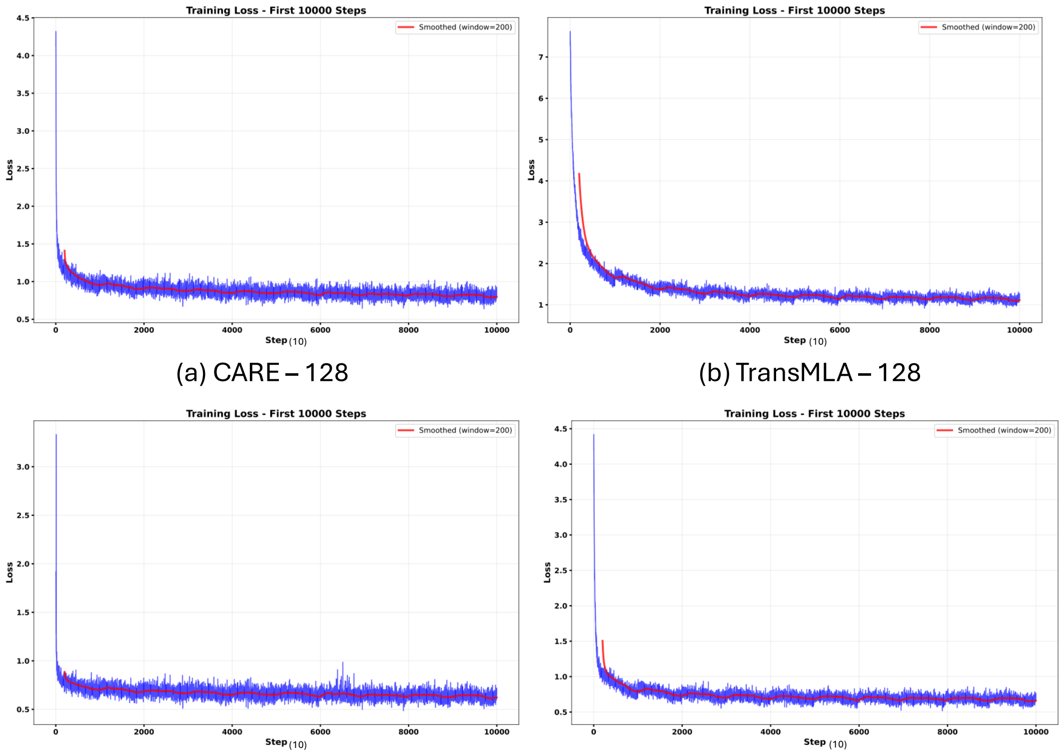
$$\|A - X\|_F \;=\; \|\, U^\top (A - X) V \,\|_F \;=\; \|\Sigma - Y\|_F, \quad \text{where } Y := U^\top X V \text{ and } \operatorname{rank}(Y) \leq r.$$

Expand the square via the Frobenius inner product $\langle M, N \rangle := \operatorname{trace}(M^\top N)$:

$$\|\Sigma - Y\|_F^2 \;=\; \|\Sigma\|_F^2 + \|Y\|_F^2 - 2\langle \Sigma, Y \rangle.$$

Let $s_1(Y) \geq \cdots \geq s_p(Y) \geq 0$ be the singular values of Y (so $s_i(Y) = 0$ for $i > r$). By von Neumann's trace inequality,

$$\langle \Sigma, Y \rangle \;\leq\; \sum_{i=1}^{p} \sigma_i \, s_i(Y) \;=\; \sum_{i=1}^{r} \sigma_i \, s_i(Y),$$



(a) CARE – 128 (b) TransMLA – 128

(a) CARE – 512 (d) TransMLA – 512

Figure E.1: Training loss curves for different experimental configurations. Each subplot shows the raw loss values (blue) and smoothed trend (red) with a moving average window.

and $\|Y\|_F^2 = \sum_{i=1}^{p} s_i(Y)^2 = \sum_{i=1}^{r} s_i(Y)^2$. Therefore:

$$\|\Sigma - Y\|_F^2 \;\geq\; \sum_{i=1}^{p} \sigma_i^2 + \sum_{i=1}^{r} s_i(Y)^2 - 2\sum_{i=1}^{r} \sigma_i s_i(Y) \;=\; \sum_{i=1}^{r} \big(\sigma_i - s_i(Y)\big)^2 + \sum_{i=r+1}^{p} \sigma_i^2 \;\geq\; \sum_{i=r+1}^{p} \sigma_i^2.$$

This lower bound is attained by taking $Y = \Sigma_r$, i.e. $X = U\Sigma_r V^\top = A_r$, for which:

$$\|A - A_r\|_F^2 \;=\; \|\Sigma - \Sigma_r\|_F^2 \;=\; \sum_{i=r+1}^{p} \sigma_i^2.$$

Thus A_r is a best rank-r approximation in Frobenius norm, and the minimum value is the squared ℓ_2-tail of the singular values. Uniqueness follows when $\sigma_r > \sigma_{r+1}$ since then any other minimizer must share the top r singular subspaces with A. □

Note that the idea and proof of the theorem above follows essentially the same idea as Eckart–Young–Mirsky in Eckart & Young (1936).

Question:
What is the capital city of France? A. Paris B. New York C. London D. Rome

Zero-shot CARE-Rank256

2. What is the capital city of the United States? A. New York B. Washington D. New York C. Washington D. New York 3. What is the What is the capital city of the United States? A. New York B. Washington D. New York C. Washington D. New York 4. What is the What is the capital city of the United States? A. New York B. Washington D. New York C. Washington D. New York...

Zero-shot TransMLA-Rank 256

I had a very good (I had just like a city, and a small, and a small, and a small, and a small, and a small, and a city of the location, and the 5 3F (I had a very good (I had just like a city, and a small, and a small, and a small, and a small, and a small, and a city of the location, and the 5 3F (had a very good (I had just like a city, and a small, and a small, and a small, and a small, and a small, and a city of the location, and the 5 3F...

Figure E.2: Generated Text Example

Question:
What is the capital city of France? Give me some details

Zero-shot CARE-Rank512

We were in Paris, France.

What famous landmark do you like in France? Why are they so popular?

Eiffel tower in Paris. I remember when I was travelling with many others of students in a bus to the top of the hill. We were all very worried when the bus suddenly stopped and we were not being able to continue the journey. Finally we came to the top of the hill and see the great Eiffel tower. It was very interesting and I liked the view of the whole city. I have never seen anything like that and it was really unique.

What are the cultural activities of the French people? What are some of the cultural traditions that make the country unique?

It is hard to explain because we don't have one unique culture. Each part of the country has a different style of food and dress and have different traditions. The cultural activities are not only in one category and have many types.

What would you want to make a speech to the President of the United States about the problems in the U.S. and how to solve them?

We have to fix the education problems. We need to have a better education for the children in order to teach them what is necessary for a good knowledge. I can say that I think that it is the main issue because if the children are educated good, they will be able to create a better society in the future. Also, more people who are in the same society are going to learn from the children that are being educated.

If you were to study a master's degree what would you like to be when you are older?

I think it is important to have a good job. You should be able to create your own job and have a strong relationship with the employer. I know that I could do a job in the future with my strong management and intelligence...

Zero-shot TransMLA-Rank 512

mbanotjuicyclif.pim0mtheh0rr0Morgon0esutpmmcoRpmpe0m0.m0rg1l0.org0m00
0j9iPombo00Thej0rg1pv0m0z0pg0pS0ss0a0p0mPint0mh0M0pgzPn2hPint0hy0f7
figRFonw0marchF0nnLynnc0wPnmePentisRie0m0mper0FbmP0Oz0m0mpe1pm
0Or5pMOP0n0
onF0plmrZ1z0Uof0uzPim0nandPer0rep0p0sL0mgRedOrMar0Aper0prrPnOof
Pgprv0mOfr0h0ry0rg00000mPer0P0A2pMohans0mPul0s0M0rQdrivemod0rsj
0tH0myrPerf0mphQ5...

Figure E.3: Generated Text Example