DepthWeave-KV

Token-Adaptive Cross-Layer Residual Factorization for Long-Context KV Cache Compression

Independent paper-reading presentation

arXiv:2607.06523v1 — 20 September 2026

Paper authors: Anna Córdoba, Adam Puente Tercero, Nerea Angulo Hijo, Mar Linares Tercero, Julia Barrientos, Ainhoa Miranda, Jesús Olivera. Submitted 7 July 2026.

Share common structure; preserve important deviations

Core proposal

Shared low-rank KV bases across neighboring layers, plus token-specific residual capacity controlled by attention-output error.

- Reported: 8.3× KV reduction; average score 62.9 vs. 63.8 for full KV.
- Reported decode throughput: 72.8 vs. 42.1 token/s.
- Evidence boundary: model and hardware are unspecified; results are not independently reproduced.

Paper, Secs. 3–4, pp. 3–5; Table 1. All results are author-reported.

Uniform compression can lose late-useful evidence

- Long contexts make repeated KV reads and cache residency expensive.
- An entity or instruction boundary may matter much later in generation.
- Neighboring layers can share structure while retaining different details.
- Design goal: spend fidelity across both depth and tokens.

Paper, Sec. 1, pp. 1–2. Motivation, not a measured layer-redundancy result.

The proposal combines three compression decisions

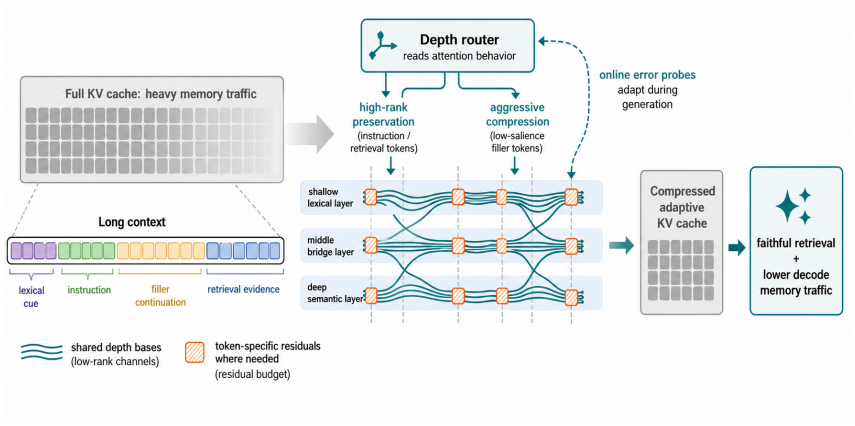
Decision	Representative comparison	Proposed choice
Which tokens?	Eviction / retention	Keep compressed tokens
Which representation?	Layer sharing / low rank	Shared bases + residuals
How much fidelity?	Fixed allocation	Token routing + feedback

Closest conceptual comparison

MiniCache and KVSharer exploit depth redundancy. Here, layers share a subspace while retaining token-local coefficients and corrections.

Paper, Sec. 2, pp. 2–3. Comparison roles as described by this paper.

Overview: residual capacity follows token importance



Paper, Fig. 1, p. 2; original source image. Conceptual diagram, not experimental evidence.

Factorization separates shared and local information

$$\widehat{X}^{Z}_{\ell,h,t} = A^{Z}_{\ell,h,t} B^{Z}_{m,h} + g^{Z}_{\ell,h,t} R^{Z}_{\ell,h,t}, \quad Z \in \{K, V\}.$$

- $B \in \mathbb{R}^{r_b \times d}$: shared across tokens and layers in window $\mathcal{W}_m$.
- $A \in \mathbb{R}^{1 \times r_b}$: coefficients specific to layer, head and token.
- $R \in \mathbb{R}^{1 \times d}$: residual reconstruction; $g \in [0, 1]$: gate.
- ℓ, h, t: layer, head, token; d: head dimension; r_b: basis rank.

Paper, Eq. (1), p. 3. K and V are handled separately.

Quantization reduces storage, but updates need detail

- Initialize per-window, per-head bases with a randomized sketch after prefill.
- Update bases with an exponential moving average during generation.
- Store coefficients in 8-bit blocks; residuals in 4-bit groups.
- Not specified: basis rank, window size, overlap resolution, quantization groups.
- Open implementation question: how do old coefficients stay valid after basis changes?

Paper, Sec. 3, p. 3. Last two points: missing details identified in this reading.

Routing uses salience and recent output error

$$\rho_{m,h,t} = \rho_{\min} + (\rho_{\max} - \rho_{\min})\mathbf{1}[\sigma(u^\top \phi_{m,h,t}) > \tau_m].$$

- ϕ: accumulated attention, maximum spike, boundary indicators, probe error.
- u: learned router weights; τ_m: adaptive memory-budget threshold.
- Lower τ_m promotes more tokens to higher residual capacity.
- Equation: two levels. Prose: “three levels, but lists 0, 2, 4 and 8.

Paper, Eq. (2) and Sec. 3, p. 3. The multi-level mapping is not fully defined.

Output probes target behavior after aggregation

$$e_{\ell,h} = \frac{\|O_{\ell,h} - \widehat{O}_{\ell,h}\|_2}{\|O_{\ell,h}\|_2 + \epsilon},$$

$$O = \mathrm{Attn}(Q, K, V), \qquad \widehat{O} = \mathrm{Attn}(Q, \widehat{K}, \widehat{V}).$$

- Every p decode steps, sample heads and recent queries.
- High error: lower the threshold; persistently low error: raise it.
- This is a relative, dimensionless output error, not a task-correctness guarantee.

Paper, Eq. (3), p. 4. O notation introduced here only to shorten the original equation.

Feedback needs a recoverable high-fidelity reference

- The paper temporarily materializes a higher-fidelity reference cache.
- It does not explain how previously discarded information is recovered.
- Promoting an old token needs residual information, not just a larger rank label.
- Probe interval, tolerances, peak memory and token-level error attribution are unspecified.

Paper, Sec. 3, pp. 3–4; implementation questions from this reading.

Frozen LLM weights do not make the method training-free

Offline

Residual gates and router parameters are trained once on unlabeled long-context text with a frozen teacher-cache objective.

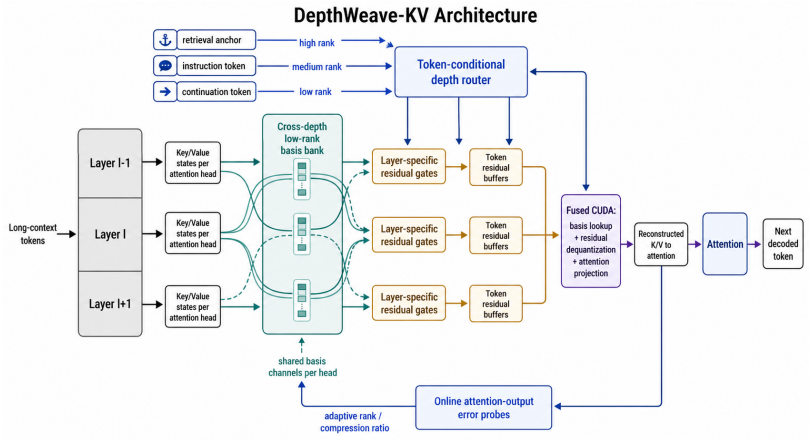
Deployment

No task-specific calibration; online basis and threshold adaptation. Base-model weights remain unchanged.

Training data, loss details, optimization and transfer across models are not reported.

Paper, Sec. 3, p. 4. This distinction qualifies the abstract's calibration-free claim.

Fusion aims to avoid writing reconstructed KV back



Paper, Fig. 2, p. 4; original image. Basis lookup, residual dequantization and attention are fused.

Evaluation reports a configuration, not a full frontier

- Default: 64K prefill, greedy decoding, 512 generated tokens.
- LongBench / L-Eval; QA, summarization and Needle retrieval.
- Compressed baselines target quality with at least 4× KV reduction; actual budgets differ.
- Model, GPU, batch size, subset weights, sample counts and run variance are missing.
- Needle protocol says 16K–128K averaging; the main table caption says 64K.

Paper, Sec. 4, pp. 4–5; Table 1. Preserve the Needle length ambiguity.

Reported quality remains below full KV

Method	Avg. score (%)	Needle (%)	KV reduction
Full KV	63.8	98.7	1.0×
MiniCache	59.8	89.4	6.4×
ChunkKV	60.7	91.2	6.0×
TailorKV	61.4	92.6	6.5×
DepthWeave-KV	62.9	96.1	8.3×

- Full-KV gaps: -0.9 average-score points; -2.6 Needle points.
- TailorKV gains: $+1.5$ and $+3.5$ points, respectively.
- $8.3\times$ leaves about 12.05% of the original KV memory, not total VRAM.

Table 1, p. 5; selected rows re-typeset from same-version TeX because PDF tables overlap.

Reported speedup depends on unspecified hardware

Method	token/s ↑	TTFT s ↓	PPL Δ ↓	Error ↓
Full KV	42.1	6.84	0.00	0.000
MiniCache	63.2	5.48	0.24	0.061
TailorKV	65.0	5.36	0.18	0.049
DepthWeave-KV	72.8	4.89	0.09	0.027

- Calculated: +72.9% throughput vs. full KV; +12.0% vs. TailorKV.
- TTFT decreases by 1.95 s vs. full KV; no timing breakdown is given.
- At the reported precision, 0.09 is exactly half of 0.18.

Table 1 and Sec. 4, p. 5. Prose says “less than half; table values do not support that wording.

Task-family evidence is limited to aggregate gaps

Retrieval / extractive group

Qasper, HotpotQA, MultiFieldQA-en, TriviaQA and Needle: gap to full KV is 1.4 points for DepthWeave-KV, 5.6 for MiniCache, 3.2 for TailorKV.

Summarization group

GovReport and QMSum: DepthWeave-KV improves over TailorKV by 0.8 points and ChunkKV by 1.7 points.

No task-level absolute scores or uncertainty estimates are supplied.

Paper, Sec. 4, “Breakdown by task family, p. 5. These are prose aggregates.

Residuals and routing preserve retrieval in Table 2

Variant	Avg. score	Needle (%)	KV reduction
Complete	62.9	96.1	8.3×
No cross-depth sharing	61.5	93.2	5.9×
No token router	61.2	92.4	7.9×
No online tracking	61.7	93.5	8.5×
No residual gates	61.0	91.8	8.4×
Shared bases only	58.9	87.6	10.2×

Thresholds may be retuned: these are not strictly equal-budget ablations.

Table 2, p. 5, re-typeset from TeX. Sec. 5, pp. 5–6. Both score columns: higher is better.

The fused kernel mainly changes systems performance

Variant	Needle (%)	KV ×	token/s	TTFT s
Complete	96.1	8.3	72.8	4.89
No fused CUDA	95.9	8.3	61.9	5.58
No online tracking	93.5	8.5	73.4	4.84
Shared bases only	87.6	10.2	76.4	4.62

- Removing fusion lowers throughput by 15.0%; adding it raises throughput by 17.6% (different denominator).
- The complete method is not the fastest or smallest ablation.
- Probe overhead is not isolated when compression also changes.

Table 2, p. 5; Sec. 5, p. 6. Percentage calculations made for this reading.

Conflicting ablation reports should remain visible

Case	Sec. 4 prose	Table 2
No router: score / Needle	61.1 / 91.8	61.2 / 92.4
No probes: Needle	89.9 (“further”)	93.5
Direct layer sharing	6.9×, error .052	No matching row

- The 89.9 value may describe a cumulative ablation, but this is not explicit.
- This presentation uses Table 2 for systematic component comparisons.
- PDF p. 5 tables overlap and extend beyond columns; TeX enables recovery of the entries.

Paper, Sec. 4 and Table 2, p. 5. Do not merge or average conflicting results.

Reproduction requires more than the conceptual equations

- Define residual decoding, depth-window overlap and historical coefficient updates.
- Explain high-fidelity probes and recovery of promoted residuals.
- Release model / hardware / training details and exact cache layouts.
- Measure matched-budget curves, per-task variance and length-by-depth retrieval.
- Check all metadata and temporary caches in peak memory accounting.

Reading analysis based on omissions in Secs. 3–5. These are proposed checks, not completed experiments.

Source artifacts also warrant independent verification

- PDF p. 1 retains an ICML 2024 proceedings footer; arXiv v1 is dated July 2026.
- The source contains a “Fixed synthetic byline comment.
- Unrelated bibliography entries appear alongside cache papers; the source uses `\nocite{*}`.
- These are observable document anomalies, not proof of misconduct.
- No runnable CUDA implementation appears in the downloaded source bundle.

Paper p. 1, references pp. 6–9, main.tex and downloaded source manifest. External provenance not established.

Useful design idea; performance claims need validation

1. Share depth-local structure while preserving token-specific deviations.
2. Combine behavior-aware feedback with compression-aware execution.
3. Treat the reported 8.3× / 72.8 token/s as conditional, unreproduced results.

Authors' future directions

Non-contiguous layer groups, risk-aware routing, batching, prefix reuse and disaggregated serving. Not evaluated in this version.

Paper, Secs. 6–7, p. 6; limitations from this reading. Designed for a 20–30 minute discussion.

Source and reading scope

Anna Córdoba et al.
DepthWeave-KV: Token-Adaptive Cross-Layer Residual Factorization for Long-Context KV Cache Compression.

`https://arxiv.org/abs/2607.06523v1`

Read: complete main.tex + body.tex, PDF pp. 1–9, all three equations, two figures and two tables. No appendix in the retrieved version.

Tables re-typeset from the same-version TeX; original PNG figures retained. No independent model runs or external-reference audit.

Retrieved 20 September 2026. Chinese companion notes contain full baseline and ablation tables.