FREEZE THE DECODER, HEAL THE ENCODER: PARAMETER-EFFICIENT ADAPTATION FOR SVD-BASED KV-CACHE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

The KV cache dominates inference memory, especially for vision–language models, which must cache image and text tokens alike. A pretrained model can be retrofitted with a compact, multi-head-latent-attention-style cache by SVD-factorizing its key/value projections into a down-projection ("encoder") and an up-projection ("decoder"), followed by a short fine-tune ("healing") that recovers the accuracy lost to truncation. Existing conversions heal both factors, without asking whether both need it. We freeze the decoder at its SVD initialization and heal only the encoder, which trains 1/3 of the new parameters. This keeps keys and values in the original top-r output subspace, but leaves the model free to choose what the cache stores. Under per-arm learning-rate tuning and three seeds, encoder-only healing is competitive with decoder-only and full healing on Qwen2.5-VL-3B-Instruct (MME, TextVQA) and on average accuracy over six zero-shot text tasks across two 7B backbones, including a setting with 3.5–6.3 points of headroom left. No difference is statistically significant, and equivalence tests bound any difference to within 1.2–5.4% of the reference score per metric; with three seeds, smaller deficits cannot be ruled out. Encoder-only healing uses $3\times$ fewer trainable parameters and $3\times$ less optimizer-state memory (0.57 vs. 1.72 GiB), producing an identical served model. We also find that comparing healing modes at one shared learning rate makes encoder-only healing look far better than it is, with gaps that pass a 3-seed t-test (t up to 8.78) but vanish under per-arm tuning, a pitfall for comparing any fine-tuning recipes of unequal size.

1 INTRODUCTION

The KV cache dominates inference memory at long context, and vision–language models pay this cost for image and text tokens alike. Multi-head latent attention (MLA) (DeepSeek-AI, 2024) shrinks the cache by storing one low-dimensional latent per token instead of full keys and values. Training MLA from scratch is expensive, so recent work retrofits it onto already-pretrained models (Meng et al., 2025; Ji et al., 2025; Zhou et al., 2026). Each layer's key/value weights are SVD-factorized into an *encoder*, which projects the hidden state into the cached latent, and a *decoder*, which maps the latent back to keys and values. Truncating the SVD loses accuracy, so a short fine-tune, called *healing*, follows.

Existing conversions heal both factors as a fixed part of the recipe. That is a costly default. The decoder holds $2/3$ of the new parameters, and with AdamW every trained parameter also carries two optimizer moment buffers. If the decoder does not need training, healing it wastes memory in exactly the setting, adapting a large multimodal model, where memory is scarce.

We find empirically that, in the settings we test, healing the decoder buys no detectable accuracy. We freeze the decoder at its SVD initialization and heal only the encoder (Figure 1). The encoder is the lossy step, since it decides which d_c directions of each hidden state the cache keeps. Freezing the decoder restricts keys and values to the original top-d_c output subspace (Eq. 4), but leaves that choice free. Whether the restriction costs accuracy is an empirical question; in our experiments, any cost is too small to detect.

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.

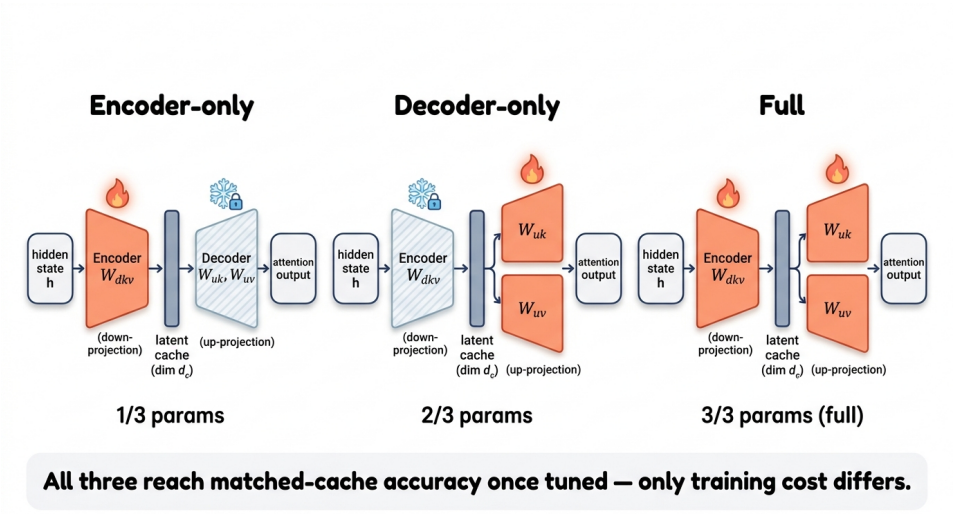

Figure 1: **Freeze the decoder, heal the encoder.** Orange factors are trained and blue hatched factors are frozen at their SVD initialization (Eq. 2). Encoder-only healing (ours) trains $1/3$ of the new parameters, decoder-only $2/3$, and full healing, the default in prior conversions, all of them. Once each is tuned to its own learning rate, we detect no accuracy difference among them.

Encoder-only healing is competitive with decoder-only and full healing on Qwen2.5-VL-3B-Instruct (MME and TextVQA, 98–100% of the uncompressed model) and on average zero-shot accuracy over six text tasks with OpenLLaMA-7B-v2 and Qwen2.5-7B. In the text setting, every arm leaves 3.5–6.3 points of headroom and all three recover a similar fraction of it, so the result is not only a ceiling effect. No pairwise difference is significant, and two one-sided tests bound every difference to within 1.2–5.4% of the reference score (0.59 and 1.66 points on the two text backbones). These bounds are wide because we have only three seeds per arm: our evidence shows that encoder-only healing is competitive, not that it is interchangeable with full healing. Encoder-only healing does this with $3\times$ fewer trainable parameters and $3\times$ less optimizer-state memory than full healing, and yields an identical served model.

Establishing even this required care. At one shared learning rate, encoder-only healing appears to beat both alternatives by large, 3-seed-significant margins. With per-arm tuning, the gap vanishes: the shared rate suited the smallest arm and destabilized the larger ones (§5.3). We therefore report only per-arm-tuned comparisons, and document the pitfall for others comparing fine-tuning recipes of unequal size.

Contributions.

1. **Encoder-only healing** (§3): a parameter-efficient healing recipe for SVD-based KV-cache conversion that freezes the decoder at its SVD initialization and trains only the encoder, with an exact characterization of the maps it can reach.
2. **Evidence that it is competitive across modalities** (§5.1): under per-arm tuning and 3 seeds, encoder-only healing shows no significant difference from decoder-only and full healing on a vision–language model and on two text backbones, including a setting with headroom left, with confidence intervals and equivalence bounds on every difference.
3. **A measured memory saving** (§5.2): $3\times$ fewer trainable parameters and $3\times$ less AdamW optimizer-state memory than full healing (0.57 vs. 1.72 GiB).
4. **An evaluation pitfall** (§5.3): a shared learning rate makes the smallest arm look far better, and multi-seed testing does not catch it, because the rate inflates the larger arms' variance as well as lowering their means.

2 RELATED WORK

Post-hoc MLA conversion. DeepSeek-V2 (DeepSeek-AI, 2024) trains MLA from scratch. TransMLA (Meng et al., 2025) and MHA2MLA (Ji et al., 2025) convert pretrained models to an MLA-style cache via SVD followed by healing, and CARE (Zhou et al., 2026) adds per-layer rank allocation and activation-whitened initialization. *Gap:* all three heal both factors as a fixed step, so none can tell whether the decoder needs training at all. We isolate the two factors and show that only the encoder does.

Improving the SVD step itself. Other work improves truncation or calibration rather than healing: SVD-LLM (Wang et al., 2025) and Palu (Chang et al., 2025a) use truncation-aware or low-rank KV projections; xKV (Chang et al., 2025b) shares bases across layers; KQ-SVD (Lesens et al., 2025) and Zero Sum SVD (Abbasi et al., 2026) target attention fidelity or loss sensitivity; Eigen Attention (Saxena et al., 2024), ReCalKV (Yan et al., 2025), and LoRC (Zhang et al., 2024) refine calibration, head ordering, and rank allocation; STAR-KV (Bhatnagar et al., 2026) learns the rank per attention head and block with differentiable soft thresholding, and factorizes keys and values differently according to their sensitivity; A^3 (Wong et al., 2025) and Benfenati et al. (2026) give analytical or manifold-constrained approximations; RAP (Xin et al., 2026) aligns truncation with RoPE. KV-CoRE (Chen et al., 2026) benchmarks compressibility, and MosaicKV (Qiang et al., 2026) and transform coding (Staniszewski et al., 2025) target serving-time storage. These methods change the factorization; encoder-only healing concerns the fine-tune that follows. It applies directly to any method that produces a fixed encoder–decoder split, but with adaptive per-head ranks, as in STAR-KV, each head retains a different subspace, and whether freezing the decoder still suffices is untested (§6).

Asymmetric low-rank factors. For LoRA adapters $\Delta W = BA$, LoRA-FA (Zhang et al., 2023) freezes the randomly initialized down-projection A and trains only B; Zhu et al. (2024) analyze the different roles of A and B. Freezing one factor is LoRA-FA's idea, not ours. *Gap:* in LoRA, one factor starts random and the other at zero. In SVD conversion, both come from a truncated SVD of pretrained weights, so the frozen factor fixes a meaningful subspace. We find that in this setting the up-projection can be frozen, the reverse of LoRA-FA's choice.

Other axes of KV-cache compression. Grouped-query attention (Ainslie et al., 2023) and its variants (Chen et al., 2024b; Khan et al., 2024; Meng, 2026; Jin et al., 2024; Chen et al., 2024a) reduce the number of K/V heads; MLKV (Zuhri et al., 2024) shares them across layers; MLRA (Liu et al., 2026) restructures the latent for tensor parallelism; Native Sparse Attention (Yuan et al., 2025) sparsifies attended tokens; VideoMLA (Yesiltepe et al., 2026) brings latent caching to video diffusion. Training-free methods also factorize the cached tensors themselves rather than the projection weights: JoLT (Krishnan & Schulz, 2026) applies a partial Tucker decomposition along the token and feature modes with a quantized residual, allocating ranks and bit-widths under one storage budget, and PuzzleKV (Wang et al., 2026) decomposes fixed-length pages of each head's cache independently during decoding. These methods target near-lossless serving-time compression and need no fine-tuning, so the question of which factor to heal does not arise for them. Our scope is different: we reduce the *adaptation* memory of projection-based latent caches, which require a short fine-tune after conversion. All of these axes are complementary to ours.

Fair comparison of fine-tuning recipes. Biderman et al. (2024) show that conclusions about LoRA versus full fine-tuning depend on learning-rate choices for methods of different capacity. We add a concrete case in which a shared rate also inflated variance, so that a spurious gap passed a multi-seed test.

3 METHOD: FREEZE THE DECODER, HEAL THE ENCODER

Why convert, and why heal. A standard attention layer with n_h heads of width d_h caches a key and a value per token and head, of total width $d_{kv} = n_h d_h$ each. MLA instead caches a single latent of width $d_c \ll 2d_{kv}$, which cuts cache bytes per token by a factor of $2d_{kv}/d_c$. To obtain an MLA cache without pretraining from scratch, we factorize the existing weights. The factorization discards

information, so a short fine-tune is needed to recover accuracy. The question is which weights that fine-tune should touch.

SVD conversion. Let d be the model's hidden width and $W_k, W_v \in \mathbb{R}^{d_{kv} \times d}$ a layer's key and value projections, written per attention head. We stack them and take the SVD

$$M = \begin{bmatrix} W_k \\ W_v \end{bmatrix} = U \Sigma V^\top, \qquad M \in \mathbb{R}^{2d_{kv} \times d}. \tag{1}$$

Keeping the top $r = d_c$ singular values and splitting Σ_r evenly between the two factors gives

$$W_{dkv} = \Sigma_r^{1/2} V_r^\top \in \mathbb{R}^{d_c \times d}, \qquad \begin{bmatrix} W_{uk} \\ W_{uv} \end{bmatrix} = U_r \Sigma_r^{1/2} \in \mathbb{R}^{2d_{kv} \times d_c}. \tag{2}$$

The encoder W_{dkv} maps each hidden state h_t to the cached latent, and the decoder W_{uk}, W_{uv} maps the latent back to keys and values:

$$c_t = W_{dkv} h_t, \qquad k_t = W_{uk} c_t, \qquad v_t = W_{uv} c_t. \tag{3}$$

Only $c_t \in \mathbb{R}^{d_c}$ is cached. At full rank, Eq. (2) reproduces M exactly; we measure a relative reconstruction error of 2×10^{-4}, i.e. floating-point noise. At $r = d_c$ it drops the lowest-energy singular directions of M, and the model must be healed.

What freezing a factor restricts. Freezing one factor restricts which layer maps healing can reach. Assume $\sigma_r > 0$, so $\Sigma_r^{1/2}$ is invertible. With the decoder frozen at $D_0 = U_r \Sigma_r^{1/2}$ and the encoder E free, the reachable key/value maps are

$$\mathcal{F}_{\text{enc}} = \{D_0 E : E \in \mathbb{R}^{d_c \times d}\} = \{X \in \mathbb{R}^{2d_{kv} \times d} : \mathrm{col}(X) \subseteq \mathrm{col}(U_r)\}. \tag{4}$$

Encoder-only healing can therefore realize *any* rank-$\leq r$ map whose outputs lie in the original top-r output subspace of M. It is free to change which directions of h_t the cache keeps, and how they are combined, but not the subspace in which keys and values live. The mirror case, decoder-only healing, reaches $\mathcal{F}_{\text{dec}} = \{X : \mathrm{row}(X) \subseteq \mathrm{row}(V_r^\top)\}$: the output map is free, but the cache is fixed to store the projection of h_t onto the original top-r input directions. Full healing reaches every rank-$\leq r$ map and can move both subspaces. Two remarks follow. First, because $\mathcal{F}_{\text{enc}}$ depends only on $\mathrm{col}(D_0)$, it is unchanged by the gauge freedom $D_0 \mapsto D_0 R$, $E \mapsto R^{-1} E$ for invertible R; what matters is the subspace, not the particular factor (the parameterization can still affect optimization). Second, the SVD makes the *product* $D_0 E_0$ the Frobenius-optimal rank-r approximation of M at initialization. It says nothing about whether $\mathrm{col}(U_r)$ is a good output subspace for the downstream loss after healing.

Encoder-only healing is an empirical bet. We therefore do not claim that encoder-only healing is optimal. It is a bet that the original top-r output subspace is already sufficient, and that the accuracy lost to truncation can be recovered by choosing better what the cache stores. The bet is attractive because it trains the fewest parameters, and it is falsifiable: if the downstream task needed a different output subspace, full healing would beat it. Section 5.1 tests exactly this.

Encoder-only healing and the alternatives. The encoder has $d \cdot d_c$ parameters and the decoder has $2d_{kv} \cdot d_c$. Every checkpoint we study has a square query projection, $n_h d_h = d$, so these are exactly $1/3$ and $2/3$ of the total. We compare:

- **Encoder-only (ours)** trains W_{dkv} and freezes the decoder at Eq. (2): $1/3$ of the new parameters.
- **Decoder-only** trains W_{uk}, W_{uv} and freezes the encoder: $2/3$.
- **Full** trains both, as in TransMLA, MHA2MLA, and CARE (Meng et al., 2025; Ji et al., 2025; Zhou et al., 2026).
- **0-step** applies no healing and serves as a floor.

All modes produce the same architecture, cache size, and decode speed. They differ only in training cost.

Learning rate	Arm	MME Perception	MME Cognition	TextVQA EM
3×10^{-5} (tuned)	encoder-only (ours)	**1526.3 ± 4.0**	**625.6 ± 4.2**	0.8012 ± 0.0018
3×10^{-5} (tuned)	decoder-only	1515.3 ± 18.5	622.1 ± 13.8	0.8006 ± 0.0044
3×10^{-5} (tuned)	full	1509.8 ± 8.5	619.4 ± 16.6	**0.8025 ± 0.0055**
10^{-4} (shared)	encoder-only (ours)	**1539.1 ± 11.7**	**634.3 ± 6.2**	**0.7981 ± 0.0039**
10^{-4} (shared)	decoder-only	1392.7 ± 48.8	457.3 ± 97.8	0.6698 ± 0.1325
10^{-4} (shared)	full	1404.7 ± 23.8	538.9 ± 22.3	0.7498 ± 0.0201

Table 1: Qwen2.5-VL-3B-Instruct, $d_c = 512$, 3 seeds per arm, mean ± sample std. *Top (main result):* each arm at its tuned learning rate; no difference is significant under Welch's test (Table 2). *Bottom:* all arms at one shared rate, discussed in §5.3. Bold marks the best arm per metric within a block. Uncompressed baseline: 1534.4 / 616.4 / 0.8028.

4 Experimental setup

We ask four questions:

- **RQ1** (§5.1): How close is encoder-only healing to decoder-only and full healing on a vision–language model?
- **RQ2** (§5.1): Does the result hold on text, where accuracy is not near its ceiling?
- **RQ3** (§5.2): What does encoder-only healing save?
- **RQ4** (§5.3): How does the learning-rate protocol affect the comparison?

Vision–language setting. We convert Qwen2.5-VL-3B-Instruct at $d_c = 512$ and heal only the language backbone's self-attention; the vision tower and cross-modal projector stay frozen. The healing corpus is WikiText-2 plus COCO-Caption2017 for 300 steps. We evaluate on the full MME suite (2374 documents, all 14 categories; Perception and Cognition scores) and on TextVQA (exact match).

Text setting. We convert OpenLLaMA-7B-v2 (MHA) and Qwen2.5-7B (GQA-4) at $d_c = 512$ and heal on WikiText-2 for 6000 steps (∼12.3M tokens) and 2500 steps (∼5.1M tokens), respectively. We report average 0-shot accuracy on MMLU, ARC-C, PIQA, HellaSwag, OpenBookQA, and Winogrande.

Protocols and baselines. We compare encoder-only healing against decoder-only and full healing, the 0-step floor, and the uncompressed model. Every main configuration uses 3 seeds, and we report mean ± sample standard deviation. Our main protocol is *tuned*: the learning rate is swept separately for each arm and each arm uses its best; on the vision–language model, all three arms select 3×10^{-5}. For RQ4 we also run a *shared* protocol, which trains all arms at one learning rate (10^{-4}). Because arms can have very different variances, we compare them with Welch's unequal-variance t-test over seeds. For each difference we report a 95% Welch confidence interval and the smallest symmetric margin δ for which two one-sided tests (TOST, $\alpha = 0.05$, i.e. the 90% interval lies within $\pm\delta$) establish equivalence. We did not fix an equivalence margin in advance, so we report the margin the data support rather than a pass/fail verdict. Optimizer-state memory is measured for AdamW.

5 Results

5.1 RQ1 and RQ2: Encoder-only healing is competitive with full healing

Takeaway 1: encoder-only healing is competitive on the vision–language model. In the tuned block of Table 1, encoder-only healing has the highest MME means and full healing the highest TextVQA mean. None of the nine pairwise differences is significant under Welch's test. The smallest p-value is 0.060, for encoder-only *above* full healing on MME Perception ($t = 3.04$); no difference in favor of either alternative comes close (all $|t| \leq 0.39$). Table 2 gives the confidence intervals: any encoder-only deficit is bounded to 2.7% of the baseline on MME Perception, 5.4% on MME

Setting	Metric	enc − full [95% CI]	enc − dec [95% CI]	TOST δ
Qwen2.5-VL-3B	MME Perc.	$+16.5$ $[-1.3, 34.3]$	$+11.0$ $[-32.4, 54.4]$	41.1 (2.7%)
Qwen2.5-VL-3B	MME Cog.	$+6.2$ $[-32.0, 44.4]$	$+3.5$ $[-27.5, 34.5]$	33.0 (5.4%)
Qwen2.5-VL-3B	TextVQA	-0.0013 $[-0.0135, 0.0109]$	$+0.0006$ $[-0.0088, 0.0100]$	0.010 (1.2%)
Qwen2.5-7B	avg. acc.	-0.28 $[-2.28, 1.72]$	-0.33 $[-2.21, 1.55]$	1.66 (2.6%)
OpenLLaMA-7B	avg. acc.	-0.16 $[-0.63, 0.31]$	-0.25 $[-0.70, 0.20]$	0.59 (1.2%)

Table 2: Differences between encoder-only healing and each alternative under per-arm tuning (3 seeds per arm), with 95% Welch confidence intervals. The TOST margin is the smallest δ for which two one-sided Welch tests ($\alpha = 0.05$) establish equivalence to *both* alternatives; in parentheses, relative to the uncompressed baseline (vision–language) or to the full-healing mean (text). Margins were not fixed in advance.

Model ($d_c = 512$)	0-step	decoder-only	full	encoder-only (ours)
Qwen2.5-7B	35.72	63.68 ± 0.24 (103M)	63.63 ± 0.13 (154M)	63.35 ± 0.83 (51M)
OpenLLaMA-7B	33.93	50.08 ± 0.13 (103M)	49.99 ± 0.09 (154M)	49.83 ± 0.22 (51M)

Table 3: Average 0-shot accuracy over six text tasks at matched KV-cache size; trainable parameters in parentheses. 3-seed mean $\pm$ sample std, each arm at its own best learning rate. No pairwise difference is significant under Welch's test (smallest $p = 0.18$); see Table 2 for confidence intervals.

Cognition, and 1.2% on TextVQA. All three arms reach 98–100% of the uncompressed model (1534.4 / 616.4 / 0.8028). Because this is close to the ceiling, there is little room for any arm to fall behind; the text setting addresses this.

Takeaway 2: encoder-only healing is competitive on text, with headroom left. Table 3 shows the tuned text comparison. Here encoder-only has the lowest mean on both backbones, by 0.28–0.33 points on Qwen2.5-7B and 0.16–0.25 points on OpenLLaMA-7B, and it is the noisiest arm on Qwen2.5-7B ($\pm$0.83 vs. $\pm$0.24 and $\pm$0.13). No difference is significant (smallest Welch $p = 0.18$). Equivalence holds only within $\pm$0.59 points on OpenLLaMA-7B and $\pm$1.66 points on Qwen2.5-7B (Table 2), so a small deficit of a few tenths of a point is consistent with our data. Every arm leaves 6.3 (OpenLLaMA) and 3.5 (Qwen2.5-7B) points of headroom to the uncompressed model, and all three recover the same fraction of the gap (72–73% and 88–89%). The comparison is therefore not only a ceiling effect. Together, the two settings show that, at $d_c = 512$, any accuracy cost of restricting outputs to $\mathrm{col}(U_r)$ is too small to detect with three seeds, near or away from the ceiling, on two modalities and on both MHA and GQA backbones. Establishing equivalence at a tighter, task-relevant margin would need more seeds. Because we report the six-task average, this result does not show that encoder-only healing is competitive on every individual task; an average could hide a regression on one task offset by gains on others.

5.2 RQ3: What encoder-only healing saves

Takeaway 3: the saving is training memory. Encoder-only healing trains 3$\times$ fewer parameters (51M vs. 154M on the 7B text backbones; Table 3). Because AdamW keeps two moment buffers per trainable parameter, it also uses 3$\times$ less optimizer-state memory than full healing (0.57 vs. 1.72 GiB, measured). On a memory-limited GPU, this can decide whether a larger batch fits, or whether healing fits at all without offloading. The served model is unchanged: all modes produce the same architecture, cache size, and decode speed.

Takeaway 4: the saving is not speed or stability. Wall-clock time barely changes (27.3 vs. 28.3 minutes on the text setting, $\sim$3.5%), because gradients must still flow back through the frozen decoder to reach the encoder. Encoder-only healing is also not more stable across seeds: on text it is noisier than full healing at its own optimum ($\pm$0.83 vs. $\pm$0.13), though comparably stable on the vision–language model once tuned.

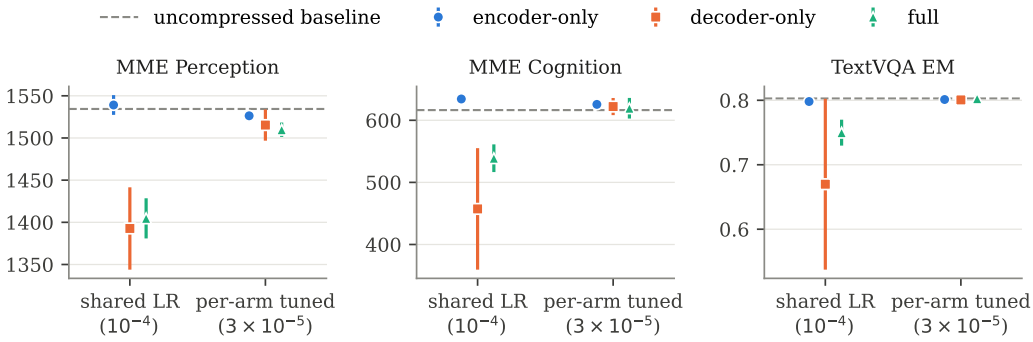

Figure 2: **A shared learning rate exaggerates encoder-only healing's advantage.** Qwen2.5-VL-3B-Instruct at $d_c = 512$; mean $\pm$ std over 3 seeds per arm (values in Table 1). *Left of each panel:* at one shared rate, encoder-only appears far better, and the larger arms are both lower and noisier. *Right:* with each arm at its own tuned rate, all three arms are close to the uncompressed baseline and no difference is significant.

5.3 RQ4: Why per-arm tuning matters

Our first comparison of the healing modes used one shared learning rate, and it suggested that encoder-only healing is not merely cheaper but far more accurate. That conclusion was wrong. Figure 2 and the bottom block of Table 1 show why.

Takeaway 5: a shared rate produces a large, significant, and spurious gap. At 10^{-4}, encoder-only leads decoder-only by 146 MME Perception points, 177 MME Cognition points, and 0.128 TextVQA EM, with t up to 8.78 (Welch $p = 0.003$). The gap disappears under per-arm tuning, and a further single-seed run at 10^{-5} agrees: the margins are $+21.2$ / -10.0 / -0.0086, small and inconsistent in sign. The shared rate also inflated the larger arms' variance: decoder-only's standard deviation falls from $\pm$97.8 to $\pm$13.8 on MME Cognition and from $\pm$0.1325 to $\pm$0.0044 on TextVQA, a $\sim$30$\times$ reduction, once it gets its own rate. Encoder-only, for which 10^{-4} is close to a good rate, is tight in both blocks. A tight arm near its optimum against noisy arms far from theirs is exactly the pattern a small-n test rewards, so **multi-seed replication alone did not catch the problem.**

Takeaway 6: the same happens on text. The shared protocol produced a similar encoder-only advantage on text, along with a plausible-sounding mechanism. At a shared 10^{-4} on OpenLLaMA-7B at $d_c = 2048$, the *unhealed* 0-step model (53.09) outscored full healing (48.37), suggesting that training the decoder is harmful. With per-arm rates, full and decoder-only healing both beat the 0-step model (54.78 and 54.60). Training the decoder is not harmful; it is unnecessary, and it needs a smaller learning rate than the shared protocol gave it.

Recommendations. When comparing fine-tuning recipes that train different numbers of parameters: (i) tune the learning rate separately for each arm; (ii) report per-arm variance and compare it across learning rates, since a disadvantaged arm whose variance collapses at its own rate signals this problem; (iii) check that a claimed advantage holds at more than one rate per arm.

5.4 Exploratory results

The two results below are single-seed. We report them to guide follow-up work, not at the standard of §5.1–§5.3.

Whitened SVD initialization. SVD-LLM, Palu, and CARE (Wang et al., 2025; Chang et al., 2025a; Zhou et al., 2026) calibrate the factorization with activation statistics ("whitened" SVD) and report better post-truncation accuracy. We replaced only the initialization of encoder-only healing with a WikiText-2-calibrated whitened SVD. It *underperformed* plain SVD by 7–20 points on both backbones and at every ratio tested (Table 4). Both initializations pass the full-rank reconstruction

Model	d_c	Plain init	Whitened init
OpenLLaMA-7B	2048 (4×)	54.18	42.39
OpenLLaMA-7B	512 (16×)	49.96	42.74
Qwen2.5-7B	512†	63.34	43.48

Table 4: Encoder-only healing with plain vs. whitened SVD initialization, single seed. †Qwen2.5-7B's native attention width is below 2048, so $d_c = 512$ is a $2\times$ compression of its native width.

check, so this is not an implementation error. We do not read it as refuting prior work. Our calibration set is narrower, our 0-shot suite differs from the perplexity-style evaluations those papers emphasize, and Qwen2.5-7B's activation covariance is known to be poorly conditioned for whitening because of massive activations.

Other compression ratios. An earlier pilot ran all three arms at $d_c \in \{256, 512, 1024\}$ on Qwen2.5-VL-3B-Instruct (Appendix A). It used one shared learning rate and one seed, which is the protocol §5.3 shows to be unreliable. Its $d_c = 512$ rows reproduce the spurious shared-rate advantage, so we presume the similar encoder-only lead at $d_c = 1024$ has the same cause until it is rerun with per-arm tuning. At $d_c = 256$, every arm collapses far below baseline within 300 steps, so that ratio is uninformative. The pilot does not show whether the arms separate at stronger compression; it identifies $d_c = 1024$ and a longer-budget $d_c = 256$ as the next configurations to test.

6 Limitations and conclusion

Limitations. With three seeds per arm, our comparisons can only bound differences to the margins in Table 2, not establish equivalence at a tighter, task-relevant margin, and we report six-task averages rather than per-task scores and paired per-seed differences, so we cannot rule out task-specific regressions hidden by the average. The vision–language result uses one compression ratio ($d_c = 512$), one 3B backbone, and one healing corpus and budget. Because every tuned arm is near the uncompressed baseline there, it cannot rule out a difference at stronger compression. The text result has headroom but also uses one ratio. We document the confound for one family of recipes; we have not tested how generally it appears in other PEFT comparisons, nor isolated its mechanism. Encoder-only healing lowers only the cost of producing a compressed model; the served model's quality, speed, and memory are unchanged, so it complements methods that improve the compressed model itself. We fix $r = d_c$ globally, as in Meng et al. (2025) and Ji et al. (2025); with adaptive or data-dependent rank (Zhou et al., 2026; Chen et al., 2025; Bhatnagar et al., 2026), each head or block retains a different spectrum, so the output subspace a frozen decoder imposes (Eq. 4) could bind more tightly in low-rank heads, and the asymmetry may differ. We do not compare against function-aware or calibration-aware KV compression methods. Finally, we have not measured how far full healing moves the decoder away from $\mathrm{col}(U_r)$. Principal angles between the initial and healed decoder subspaces, and the healed decoder's singular spectrum, would show whether the restriction in Eq. (4) is ever close to binding, and could predict when encoder-only healing will fall behind full healing, for example at stronger compression.

Conclusion. In our experiments, retrofitting a low-rank KV cache onto a pretrained model did not require healing the decoder. Freezing it at its SVD initialization restricts keys and values to the original top-r output subspace, yet healing only the encoder is competitive with decoder-only and full healing on a vision–language model and on two text backbones, with no significant difference and any deficit bounded to 1.2–5.4% of the reference score, with $3\times$ fewer trainable parameters and $3\times$ less optimizer-state memory. Showing this required tuning each arm's learning rate separately, since a shared rate made encoder-only healing look far better than it is, with gaps that passed a multi-seed test. Next steps are to rerun the multi-ratio comparison with per-arm tuning, to combine encoder-only healing with head-grouping compression, and to give the trained encoder its own low-rank parameterization, in the spirit of LoRA-FA, to push the savings beyond $3\times$.

Reproducibility Statement

Section 3 specifies the SVD conversion (Eqs. 1–3), the function class of each healing mode (Eq. 4), and the three healing modes. Section 4 specifies the models, healing corpora, step counts, compression ratio, evaluation suites, seeds, and learning-rate protocols. All models, datasets, and benchmarks are public. We will release the conversion, healing, and evaluation code, together with the per-arm learning-rate sweeps and per-seed results, with the camera-ready version.

Ethics Statement

This work studies which weights need fine-tuning after KV-cache compression, using public model checkpoints and public, non-sensitive benchmarks (MME, TextVQA, MMLU, and other 0-shot QA and reasoning tasks). It involves no human subjects. Reducing the memory needed to adapt compressed models is a modest resource-reduction contribution; the saving does not extend to inference cost.

AI Use Statement

In this work, we used generative AI tools to help refine hypotheses and the framing of the research question, and to implement methods, including creating and editing the code for SVD conversion, healing, and evaluation. We have not used generative AI tools to design the experimental methodology or to interpret results, and the remaining required-disclosure tasks (synthetic data, theoretical models, mathematical claims or proofs, translation, dataset cleaning, and qualitative analysis) are not applicable to this work. Additionally, we used generative AI tools to draft, restructure, and edit the manuscript text. We have reviewed all AI-assisted work: AI-assisted code was checked by the authors before use, and every number reported in the paper was checked against saved experimental results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

Ali Abbasi, Chayne Thrash, Haoran Qin, Shansita Sharma, Sepehr Seifi, and Soheil Kolouri. Zero sum SVD: Balancing loss sensitivity for low rank LLM compression, 2026.

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

Luca Benfenati et al. Don't be so stief! learning KV cache low-rank approximation over the Stiefel manifold, 2026.

Priyansh Bhatnagar, Ashkan Moradifirouzabadi, Se-Hyun Yang, SeungJae Lee, Jungwook Choi, and Mingu Kang. STAR-KV: Low-rank KV cache compression via soft thresholding for adaptive rank control, 2026.

Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Kai-Chiang Wu, and Mohamed S. Abdelfattah. Palu: Compressing KV-cache with low-rank projection. In *International Conference on Learning Representations*, 2025a.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Hung-Yueh Chiang, Yash Akhauri, Xilai Dai, Huiqiang Jiang, Yucheng Li, Luis Ceze, Kai-Chiang Wu, and Mohamed S. Abdelfattah. xKV: Cross-layer SVD for KV-cache compression, 2025b.

Jian Chen, Zhuoran Wang, Jiayu Qin, Ming Li, Meng Wang, Changyou Chen, Yin Chen, Qizhen Weng, and Yirui Liu. Towards dynamic KV-cache compression: Fine-grained evaluation of key and value ranks in LLMs, 2025. NeurIPS 2025 LLM Evaluation Workshop.

Jian Chen, Zhuoran Wang, Jiayu Qin, Ming Li, Meng Wang, Changyou Chen, Yin Chen, Qizhen Weng, and Yirui Liu. KV-CoRE: Benchmarking data-dependent low-rank compressibility of KV-caches in LLMs, 2026.

Yilong Chen, Linhao Zhang, Junyuan Shang, Zhenyu Zhang, Tingwen Liu, Shuohuan Wang, and Yu Sun. DHA: Learning decoupled-head attention from transformer checkpoints via adaptive heads fusion. In *Advances in Neural Information Processing Systems*, 2024a.

Yuang Chen, Cheng Zhang, Xitong Gao, Robert D. Mullins, George A. Constantinides, and Yiren Zhao. Optimised grouped-query attention mechanism for transformers (AsymGQA), 2024b. ICML 2024.

DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model, 2024.

Tao Ji et al. Towards economical inference: Enabling DeepSeek's multi-head latent attention in any transformer-based LLMs, 2025.

Qingyun Jin, Xiaohui Song, Feng Zhou, and Zengchang Qin. Align attention heads before merging them: An effective way for converting MHA to GQA, 2024.

Zohaib Khan, Muhammad Khaquan, Omer Tafveez, Burhanuddin Samiwala, and Agha Ali Raza. Beyond uniform query distribution: Key-driven grouped query attention (KDGQA, DGQA, PGQA), 2024.

Rahul Krishnan and Volker Schulz. A JoLT for the KV cache: Near-lossless KV cache compression via joint rank-bit allocation, 2026.

Damien Lesens, Beheshteh T. Rakhshan, and Guillaume Rabusseau. KQ-SVD: Compressing the KV cache with provable guarantees on attention fidelity, 2025.

Songtao Liu, Hongwu Peng, Zhiwei Zhang, Zhengyu Chen, and Yue Guo. Multi-head low-rank attention. In *International Conference on Learning Representations*, 2026.

Fanxu Meng. GQLA: Group-query latent attention for hardware-adaptive LLM decoding, 2026.

Fanxu Meng, Pingzhi Tang, Zengwei Yao, and Muhan Zhang. TransMLA: Multi-head latent attention is all you need, 2025.

Sheng Qiang, Ruiwei Chen, Yinpeng Wu, Jinyu Gu, Zhichao Hua, Yubin Xia, Binyu Zang, and Haibo Chen. MosaicKV: Serving long-context LLMs with dynamic two-dimensional KV cache compression, 2026.

Utkarsh Saxena et al. Eigen attention: Attention in low-rank space for KV cache compression, 2024.

Konrad Staniszewski et al. KV cache transform coding for compact storage in LLM inference, 2025.

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*, 2025.

Zizhong Wang, Jieying Wang, Zhao Zhang, and Jiajia Li. PuzzleKV: Page-wise low-rank decomposition for KV cache compression, 2026.

Jeffrey T. H. Wong, Cheng Zhang, Xinye Cao, Pedro Gimenes, Christos-Savvas Bouganis, George A. Constantinides, Wayne Luk, and Yiren Zhao. a^3: An analytical low-rank approximation framework for attention, 2025.

Jihao Xin, Tian Lyu, David Keyes, Hatem Ltaief, and Marco Canini. RAP: KV-cache compression via RoPE-aligned pruning, 2026.

Xianglong Yan, Zhiteng Li, Tianao Zhang, Haotong Qin, Linghe Kong, Yulun Zhang, and Xiaokang Yang. ReCalKV: Low-rank KV cache compression via head reordering and offline calibration, 2025.

Hidir Yesiltepe, Jiazhen Hu, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, Hoda Eldardiry, and Pinar Yanardag. VideoMLA: Low-rank latent KV cache for minute-scale autoregressive video diffusion, 2026.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23078–23097, 2025.

Longteng Zhang et al. LoRA-FA: Memory-efficient low-rank adaptation for large language models fine-tuning, 2023.

Rongzhi Zhang, Kuang Wang, Liyuan Liu, Shuohang Wang, Hao Cheng, Chao Zhang, and Yelong Shen. LoRC: Low-rank compression for LLMs KV cache with a progressive compression strategy, 2024.

Zhongzhu Zhou, Fengxiang Bie, Ziyan Chen, Zhenyu Zhang, Yibo Yang, Junxiong Wang, Ben Athiwaratkun, Xiaoxia Wu, and Shuaiwen Leon Song. CARE: Covariance-aware and rank-enhanced decomposition for enabling multi-head latent attention, 2026.

Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez de Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In *International Conference on Machine Learning*, 2024.

Zayd Muhammad Kawakibi Zuhri, Muhammad Farid Adilazuarda, Ayu Purwarianti, and Alham Fikri Aji. MLKV: Multi-layer key-value heads for memory efficient transformer decoding, 2024.

A MULTI-RATIO PILOT UNDER A SHARED LEARNING RATE

Table 5 reports the pilot discussed in §5.4. It predates the tuned study and uses one learning rate shared across arms and a single seed. Because §5.3 shows this protocol can manufacture a spurious advantage for the smallest arm, the arm ranking here should not be read as evidence for or against any healing mode.

d_c (compression)	Arm	MME Perception	MME Cognition	TextVQA EM
256 (8×)	encoder-only	905.2	188.2	0.4658
256 (8×)	decoder-only	964.6	173.9	0.4310
256 (8×)	full	1025.2	150.0	0.4706
512 (4×)	encoder-only	1521.3	603.9	0.7992
512 (4×)	decoder-only	1426.8	588.9	0.7470
512 (4×)	full	1410.6	524.6	0.7318
1024 (2×)	encoder-only	1530.4	621.1	0.7994
1024 (2×)	decoder-only	1407.8	413.6	0.7304
1024 (2×)	full	1458.6	468.6	0.6932

Table 5: Qwen2.5-VL-3B-Instruct, three healing arms at three compression ratios, one learning rate shared across arms, single seed. Uncompressed baseline: 1534.4 / 616.4 / 0.8028. See Table 1 for $d_c = 512$ under per-arm tuned rates.