Neural Attention Search Linear

Towards Adaptive Token-Level Hybrid Attention Models

Paper authors: Difan Deng, Andreas Bentzen Winje,
Lukas Fehring, and Marius Lindauer

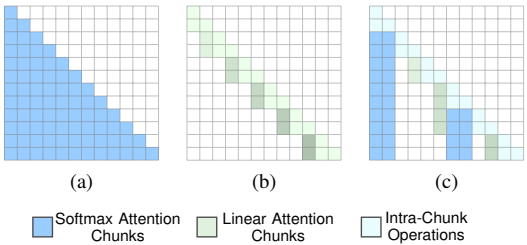
Leibniz University Hannover / L3S

Independent paper discussion

arXiv:2602.03681v2 — revised June 2, 2026
Reading date: October 15, 2026

Learn which historical blocks need explicit KV memory.

Two memory types offer different trade-offs



Softmax
Explicit historical KV.
Expensive global reads.

GDN
Fixed matrix state.
Compression can lose detail.

NAtS-L
Retain selected KV blocks.
Compress the others.

Paper, Fig. 1, PDF p. 2; Secs. 1–3, pp. 1–4. Blue: softmax; green: linear state; cyan: intra-chunk operations.

Route memory writes; mix reads for every query

The routing unit is a 64-token block within each head group

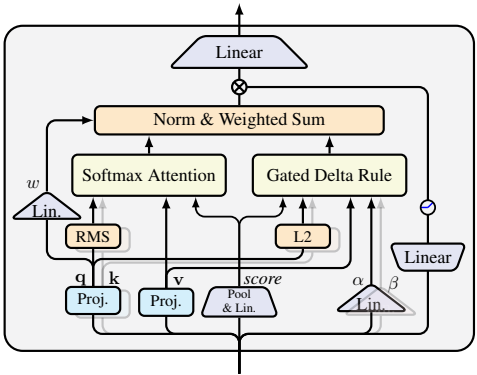
A block is assigned to explicit softmax KV or to the GDN state.

Decision	Input signal	Effect
Storage route	Mean-pooled block features	Select the historical KV columns
Readout mixture	Current query vector	Weight both normalized outputs

All queries can read both memories. The route does not select a single query-side expert.

Paper, Secs. 4.1 and 4.3, Eq. (8)–(14), (21)–(22), PDF pp. 4–6.

Shared projections feed two different operators



- Shared Q, K, V projections: linear, short convolution, SiLU.
- Softmax Q, K: RMS normalization.
- GDN Q, K: L2 normalization.
- Normalize each output, mix, gate, and project.

Paper, Fig. 2 and Sec. 4.3, PDF pp. 5–6. Figure cropped from the paper.

A small router selects historical KV blocks

$$\text{score}_t = W^{\text{score}}\,\text{Mean}(X_{[t]}), \qquad \text{opt}_t = \arg\max_a \text{softmax}(\text{score}_t)_a$$

- $X_{[t]} \in \mathbb{R}^{C \times d}$; two scores per block and head group.
- Softmax loads selected historical KV; GDN writes selected blocks.
- 800M setup: 6 GDN heads, 18 softmax heads; each group of 3 softmax heads shares a routing mask.

Training objective

Language modeling loss only: no fixed softmax quota or balancing loss.

Paper, Eq. (8)–(14), Sec. 4.3, Appendix A.1, PDF pp. 4–6, 15.

Commit the route only after the block is complete

1. At a block boundary, save the incoming GDN state.
2. Within the current block, compute **both causal branches**: temporarily update the state and cache the new KV.
3. Once the entire block is visible, select its route for future queries.

Selected: GDN

Keep the state update.
Discard this block's temporary KV.

Selected: softmax

Keep the KV. Undo GDN content writes, while retaining decay.

Paper, Sec. 4.1, PDF p. 4; GDN-specific decay in Eq. (20), p. 5. Process summarized for teaching.

GDN corrects memory and keeps its clock running

$$S_i = \alpha_i S_{i-1}(I - \beta_i k_i k_i^\top) + \beta_i v_i k_i^\top$$

$$\underbrace{S_i = \alpha_i S_{i-1} + \beta_i (v_i - \alpha_i S_{i-1} k_i) k_i^\top}_{\text{Equivalent expansion: decay, then a prediction-error write}}$$

- $S_i \in \mathbb{R}^{d_v \times d_k}$; α_i controls forgetting, β_i the write strength.
- For a softmax-selected block, no GDN content write is committed:

$$S_{[t+1]} = \left(\prod_{i \in [t]} \alpha_i \right) S_{[t]}.$$

Paper, Eq. (19)–(20), PDF p. 5. Middle equation is an algebraic teaching expansion.

Normalize branch scales before query-dependent mixing

$$O_i = w_i^{\mathrm{nla}}\,\mathrm{RMSNorm}(O_i^{\mathrm{nla}}) + w_i^{\mathrm{la}}\,\mathrm{RMSNorm}(O_i^{\mathrm{la}})$$

$$\begin{bmatrix} w_i^{\mathrm{nla}} \\ w_i^{\mathrm{la}} \end{bmatrix} = \mathrm{softmax}\left(\begin{bmatrix} W^{\mathrm{nla}} \\ W^{\mathrm{la}} \end{bmatrix} q_i \right)$$

- Softmax and GDN outputs have different scale behavior.
- Separate RMSNorm makes their combination easier to control.
- The current query sets nonnegative, sum-to-one weights per head.

Two distinct control signals

Block features decide *what to retain*; queries decide *how to read*.

Paper, Eq. (21)–(22), PDF p. 6.

Sparse training skips inactive routing gradients

For a continuously relaxed softmax mask, a useful interpretation is

$$\frac{\partial \mathcal{L}}{\partial M_{ij}} = P_{ij}\langle g_i, V_j - O_i\rangle, \quad P_{ij} = \frac{e^{A_{ij}}}{\sum_k e^{A_{ik}} M_{ik}},$$

where $g_i = \partial \mathcal{L}/\partial O_i$. Aggregate columns within each block.

- Evaluating inactive softmax choices would revisit all KV pairs.
- The method sets inactive score gradients to zero to preserve sparse cost.
- The paper does not fully specify the surrogate chain through hard arg max.

Paper, Sec. 4.2, Eq. (15)–(17), PDF pp. 4–5. Equation above is a continuous-mask teaching derivation.

The softmax term can still grow quadratically

The paper suppresses fixed width and chunk factors:

$$\mathcal{O}(LL_{\mathrm{nla}} + L_{\mathrm{la}}), \qquad \rho = \frac{L_{\mathrm{nla}}}{L} \implies \mathcal{O}(\rho L^2 + L).$$

- A constant retained fraction reduces the coefficient, not the order.
- No budget constraint guarantees a fixed number of retained KV.
- Every query still reads the linear state and computes local outputs.

Illustrative single-head memory, with width d_h:

$$\underbrace{\mathcal{O}(d_h^2)}_{\text{matrix state}} + \underbrace{\mathcal{O}((L_{\mathrm{nla}} + C)d_h)}_{\text{retained KV + current block}} .$$

Paper, Sec. 4.1, PDF p. 4. Fraction substitution and memory decomposition are this presentation's analysis.

The strongest variant mixes both across and within layers

Model	Parameters	Layer composition
GDN	793M	21 GDN
Mamba2	801M	48 Mamba2
Transformer	778M	24 softmax
GDN Hybrid	802M	17 GDN + 5 softmax
NAtS-L	794M	21 NAtS-L
NAtS-L Hybrid	793M	15 GDN + 6 NAtS-L

A material configuration difference

NAtS-L uses RoPE in its softmax branch. NAtS-L Hybrid uses no positional encoding in that branch.

Paper, Sec. 5, PDF p. 6. Exact layer counts shown instead of the approximate "3:1" description.

Training length is fixed; evaluation probes distinct skills

Pretraining

FineWeb-Edu; 4,096-token context; 380M / 15B tokens and 800M / 50B tokens.
AdamW; 0.5M tokens per effective batch.

Evaluation	What it measures
Commonsense / short PPL	General modeling quality
Six retrieval tasks	Read information from contexts truncated to 4,096
Long-text PPL	Modeling beyond the training length, up to 65,536
RULER retrieval subset	Retrieval at 4K, 8K, and 16K
LongBench-e subsets	QA, summarization, classification, and code
Latency	Prefill and decode with inputs up to 128K

Paper, Sec. 5.1 and Appendix A.1, PDF pp. 6–8, 15. Latency length is not a retrieval-quality result.

Retrieval improves more than short-context reasoning

800M model	Wiki PPL ↓	Commonsense ↑	Retrieval ↑
GDN	19.54	51.02	31.22
Mamba2	19.28	51.38	29.94
Transformer	19.27	50.78	33.66
GDN Hybrid	18.83	**51.52**	37.30
NAtS-L	19.17	50.84	39.75
NAtS-L Hybrid	**18.52**	51.41	**45.07**

- Retrieval gain over GDN Hybrid: +7.77 percentage points.
- Hybrid wins 5/6 retrieval tasks; FDA reaches 67.70 vs. 31.13.
- DROP favors NAtS-L: 22.90 vs. Hybrid's 20.84.

Paper, Table 1, PDF p. 8. Means: 7 commonsense tasks and 6 retrieval tasks; retrieval context capped at 4,096.

RULER extrapolation improves but still degrades

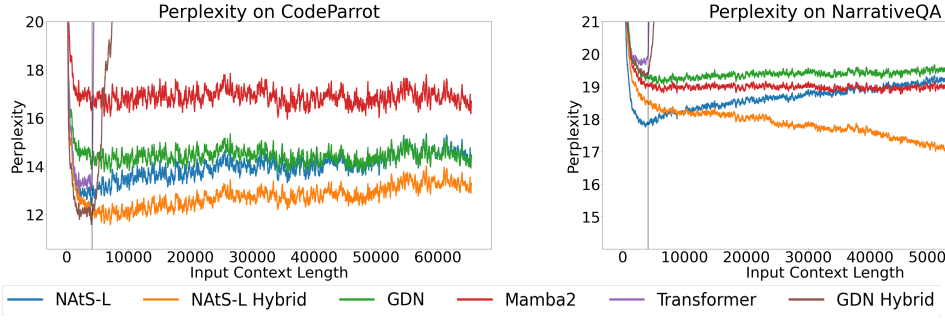
800M model	4K	8K	16K
GDN	0.25	0.14	0.04
Mamba2	0.18	0.09	0.04
Transformer	0.45	0.00	0.00
GDN Hybrid	0.47	0.02	0.00
NAtS-L	0.39	0.13	0.05
NAtS-L Hybrid	**0.49**	**0.32**	**0.21**

Hybrid is best at each length, yet drops from 0.49 to 0.21.

Single-needle retrieval is a clearer strength than harder multi-item tasks.

Paper, Table 2, PDF p. 8; per-task Fig. 9, p. 17. RULER retrieval subset; mean score on a 0–1 scale, higher is better.

Longer narrative context keeps helping the Hybrid



- NarrativeQA PPL approaches 15 near 64K; training length was 4K.
- CodeParrot uses concatenated shorter examples.
- RoPE and layer layout differ across variants, alongside routing.

Paper, two panels of Fig. 3, PDF p. 8; interpretation p. 7. Axes: context length and per-token PPL, lower is better.

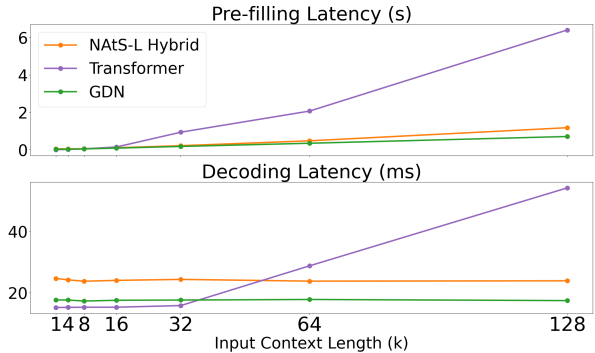
LongBench gains depend on the task

Below 4K, 800M	GDN Hybrid	NAtS-L Hybrid	Best model
GovReport	12.54	**17.26**	NAtS-L Hybrid
TriviaQA	20.10	**29.79**	NAtS-L Hybrid
SAMSum	16.21	**24.98**	NAtS-L Hybrid
Qasper	4.28	2.40	Mamba2: 4.79
TREC	**41.00**	17.00	GDN Hybrid
LCC	20.11	9.43	Mamba2: 20.68

- Hybrid wins 5/11 tasks overall; it also wins HotpotQA and MultiFieldQA.
- QA and some summarization tasks benefit; code is not a uniform strength.

Paper, Table 3, PDF p. 9. Selected task columns; task-specific scores, all higher is better.

Long inputs amortize the extra inference machinery



At 128K input

Prefill: 5.4× faster than Transformer.

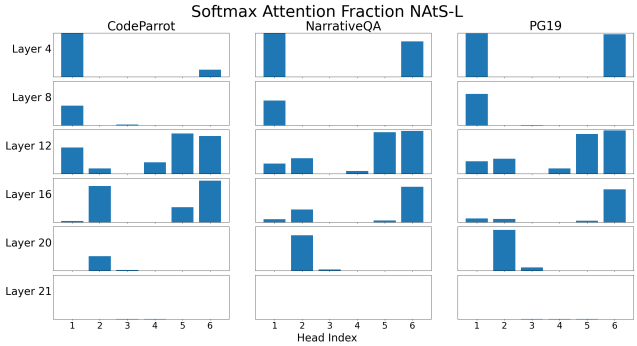
Still **1.66**× GDN's prefill latency.

Decode: 2.3× faster than Transformer.

At short inputs, Hybrid decoding is slower in the plotted comparison.

Paper, Fig. 4, PDF p. 8; ratios reported in Sec. 5.1, p. 7. Prefill: seconds; decode: milliseconds. Full timing protocol is not specified.

Routing adapts to the dataset, layer, and head group



Text datasets look similar; code changes some head assignments.

Several head groups are almost entirely linear; others mix memory types.

Paper, Fig. 5, PDF p. 8: NAtS-L Hybrid. Columns: datasets; rows: layers; x-axis: 6 head groups; bar height: softmax fraction.

Ablations support the complete two-path design

Variant	Retrieval mean ↑	Change
Default Hybrid	**33.49**	—
Only softmax intra-chunk output	30.87	−2.62
Only GDN intra-chunk output	26.98	−6.51
No decay across softmax blocks	32.24	−1.25
Normalize only after summation	29.59	−3.90
Weights from X instead of q	27.64	−5.85
Chunk size 128 instead of 64	27.85	−5.64

Query-conditioned readout and sufficiently fine block routing both matter.

Paper, Table 4, PDF p. 9. Baseline matches the 380M results in Table 6; it is not the 800M mean of 45.07. Changes computed from table values.

The useful idea is selective explicit memory

1. **Mechanism:** choose storage per block, then read both memories with query-dependent weights.
2. **Evidence:** Hybrid improves retrieval and length extrapolation, with long-input latency gains.
3. **Trade-off:** no fixed KV budget; dual intra-chunk work; task-specific quality differences.

Most informative next comparisons

Control positional encoding and layer layout; sweep KV budgets and chunk size; compare quality at matched latency and memory.

Synthesis of Secs. 4–6. Suggested comparisons are this presentation's analysis.

Paper and discussion guide

Deng, Winje, Fehring, and Lindauer (2026).
Neural Attention Search Linear: Towards Adaptive Token-Level Hybrid Attention Models.

arXiv:2602.03681v2
Author code repository linked by the paper

- Method: Sec. 4, PDF pp. 4–6.
- Main evidence: Tables 1–4 and Figures 3–6, pp. 8–9.
- Additional evidence: Appendices A–C, pp. 15–21.

Main discussion: approximately 25 minutes. Backup slides follow.

Backup: training and head configurations

Setting	380M scale	800M scale
Training tokens	15B	50B
Context length	4,096	4,096
Effective batch tokens	0.5M	0.5M
Model width	1,024	1,536
Warmup tokens	0.5B	1B
GDN heads	6	6
NAtS-L softmax heads	12	18
Softmax heads per routing group	2	3

AdamW; peak LR 3×10^{-4}, initial/final LR 3×10^{-5}; cosine schedule. Short-convolution kernel: 4. Default chunk size: 64.

Paper, Sec. 5 and Appendix A.1, PDF pp. 6, 15. Training: 4 or 8 H100 PCIe 80GB GPUs.

Backup: normalized query weights outperform alternatives

Variant	Retrieval mean ↑	Change
Default: softmax weights from q	**33.49**	—
No attention output weights	32.34	−1.15
Sigmoid weights	30.02	−3.47
Linear weights, no activation	30.57	−2.92
Weights from X	27.64	−5.85

- Separate branch normalization and normalized mixing solve different problems.
- The best mean does not imply winning each retrieval task.
- Intra-chunk ablations change local terms, not the entire historical branch.

Paper, Table 4, PDF p. 9. Changes computed from published means.

Backup: sparse baselines use a different budget protocol

Table 5, 380M	Mean ↑
Transformer	22.99
NSA / MOBA	22.05 / 21.98
GDN SWA	8.53
FlexPrefill	20.71
SnapKV	18.64
AdaKV / PyramidKV	17.96 / 18.41
NAtS-L	28.48
NAtS-L Hybrid	**34.19**

Sparse variants use 50% of the stated computational budget. NAtS-L has no fixed routing fraction.

Reporting discrepancy:
Hybrid NQ is 20.72 here, but 16.53 in Table 6. Means are 34.19 and 33.49, respectively.

Keep comparisons within each table.

Paper, Appendix B and Table 5, PDF pp. 15–16; Table 6, p. 17. The NQ discrepancy is not explained in the paper.

Backup: smaller models preserve the retrieval trend

380M / 15B tokens	Retrieval ↑	RULER 4K	8K	16K
GDN	20.78	0.15	0.07	0.02
Mamba2	19.25	0.09	0.03	0.02
Transformer	21.71	0.19	0.00	0.00
GDN Hybrid	24.97	0.21	0.07	0.00
NAtS-L	27.16	0.19	0.08	0.03
NAtS-L Hybrid	**33.49**	**0.36**	**0.30**	**0.12**

Commonsense remains close: Hybrid 43.04; Transformer 43.26.
The winning LongBench tasks differ from those at 800M scale.

Paper, Tables 6–8, PDF pp. 17–18. Retrieval: percent-scale mean; RULER: 0–1 retrieval-subset mean.

Backup: Mamba2 can also supply the compressed memory

Retrieval task	Mamba2	NAtS-L Mamba2
SWDE	13.77	**32.58**
SQD	**27.92**	21.18
FDA	4.26	**32.30**
TQA	**44.91**	41.35
NQ	9.06	**12.48**
DROP	**15.57**	12.12
Mean	19.25	**25.34**

Mean gain: +6.09 percentage points, with losses on three tasks.

This tests a different recurrent backbone, not a multi-operator search space.

Paper, Appendix C.2 and Table 9, PDF pp. 17–18. All task scores higher is better.

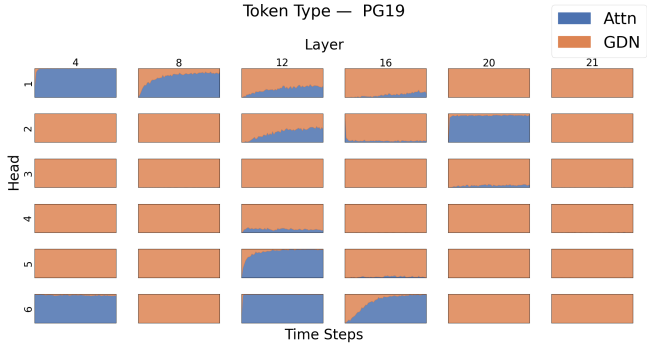
Backup: longer LongBench contexts retain task trade-offs

Context group	Task	Hybrid	Best other model
Table 10	TriviaQA	**32.17**	GDN: 22.43
	GovReport	**13.46**	NAtS-L: 8.07
	LCC	10.28	**Mamba2: 21.68**
Beyond 8K	TriviaQA	**28.60**	NAtS-L: 23.29
	GovReport	**9.10**	NAtS-L: 7.92
	LCC	13.31	**Mamba2: 18.85**

Table 10 is presented as the next bin after 4K, but its caption only says "below 8k." Table 11 is a $>$ 8K group, not a fixed 16K test.

Paper, Appendix C.4, Tables 10–11, PDF pp. 18–19. 800M models, selected task-specific scores, higher is better.

Backup: routing fractions change over time



Each cell is a head-group/layer pair; time runs horizontally.
Blue: softmax fraction. Orange: GDN fraction. Layer 8, head 1 becomes more softmax-heavy.

Paper, Fig. 6, PDF p. 9. This is storage allocation; query-side output weights are a different quantity (Fig. 12, p. 21).

Backup: the mask gradient includes renormalization

For a fixed query, let

$$Z_i = \sum_j e^{A_{ij}} M_{ij}, \quad O_i = \frac{\sum_j e^{A_{ij}} M_{ij} V_j}{Z_i}.$$

Differentiating the numerator and denominator gives

$$\frac{\partial O_i}{\partial M_{ij}} = \frac{e^{A_{ij}}}{Z_i}(V_j - O_i), \qquad \frac{\partial \mathcal{L}}{\partial M_{ij}} = P_{ij}\langle g_i, V_j - O_i\rangle.$$

- The subtraction of O_i comes from normalization.
- Sum over query positions and the KV columns of the same block.
- This differentiates a continuous mask; it does not resolve hard routing by itself.

Teaching derivation of the signal underlying Eq. (15)–(16), PDF p. 5. Inactive mask-score gradients are omitted by the method.