ICLR 2026
Poster

CARE

Covariance-Aware and Rank-Enhanced Decomposition for Enabling Multi-Head Latent Attention

Zhongzhu Zhou[13] Fengxiang Bie[1] Ziyan Chen[1] Zhenyu Zhang[4] Yibo Yang[2]
Junxiong Wang[3] Ben Athiwaratkun[3] Xiaoxia Wu[3†] Shuaiwen Leon Song[13*]

[1]University of Sydney [2]KAUST [3]Together AI [4]UT Austin

github.com/FutureMLS-Lab/CARE

zzz0906.github.io/CARE/

Problem & Motivation

Why do we need CARE?

The KV-Cache Bottleneck

In standard Multi-Head Attention (MHA), each attention head materializes and caches its own key and value vectors. The KV-cache grows as:

Memory $\propto$ n_layers × n_heads × seq_len × d_head

For a 70B model serving 128K context, the KV-cache alone can exceed 40GB — dominating GPU memory and limiting throughput.

Multi-Head Latent Attention (MLA)

Introduced in DeepSeek-V2, MLA compresses K,V into low-dimensional **latent vectors**, caches only these latents, and restores full per-head expressivity via lightweight up/down projections:

$K = (X W^a) W^b$ cache only $X W^a \in \mathbb{R}^{Txr}$

This trades modest extra FLOPs for dramatic memory reduction (up to 93.75% KV savings).

The Conversion Challenge

The open-source ecosystem is dominated by pretrained MHA/GQA checkpoints:

- Retraining under MLA from scratch is prohibitively expensive
- Can we convert existing models to MLA ***post-hoc***?

Prior methods (TransMLA, MHA2MLA, X-EcoMLA) use naïve SVD, but this has two fundamental problems.

✗ Problem 1: Wrong Objective

Naïve SVD minimizes weight reconstruction error $\|W-\hat{W}\|$, but what matters for model accuracy is **activation error** $\|XW-X\hat{W}\|$. These are fundamentally different objectives.

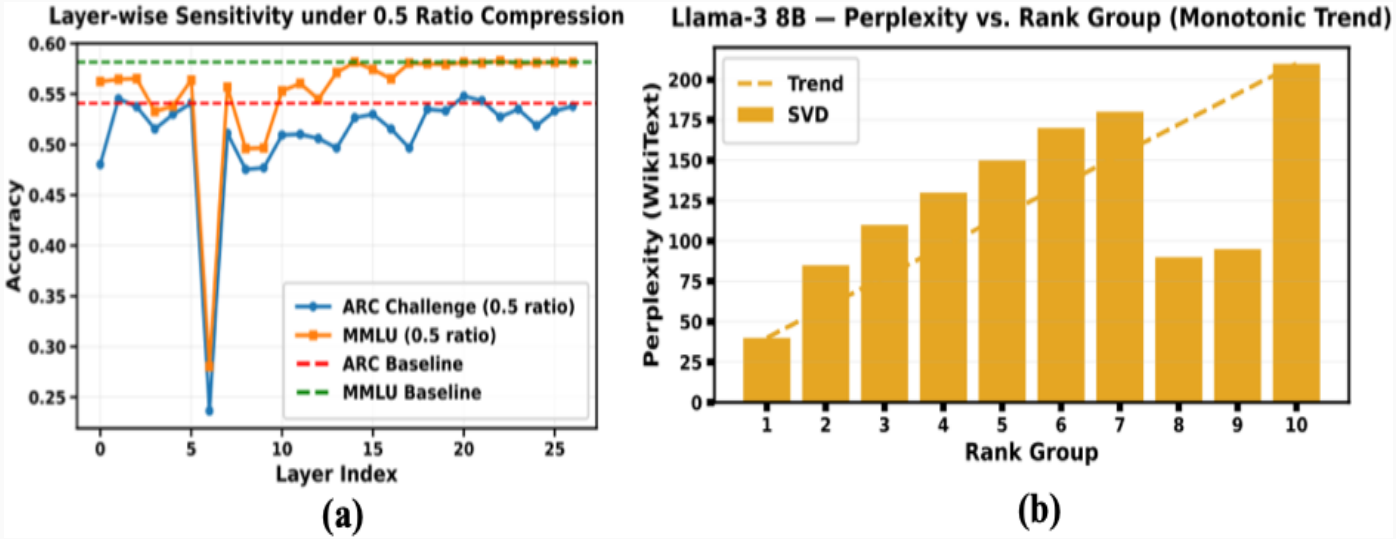
✗ Problem 2: Uniform Rank

All layers get the same rank budget, ignoring layer-wise sensitivity. Some layers tolerate aggressive compression; others collapse with even small rank reduction.

CARE addresses both problems: covariance-aware factorization for activation fidelity + per-layer rank allocation via water-filling.

Key Observations

Why naïve SVD fails for MLA conversion



(a) Layer-wise accuracy under 50% rank reduction (DeepSeek-V2-Lite)
(b) Non-monotonic perplexity when zeroing individual SVs (Llama-3-8B)

Observation 1

Accuracy-preserving rank is **not uniform** across layers. Some layers tolerate aggressive compression; others incur sharp accuracy drops.

Observation 2

Singular values are **poor proxies** for accuracy importance. The relationship is non-monotonic — SVD optimizes weight error, not activation error.

→ Need: activation-aware decomposition + per-layer rank allocation

CARE: Three-Step Pipeline

1 Step 1: Activation-Preserving Factorization

- Apply SVD on $\sqrt{C} \cdot W$ instead of W
- Directly minimizes activation error
- $\hat{W} = \sqrt{C}^{-1} U_r \Sigma_r V_r^T$

$$\min \|XW - X\hat{W}\|^2 = \|\sqrt{C}(W - \hat{W})\|^2$$

2 Step 2: Adjusted-Rank Allocation (Water-Filling)

- Score layers by spectral tail energy
- Greedy allocation to argmax layer
- K and V budgets independent

$$s(\ell) = \sigma^2_{r+1} / \sum \sigma^2_i \text{ (residual ratio)}$$

3 Step 3: KV-Parity Mapping + Healing

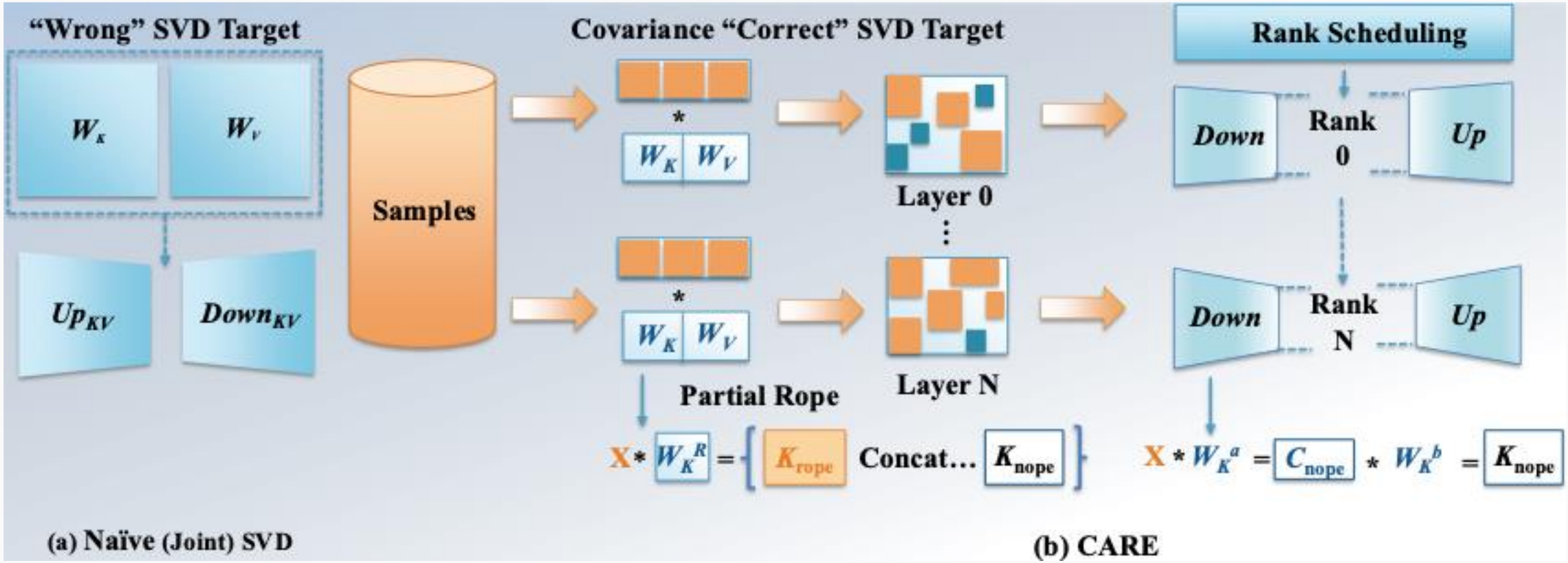
- Map to MLA with decoupled RoPE
- Cache only latents $XW^a \in \mathbb{R}^{T \times r}$
- Brief SFT healing closes remaining gap

$$W^a \leftarrow \sqrt{C}^{-1} U_r \Sigma_r, \quad W^b \leftarrow V_r^T$$

Key insight: Covariance re-weights directions by actual usage frequency → heavily-used directions are preserved even if their singular values are small.

CARE Pipeline: Naïve SVD vs. CARE

Visual comparison of the two approaches



Naïve SVD

- Directly decomposes $W = U\Sigma V^T$
- Minimizes $\|W - \hat{W}\|$ in **weight space**
- Ignores input statistics → poor activation fidelity
- Uniform rank across all layers

CARE

- Factors $\sqrt{C} \cdot W = U\Sigma V^T$ where $C = Cov[X]$
- Minimizes $\|XW - X\hat{W}\|$ in **activation space**
- Preserves directions that matter most for downstream tasks
- Per-layer rank allocation via water-filling

One-Shot Results

No fine-tuning — direct initialization comparison

215×
Perplexity Reduction

+21 pts
Accuracy Gain

50–93.75%
KV Cache Savings

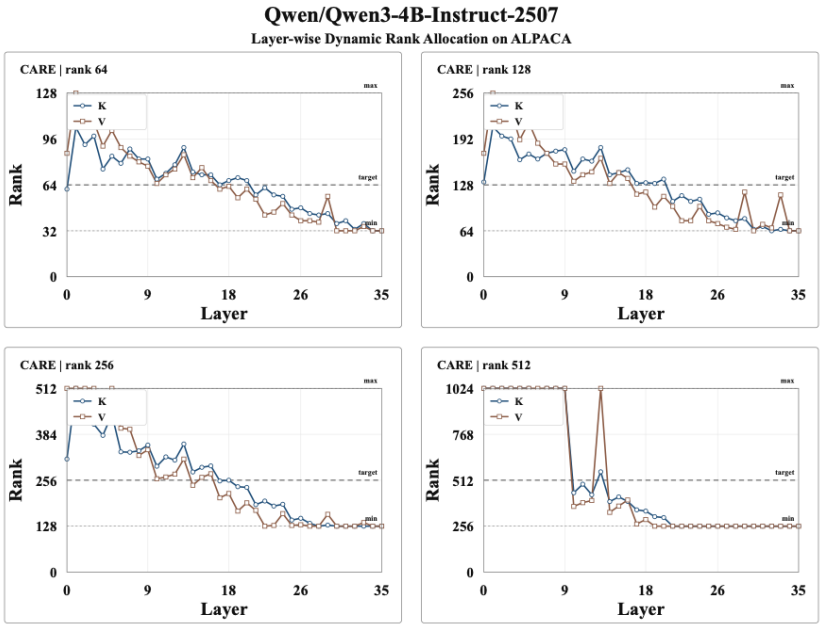
Llama-3.1-8B-Instruct | Rank 512, 50% KV Save

Method	PPL↓	ARC	ARE	Hella	PIQA	MMLU	OBQA	WG	AVG
GQA (Orig)	**7.21**	**50.3**	**80.2**	**60.2**	**79.7**	**48.1**	**34.8**	**72.7**	**58.2**
Palu (SVD)	45.4	28.2	46.0	43.4	64.2	25.0	30.8	53.4	39.3
MHA2MLA	220	25.9	41.0	39.3	61.4	25.5	26.6	56.6	37.9
ASVD	12.0	46.3	69.1	70.8	76.2	41.8	36.6	66.9	55.2
SVD-LLM V2	9.63	52.4	76.7	73.6	78.5	62.3	40.4	72.2	62.2
CARE-U (Ours)	**9.64**	**52.7**	**76.3**	**74.0**	**78.7**	**62.2**	**40.6**	**72.6**	**62.3**

Qwen3-4B-Instruct-2507 | Rank 512, 50% KV Save

Method	PPL↓	ARC	ARE	Hella	PIQA	MMLU	OBQA	WG	AVG
GQA (Orig)	**10.0**	**55.9**	**83.1**	**52.7**	**76.0**	**73.4**	**32.0**	**68.1**	**60.3**
Palu (SVD)	34.0	35.6	47.6	50.4	65.2	27.9	32.8	52.6	42.8
MHA2MLA	101	27.1	41.1	38.0	59.2	29.1	27.2	54.1	38.2
ASVD	15.5	47.6	66.5	67.5	73.0	56.6	35.6	62.8	55.6
SVD-LLM V2	11.9	54.6	77.4	68.5	75.7	67.7	39.8	67.3	61.1
CARE-U (Ours)	**12.0**	**55.0**	**77.2**	**69.2**	**76.2**	**67.5**	**40.0**	**68.4**	**61.6**

CARE-U wins AVG ACC on both models as a single-step initialization — no iterative optimization needed.



Rank allocation profiles across layers

Recovery: 100% MLA with Healing

Brief SFT closes the remaining gap

Llama-3.1-8B-Instruct | Rank 512, 50% KV Save | Healed on Alpaca

Method	PPL↓	ARC	ARE	HellaSwag	PIQA	MMLU	OBQA	WG	AVG
GQA (Orig)	**7.21**	**50.3**	**80.2**	**60.2**	**79.7**	**48.1**	**34.8**	**72.7**	**58.2**
TransMLA+Heal	7.93	48.5	78.5	57.9	78.6	48.7	33.4	67.5	56.1
MHA2MLA+Heal	8.49	47.5	77.1	57.5	78.3	46.2	35.4	69.3	56.4
CARE-E+Heal (Ours)	**7.55**	**49.2**	**79.0**	**59.3**	**79.2**	**51.5**	**35.2**	**72.4**	**57.6**

7.55

Best PPL among
all conversion methods

57.6 / 58.2

98.7% of original
accuracy recovered

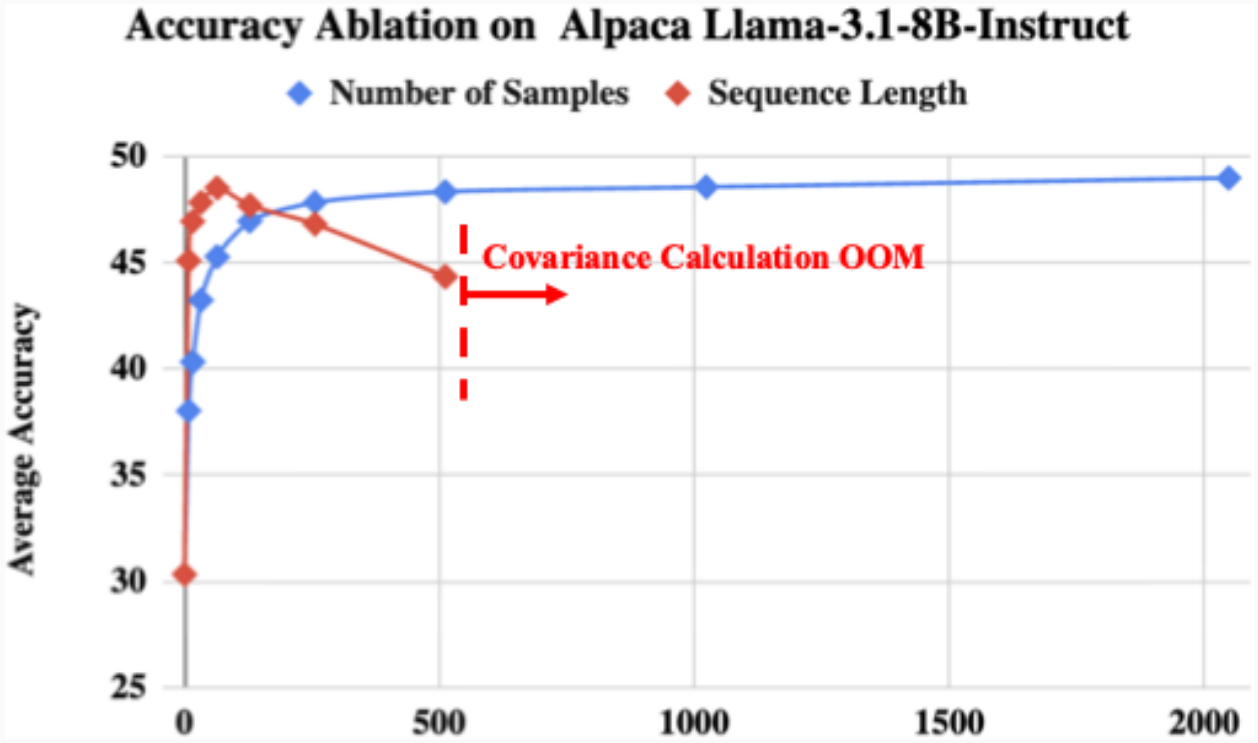
+1.5 pts

Over TransMLA
and MHA2MLA baselines

CARE provides a **stronger starting point** for healing — less fine-tuning needed to recover original model quality. The healed model is a **full MLA architecture** compatible with DeepSeek-style inference.

Ablation Study

Calibration robustness and rank consistency



Calibration ablation: √C vs C formulation, sample count sensitivity

Key Findings

✓ √C formulation

Using √C is more robust than using C directly. The square-root reweighting avoids over-emphasizing dominant directions.

✓ 256 samples suffice

Performance saturates at 256–512 calibration samples with seqlen 32. Minimal data requirement for practical deployment.

✓ Model-intrinsic rank profiles

Rank allocation profiles are consistent across Alpaca, WikiText2, PTB, C4 from 1.5B to 70B

→ CARE is **fast, data-efficient, and robust** — a practical tool for production deployment.

Conclusions & Takeaways

Covariance-Weighted Factorization

Replace naïve SVD with √C·W decomposition that directly minimizes activation error $\|XW - X\hat{W}\|$

Water-Filling Rank Allocation

Per-layer ranks via energy-driven schedule, adapting to each layer's spectral difficulty automatically

Practical MLA Deployment

Convert existing MHA/GQA models to MLA post-hoc with 50–94% KV savings, only 256 calibration samples

Results at a glance:

- One-shot: up to **215× perplexity reduction** and **+21 pts accuracy gain** vs baselines at matched KV budgets
- After healing: **98.7% of original accuracy** with 50% KV cache reduction (PPL 7.55 vs original 7.21)
- Only 256 calibration samples needed — fast, data-efficient, model-agnostic

Thank you!

github.com/FutureMLS-Lab/CARE
zzz0906.github.io/CARE/

zhongzhu@together.ai