GPA: General Principled Framework for Linearizing Softmax Attention via KV Cache Approximation

Ning Wang[123*], Zekun Li[123*], Tongxin Bai[3]
Man Yao[12], Chunlei Men[3]†, Guoqi Li[12]†

NeurIPS 2026 / Sydney / Poster S3607
10 December, 17:00–20:00 AEDT / Hall 1–4

NEURAL INFORMATION PROCESSING SYSTEMS

GPA converts pretrained softmax attention into fixed-capacity recurrent attention by approximating its KV cache.

With a fixed number of slots per layer, state size is independent of context length and recurrent computation scales linearly with it.

Five principles

T1 Exact KV deduplication

Exact matches can share a representative and count.

GPA accumulates soft counts for each slot (19–20).

T2 Content–position factorization

Separate token content from its rotary position.

GPA stores the position correction separately (21).

T3 Rotary-position compression

Approximate the positional contribution to attention.

GPA adds the projected positional state to counts (23).

T4 Inter-layer capacity bound

Adjacent-layer matches bound effective types (25).

The bound guides the choice of slot budgets.

T5 Hierarchical residual encoding

Stack fixed-capacity states to model residual error.

GPA passes each reconstruction residual onward (22).

Compositional bounds — Theorem 1

Effective KV types before imposing a fixed budget

$$M_\ell \le \min\left(T, \frac{|\mathcal{V}|}{\prod_{r=1}^{\ell} p_r}\right) \tag{25}$$

End-to-end deviation under local Lipschitz bounds

$$E_{L,t} \le \sum_{\ell=1}^{L} e_{\ell,t} \prod_{r=\ell+1}^{L} \Lambda_r \tag{27}$$

T is context length. Vocabulary size and positive inter-layer match ratios p_r determine the effective type bound. Downstream Lipschitz constants weight each local error bound.

GPA recurrent update

Pretrained Q, K, and V projections

Default capacity budget (T4)

Level 0 — Coarse KV estimate — 64 slots → **Level 1** — Residual refinement — 64 slots

Cross-level routing combines the two readouts (24).

Accumulate and normalize slot counts — T1

$$n_t^{(i)} = n_{t-1}^{(i)} + f_t^{(i)}, \overline{f}_t^{(i)} = \frac{f_t^{(i)}}{\max\left(n_t^{(i)}, 1\right)} \tag{19}$$

Update the recurrent KV state — T1, T4

$$S_t^{kv(i)} = \operatorname{diag}\left(1 - \overline{f}_t^{(i)\top}\right) S_{t-1}^{kv(i)} + \overline{f}_t^{(i)\top} m_t^{(i)} \tag{20}$$

Accumulate the positional correction — T2, T3

$$S_t^{p(i)} = S_{t-1}^{p(i)} + f_t^{(i)\top}\left(k_{pt} - k_t^{(0)}\right) \tag{21}$$

Pass the residual to the next level — T5

$$m_t^{(i+1)} = m_t^{(i)} - f_t^{(i)} S_t^{kv(i)} \tag{22}$$

Correct the readout scores — T3

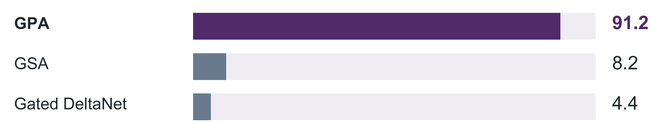
$$c_t^{(i)} = n_t^{(i)} + q_t S_t^{p(i)\top} \tag{23}$$

The superscript i indexes memory levels within a layer. f denotes the slot assignment at each step. The overbar denotes count-normalized weights. m concatenates K and V.

Evaluation

Single-needle retrieval at 32K tokens

S-NIAH-1 accuracy (%) / Qwen2.5-7B conversions



20B fine-tuning tokens per conversion (Table 15).

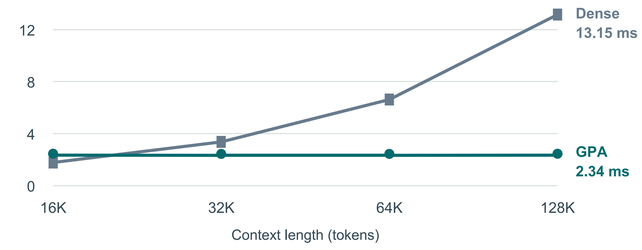
LongBench mean scores

Qwen2.5	Dense	GSA	GPA
7B	40.65	36.62	**38.80**
14B	45.65	42.34	**44.02**

Mean across five task categories (Table 3).

Attention-kernel decoding latency

GPU latency per decoding step (ms)



5 MiB GPA state vs. 258–2,050 MiB dense KV cache

A800 80 GB, batch 2. Dense cache sizes are consistent with one layer. GPA uses more prefill workspace (Table 21).

Cross-layer hybrid attention

Variant	S-NIAH-2	S-NIAH-3	MMLU
Dense	99.2	100.0	74.2
Pure GPA	10.4	5.4	56.7
Hybrid GPA	**99.8**	**89.6**	**63.6**

Qwen2.5-7B accuracy (%). S-NIAH at 32K tokens. Hybrid GPA retains 12.5% softmax layers (Table 16).

[1] Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] Beijing Academy of Artificial Intelligence

* Equal contribution. † Corresponding authors.
clmen@baai.ac.cn / guoqi.li@ia.ac.cn

Conference page
Poster 152654

Code
github.com/fisalt/GPA