FusedKV: Reconstructing KV Caches with Cross-Layer Fusion

Paper discussion: quality, memory, and inference trade-offs

Reading presentation of work by Hongzhan Lin et al.

arXiv:2512.03870v3 (19 February 2026)
Reading: 20 September 2026

Half the cache can improve language modeling

- Upper layers reconstruct K/V from early and middle layers.
- At 1.5B: WikiText PPL 13.67 $\rightarrow$ 13.33; mean task accuracy 54.55 $\rightarrow$ 55.82.
- Cache capacity falls by 50%; decoding latency does not always improve.
- Long-context quality remains below the full-cache baseline.

Tables 1–2, 16; PDF pp.5, 7, 23. About 25 minutes + backup.

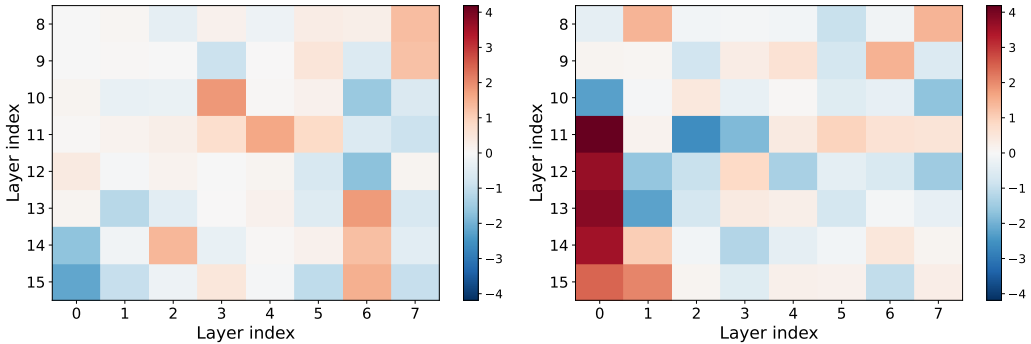
Capacity and bandwidth are different bottlenecks

$$\text{KV elements} = 2LSH_{kv}D$$

- L: layers; S: sequence length; H_{kv}: KV heads; D: head dimension.
- Multiply by batch size and bytes per element for payload memory.
- GQA shares within layers; CLA and YOCO share across layers.
- Research question: should keys and values use the same source layer?

Sec.1–2; Table 1, PDF pp.2–5.

Learned fusion suggests asymmetric source roles



Left: key weights. Right: value weights. Columns: source layers; rows: upper layers. Signed coefficients are not probabilities.

Fig.2, PDF p.2. Exploratory 16-layer, 1B model; zero-based plot indices.

Only the lower half stores independent caches

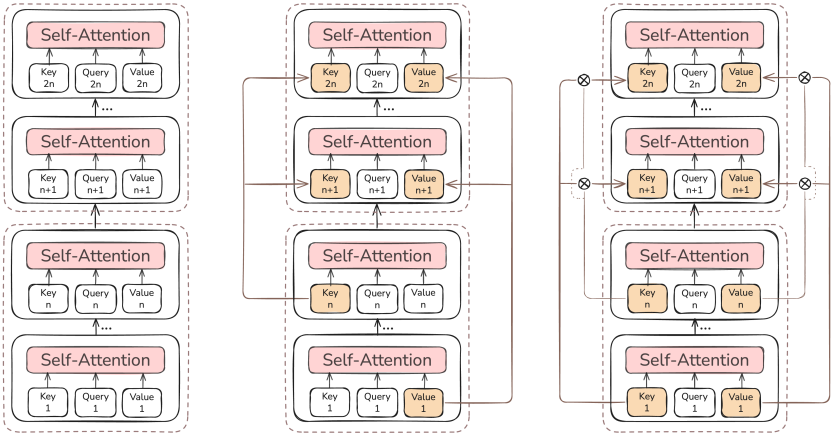


Fig.3, PDF p.4. Queries and upper-layer computation remain layer-specific.

FusedKV learns two-source, channel-wise fusion

$$K^i = a_{i,1} \odot K^1 + a_{i,n} \odot K^n,$$
$$V^i = b_{i,1} \odot V^1 + b_{i,n} \odot V^n, \qquad i > n = L/2.$$

- $K^i, V^i \in \mathbb{R}^{S \times d}$ in the paper's simplified notation.
- Learned vectors broadcast over tokens; each target layer has its own weights.
- Weights need not be positive or sum to one.
- Train the architecture end to end; no frozen-teacher KV reconstruction loss is specified.

Sec.2.1–2.3, Eqs.(1), (4)–(5), PDF pp.3–4.

Lite separates the key source from the value source

$$K^i = K^n, \qquad V^i = V^1, \qquad i > n.$$

- No two-source fusion: read one key cache and one value cache.
- For 16 layers, layers 1–8 store caches; layers 9–16 reuse K^8, V^1.
- Shared K/V does not imply identical attention: upper-layer queries still differ.
- This is an architecture trained for sharing; arbitrary pretrained checkpoint conversion is not established.

Sec.2.3, Eq.(6), Fig.3, PDF p.4.

Post-RoPE fusion needs paired key weights

$$(R_m q)^T W (R_t k) = q^T R_m^T W R_t k.$$

$$w_{2j} = w_{2j+1} = c_j \quad \Longrightarrow \quad W = c_j I \quad \Longrightarrow \quad c_j q^T R_{t-m} k.$$

- Tie both coordinates in each RoPE rotation pair.
- Unequal weights introduce absolute-position terms involving $m + t$.
- Linear sums preserve the property if every source meets the constraint.
- Assume compatible position/frequency conventions across sources.

Sec.2.4, Eqs.(7)–(8), PDF pp.4–5; A.4, pp.16–17. Matrix form is a teaching derivation.

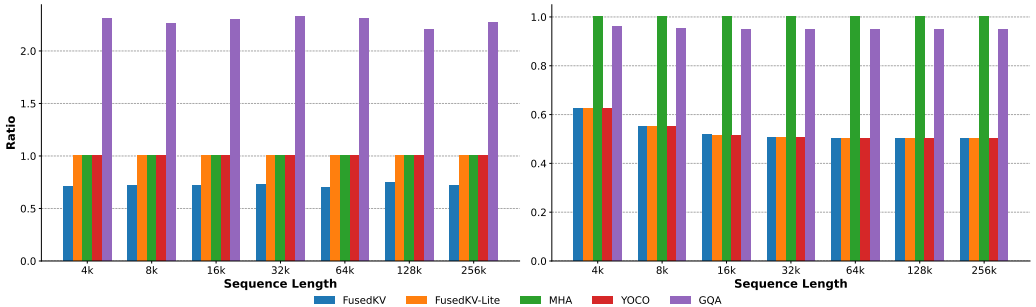
Half the storage can require 50% more cache I/O

Method	Cache capacity	Cache I/O
Vanilla	1.0	1.0
YOCO / Lite	0.5	1.0
FusedKV	0.5	1.5

- All ratios hold KV head count and precision fixed.
- Fuse on demand; retaining all reconstructed upper caches defeats the capacity saving.
- Table 1 counts attention computation only, excluding FFNs.

Table 1, PDF p.5. Ratios derived from the reported element counts.

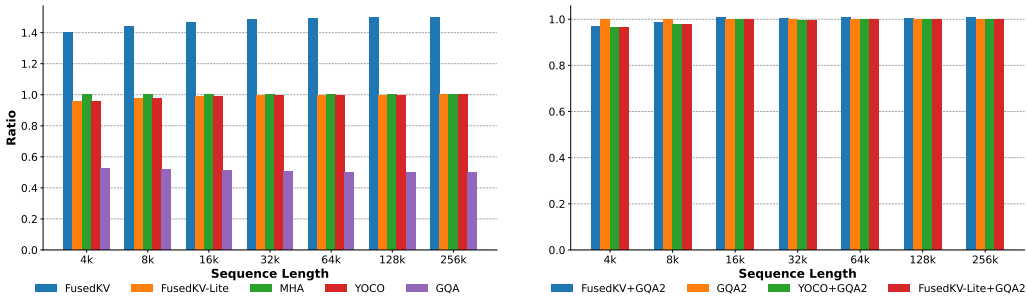
Prefill improves while kernel throughput falls



Left: throughput, higher is better. Right: TTFT, lower is better. FusedKV kernel throughput is 28.4% lower on average; TTFT is about halved at $S \geq 8$k.

Sec.2.5, Fig.4, PDF pp.5–6; normalized to MHA.

Decode cost depends on the inference regime



Left: memory-bound TPOT is about 1.5× for FusedKV. Right: $H_q = 128, H_{kv} = 2$; overhead is hidden. Lite stays close to baseline.

Sec.2.5, Fig.5, PDF p.6. Lower TPOT is better; ratios use the respective baselines.

Evaluation spans four model sizes

Parameters	Layers	Heads	Training tokens
332M	12	16	200B
650M	16	16	200B
1.5B	24	16	400B
4B	36	32	800B

- Qwen3-style decoder; FineWeb-Edu; 8192 context; vocabulary 128,000.
- 500M-token validation split; WikiText perplexity.
- Seven five-shot task accuracies, averaged; no multi-seed uncertainty reported.

Sec.3.1/3.3; Tables 4,6; PDF pp.6–8,14–15.

At 1.5B, both variants improve mean accuracy

Method	Cache	Valid loss	WikiText	Avg. acc.
Vanilla	1	2.241	13.67	54.55
CLA	1/2	2.258	14.19	53.91
YOCO	1/2	2.244	13.65	54.19
GQA	1/2	2.245	13.74	54.58
Lite	1/2	2.225	13.45	55.30
FusedKV	1/2	2.221	13.33	55.82

FusedKV: +1.27 percentage points in average accuracy; 2.49% relative reduction in WikiText PPL. Valid loss is not perplexity.

Table 2, PDF p.7. GQA uses two Q heads per KV head by default.

Smaller models show mixed downstream results

Size	Method	WikiText ↓	Avg. acc. ↑
332M	Vanilla	22.85	43.33
	Lite	22.78	43.19
	FusedKV	22.35	43.59
650M	Vanilla	18.47	48.08
	Lite	18.55	48.51
	FusedKV	18.09	47.45

Lower language-model loss does not guarantee higher task accuracy. The prose reverses the small-model winners; these are the table values.

Table 2 and Sec.3.2, PDF pp.7–8.

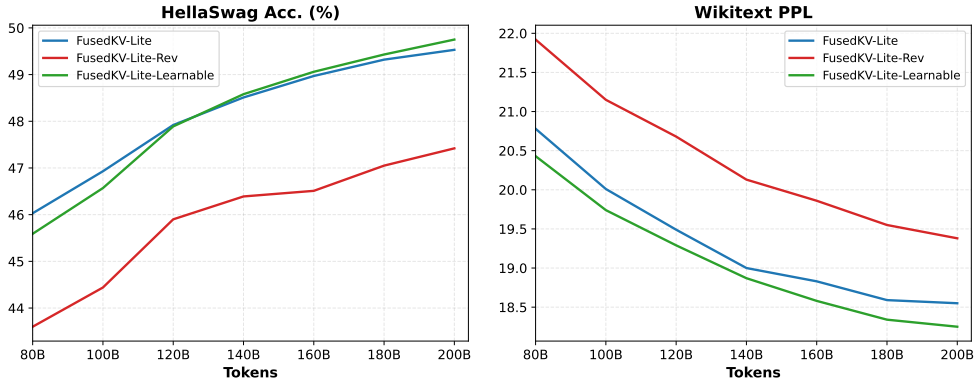
Larger scale and longer training retain gains

4B model	Valid loss	WikiText	Avg. acc.
Vanilla	2.002	9.18	59.07
FusedKV	1.978	8.94	60.01

- Scaling experiments cover 332M–4B, with different token budgets.
- 1.5B training continues from 400B to 600B tokens at LR 3×10^{-5}.
- Reported learning curves retain an advantage in the extended run.
- Evidence for persistence over the tested range; not a proof of asymptotic scaling.

Table 3, Fig.7; A.5, Figs.11–12; PDF pp.8,17–19.

The direction of asymmetry matters



Reversed sharing uses K^1, V^8; the proposed Lite uses K^8, V^1. Learnable channel weights improve WikiText over fixed Lite.

Fig.8, PDF p.9. 650M model, 200B tokens. No LAMBADA panel is shown.

Source-layer choice supports a qualified conclusion

K source	V source	Valid loss	WikiText	Avg. acc.
8	1	2.473	18.55	48.51
8	5	2.489	18.62	48.22
8	8	2.498	19.21	47.18
5	1	2.473	18.40	48.39
3	1	2.479	18.60	48.00

- Earlier values help at fixed key source.
- Middle-layer keys give the best mean accuracy among tested choices; layer 5 wins WikiText.
- Larger shallow-layer gradient norms are suggestive, not causal proof of better learning.

Table 17, PDF p.23; A.8, p.24; gradient analysis: Sec.4/Fig.9, pp.9–10.

GQA and MLA compose, with residual trade-offs

650M method	Cache vs. MHA	WikiText	Avg. acc.
GQA	1/2	19.05	47.04
GQA + FusedKV	1/4	18.48	47.15
MLA	1/2	21.03	45.03
MLA + FusedKV	1/4	20.08	44.79

- GQA TPOT: 0.512 $\rightarrow$ 0.741 relative to MHA; slower than GQA alone.
- MLA fuses compressed latent KV and positional keys.
- MLA jointly compresses K/V; Lite is not directly transferable.

A.6.1–2, Tables 8,11, PDF pp.18–21. Separate appendix experiments.

MoE and hybrid SWA do not improve every metric

- MoE: 8 experts, top-2, 100B tokens; WikiText 19.62 (Vanilla) $\rightarrow$ 19.50 (FusedKV).
- Vanilla MoE training loss 2.513 is lower than Lite 2.524, despite the prose claim.
- Hybrid SWA: window 512; three window layers per full-attention layer.
- FusedKV +WF: cache 29.7% $\rightarrow$ 12.9% of full-attention reference; mean accuracy 47.84 $\rightarrow$ 48.88.
- But WikiText worsens 24.04 $\rightarrow$ 29.63; source selection must respect attention type.

A.6.3–4, Tables 12–13, PDF pp.21–22.

Four-bit KV quantization is the stronger combination

1.5B FusedKV setting	WikiText	Avg. acc.
FP16	13.33	55.82
4-bit KV (KIVI)	13.39	55.63
2-bit KV (KIVI)	15.49	50.73
50% weight pruning (Wanda)	20.63	48.56

- 4-bit adds a 4-fold payload reduction relative to FusedKV FP16.
- Combined ideal payload: 1/8 of full-cache FP16 Vanilla, excluding quantization metadata.
- Wanda storage reduction concerns model weights, not KV capacity.

A.6.5, Tables 14–15, PDF pp.22–23.

128k quality is better than YOCO, below Vanilla

Model	RULER average	Difference vs. Vanilla
Vanilla	49.11	—
YOCO	17.32	−31.79
Lite	42.31	−6.80
FusedKV	42.00	−7.11

- Additional training: 50B tokens at 32k, then 25B tokens at 128k.
- RoPE bases: 50,000 then 500,000; not zero-shot extrapolation.
- Subtask losses can be much larger: multikey_2 is 63.0 $\rightarrow$ 4.0 for FusedKV.

A.7, Table 16, PDF p.23. Higher scores are better.

The contribution is a useful design trade-off

- K and V need not share the same source layer.
- Two-source fusion can halve cache capacity and improve language modeling.
- Lite avoids fusion I/O; full FusedKV is sensitive to arithmetic intensity.
- Long-context retrieval and some downstream metrics still regress.
- Reader assessment: choose using task quality, memory, TTFT, and TPOT together.

Evidence synthesis: Secs.2–4 and A.6–A.8. No independent reproduction performed.

References and evidence scope

Primary source

Hongzhan Lin et al.

Reconstructing KV Caches with Cross-Layer Fusion for Enhanced Transformers.

arXiv:2512.03870v3, 19 February 2026.

`https://arxiv.org/abs/2512.03870v3`

- Read author TeX, referenced tables, and the 24-page PDF, including Appendix A.1–A.8.
- Author code link: `https://github.com/LivingFutureLab/FusedKV`
- Tables in this deck are selected transcriptions; figures are author originals.
- The following slides are backup material.

Paper metadata and source bundle retrieved 20 September 2026.

Backup: initialization changes the result

650M / 200B tokens	Valid loss	WikiText	Avg. acc.
Equivalent initialization	2.475	18.06	48.69
Normal initialization	2.478	18.33	48.18

- Iterative fusion carries an auxiliary cache through upper layers.
- Equivalent initialization expands the recurrence into two-source coefficients, then trains them independently.
- Table 5's iterative row conflicts with related main-table values; do not silently repair it.
- Independent cache reconstruction does not remove the hidden-state dependency between layers.

A.3, Table 5, PDF pp.14–16.

Backup: MLA timing is simulated, not deployed

Heads	Metric (ms)	MLA	MLA + FusedKV
16	TTFT	8010.93	4014.26
16	TPOT	23.58	29.39
48	TTFT	22147.56	11084.81
48	TPOT	59.64	59.64

- Input/output length: 128k/2k.
- These are InferSim estimates; the appendix does not report an integrated vLLM/SGLang benchmark.
- The 48-head compute-bound scenario uses the H20 compute-to-memory ratio.
- Do not transfer these numbers to arbitrary hardware or serving workloads.

A.6.1, Tables 9–10, PDF pp.19–20.

Backup: prefill skipping has a specific scope

- Teaching interpretation: only the final prompt position is needed for first-token output.
- Lower layers build the stored history; upper layers consume those caches.
- Earlier upper-layer hidden states no longer need to generate stored upper-layer KV.
- The final position still traverses upper-layer queries, attention, and FFNs.
- This does not eliminate all upper-layer work during full-token loss training or arbitrary sequence-output requests.

Interpretation of Sec.2.5 and Table 1, PDF pp.5–6; not an independently benchmarked implementation.