SAS: learn which context blocks help prediction

Simple Attention Sparsification via End-to-End Optimization of Context Ranking

Zhiwei Li, Lei Zhu, Hao Gu, Xiang Hu, Yan Wang, Haitao Mi,
Sirui Han, Leowei Liang, Zhijiang Guo

arXiv:2609.13141v1 | Reading discussion, 18 September 2026

Takeaway

Train block rankings through continuous gates and the LM loss;
use ordinary hard Top-K sparse attention at inference.

Independent reading presentation; approximately 25 minutes + backup.

Paper title and authors: PDF p. 1. Author spelling follows the PDF.

The scarce resource is the decode attention budget

- Dense decoding reads the growing KV context for each new token.
- A block selector ranks history, then retains only K blocks.
- Which blocks best preserve prediction quality under this budget?

$$\underbrace{\mathrm{softmax}(\mathbf{q}\mathbf{K}^{\top})\mathbf{V}}_{\text{dense}} \quad \longrightarrow \quad \underbrace{\mathrm{softmax}(\mathbf{q}\mathbf{K}_{\mathcal{S}}^{\top})\mathbf{V}_{\mathcal{S}}}_{\text{selected tokens } \mathcal{S}}$$

Scope

Post-training adaptation of an existing dense model. The main setting freezes the backbone and trains only a lightweight selector.

Secs. 1, 3 and 5.1, PDF pp. 1–4, 9–10; Eqs. (1)–(3).

Matching attention mass can miss predictive value

Attention distillation

- Match the dense attention distribution layer by layer.
- Hard Top-K blocks the direct LM gradient through indices.

The proposed objective

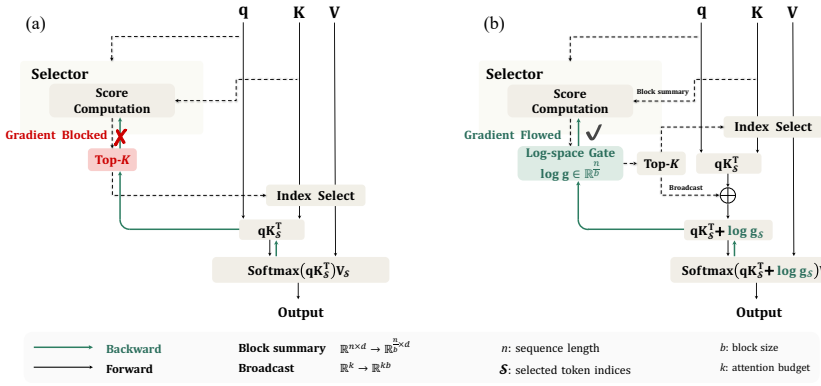
- Learn ranking from the final next-token prediction loss.
- Account for values and interactions across layers.

Controlled comparison

SAS and SeerAttention-R share the AttnGate architecture.
The central change is how the selector is supervised.

Secs. 1–2 and 5.1, PDF pp. 1–4, 9–10. Predictive relevance is the motivation, not an optimality theorem.

A continuous gate supplies the missing gradient path



Hard selection remains; the differentiable path passes through $\log \mathbf{g}$.

Original Fig. 1, PDF p. 2. Green arrows denote backward flow.

Training keeps both hard selection and soft scores

$$\mathbf{s} = \mathcal{R}_\theta\Big(\mathbf{q}, \{\mathbf{K}_{B_m}\}_{m=1}^{C}\Big), \qquad \mathbf{g} = \mathrm{softmax}(\mathbf{s}), \qquad g_0 = 1$$

$$\mathcal{I} = \mathrm{TopK}(\mathbf{g}, K), \qquad \mathcal{S} = B_0 \cup \bigcup_{m \in \mathcal{I}} B_m$$

Training attention

$$\mathbf{o} = \mathrm{softmax}\Big(\mathbf{q}\mathbf{K}_{\mathcal{S}}^{\top} + \log \mathbf{g}_{\mathcal{S}}\Big)\,\mathbf{V}_{\mathcal{S}}$$

B_0: current causal block; B_m: historical blocks of 64 tokens.
Normalize over **all** historical scores, then broadcast selected gates.
Update selector θ using $\nabla_\theta \mathcal{L}_{\mathrm{LM}}$; backbone frozen.

Sec. 4.3, Eq. (9), Algorithm 1, PDF pp. 8–9. Attention scaling is omitted as in the paper.

Inference removes the gate and keeps the learned ranking

$$\mathcal{I} = \mathrm{TopK}(\mathbf{s}, K), \qquad \mathbf{o} = \mathrm{softmax}(\mathbf{q}\mathbf{K}_{\mathcal{S}}^{\top})\mathbf{V}_{\mathcal{S}}$$

- No log-gate bias in inference attention.
- Prefill: dense attention. Decode: select blocks and gather their KV.
- GQA groups share a selection; SGLang uses FlashInfer kernels.
- SAS and SeerAttention-R use the same serving backend.

Interpretation

End-to-end training does not make Top-K indices differentiable.
The continuous training computation is a proxy for discrete inference.

Algorithm 2, PDF p. 9; inference implementation, Sec. 5.1, PDF p. 10.

Inner gates learn relative reallocation of attention

$$\widetilde{p}_i = \frac{g_{b(i)} e^{z_i}}{\sum_j g_{b(j)} e^{z_j}}, \qquad z_i = \mathbf{q}\mathbf{k}_i^\top$$

$$\left.\frac{\partial \mathcal{L}}{\partial g_m}\right|_{\text{inner}} = \sum_{i \in B_m} \frac{\widetilde{p}_i}{g_m}\, d\mathbf{o}^\top (\mathbf{v}_i - \mathbf{o})$$

$$\left.\frac{\partial \mathcal{L}}{\partial g_m}\right|_{\text{outer}} = \sum_{i \in B_m} p_i\, d\mathbf{o}^\top \mathbf{v}_i$$

Inner gating changes normalization; outer gating only rescales values.

$d\mathbf{o}$ is the upstream loss gradient. The term $\mathbf{v}_i - \mathbf{o}$ compares a token's contribution with the current attention output.

Eq. (5), PDF p. 7; Appendix D.1, PDF pp. 21–23.

Normalization calibrates history against the current block

$$\log g_m = s_m - \mathrm{LSE}(\mathbf{s}) \ (m \geq 1), \qquad \log g_0 = 0$$

- The common historical offset does **not** cancel: B_0 gets no offset.
- Softmax gates are invariant to shifting all historical scores.
- Sigmoid can saturate at 1; raw logits can collapse toward 0.
  Both can approach ungated training and learn poor rankings.

Teaching example: equal logits, one token per block

Two historical blocks + one current block:
softmax historical gates $(1/2, 1/2)$ give attention $(1/4, 1/4, 1/2)$.
Raw scores $(0, 0)$ instead give $(1/3, 1/3, 1/3)$.

Eq. (6), Fig. 3, PDF p. 7. Example constructed for explanation; not a paper experiment.

Continuous scores and sparse scope serve different purposes

Preserve ranking: binary STE gates discard score magnitudes.
A dropped token's surrogate factor can be

$$\frac{e^{z_i}}{\sum_{j\in\mathcal{S}} e^{z_j}}, \quad i \notin \mathcal{S},$$

which can grow arbitrarily large; it is not its forward probability.

Limit training scope: compute attention only on selected blocks.

$$ds_m^{\text{sparse}} = -g_m \sum_{\ell\in\mathcal{I}} g_\ell dg_\ell \quad (m \notin \mathcal{I})$$

Unselected blocks still get indirect score updates through normalization, but no independent signal from their own values. Early learning is slower.

Eqs. (7)–(8), PDF p. 8; Appendices D.2–D.3, PDF pp. 23–24.

The ablation isolates three quality-critical choices

Design (one epoch)	GPQA accuracy (%)	Output tokens
Full attention reference	56.1	8547
Inner + softmax + continuous + full	54.4	7109
Outer gate	41.6	13807
Sigmoid activation	17.0	28444
Raw-logit injection	18.8	26401
Binary gate + STE	46.0	7289
Sparse scope (SAS)	**54.8**	7410
Noise-based scope approximation	52.2	7552

Qwen3-4B, budget 2048, GPQA avg@16. Other gate variants use full scope. Standard deviations 2.0–2.8 are omitted in the original ablation table.

Table 1, PDF p. 6. Sparse scope mainly trades early convergence for lower training cost.

Fused training avoids a dense attention matrix

1. Traverse KV tiles; add the block's log-gate inside attention logits.
2. Mask unselected blocks using a per-query threshold; skip empty tiles.
3. Preserve causal current-block attention and online softmax updates.
4. Recompute probabilities in backward; sum token-logit gradients to obtain block-level log-gate gradients.

What the kernel achieves

FlashAttention-style memory efficiency with learnable block gates.
It does not eliminate scoring all candidates or all selection overhead.

Training memory/throughput comparisons are not fully quantified in the paper.

Sec. 4.3, PDF p. 9; Appendix E and Algorithms 3–4, PDF pp. 24–26.

The main evaluation freezes Qwen3 and tests transfer

Training

- Qwen3: 4B / 8B / 14B.
- OpenR1-MATH-220K, 1 epoch.
- AdamW, LR 10^{-3}, cosine.
- Batch 32; max length 32768.
- Block size 64; AttnGate only.

Evaluation

- MATH500, GPQA, AIME24/25.
- LongBench: non-thinking.
- BFCL Multi-Turn, VitaBench.
- Agent context: YaRN to 65536.
- Same trained selector across task families.

Main comparator: SeerAttention-R. Reasoning also includes Sliding Window, StreamingLLM and Quest.

Sec. 5.1, PDF pp. 9–10. The controlled ablation specifically uses 93.7K math examples (p. 6).

GPQA gains are large at a 1024-token budget

Backbone	Full	Seer-R	SAS	Gain (pp)
Qwen3-4B	56.19	39.84	50.41 (2.8)	+10.57
Qwen3-8B	60.54	39.43	53.17 (2.9)	+13.74
Qwen3-14B	65.25	45.64	61.14 (2.9)	+15.50

Accuracy in %; parentheses: reported SAS standard deviation.
At the same budget, MATH500 gains are +5.98 / +7.70 / +6.81 pp.

What this establishes

A better training signal substantially improves tight-budget selection.
SAS still falls below full attention on all three GPQA backbones here.

Table 2, PDF p. 10. Gains computed from the reported means; pp = percentage points.

Harder math improves; large budgets narrow the gap

Budget	Task / backbone	Seer-R	SAS	Gain (pp)
2048	AIME24 / 4B	55.83	68.85	+13.02
2048	AIME24 / 14B	63.65	77.76	+14.11
2048	AIME25 / 8B	43.30	58.41	+15.11
2048	AIME25 / 14B	48.70	64.19	+15.49
4096	MATH500 / 4B	94.10	93.67	−0.43
4096	GPQA / 8B	60.48	60.42	−0.06

SAS does not win every cell. AIME standard deviations are substantial.
At 4096, 4B AIME24 is 71.72 (6.8) vs. dense 71.25;
this small difference does not establish superiority to dense attention.

Table 2, PDF p. 10. Seer-R means above are the original-report rows, not the separate reproduced rows.

LongBench transfer is positive but modest on average

Budget / method	4B Avg.	8B Avg.	14B Avg.
Full	52.2	54.4	56.6
2048 / Seer-R	51.2	52.6	54.9
2048 / SAS	51.5	53.4	55.8
4096 / Seer-R	51.9	53.8	55.7
4096 / SAS	52.0	53.9	56.2

At 2048, the 8K+ bucket gains are $+0.4$ / $+1.7$ / $+2.4$ points.
Math-only selector training transfers to non-thinking long-context tasks.

Source inconsistency

Introduction p. 3 claims $+3.2$ for 4B / 2048 / 8K+.
Table 3 reports $48.8 - 48.4 = +0.4$; this presentation follows the table.

Table 3, PDF p. 10; Sec. 5.2, PDF p. 11.

BFCL improves throughout; VitaBench has exceptions

BFCL Multi-Turn	4B	8B	14B
Full	35.75	42.75	44.50
2048 / Seer-R	29.00	34.38	38.88
2048 / SAS	32.50	36.75	39.50
4096 / SAS	34.38	41.25	44.00

VitaBench, Qwen3-14B:

At 4096, SAS Avg@4 is 34.2 / 28.7 / 12.3 across the three scenarios.

But at 2048, Delivery Avg@4 drops: Seer 32.2 $\rightarrow$ SAS 30.0.

At 4096, OTA Pass@4 is SAS 27.0 vs. Seer 29.0 vs. full 37.0.

Average success and repeated-trial metrics should be kept separate.

Tables 4–5, PDF p. 11. Selected metrics shown; no claim of uniform dominance.

Joint continued pretraining is feasible, not uniformly better

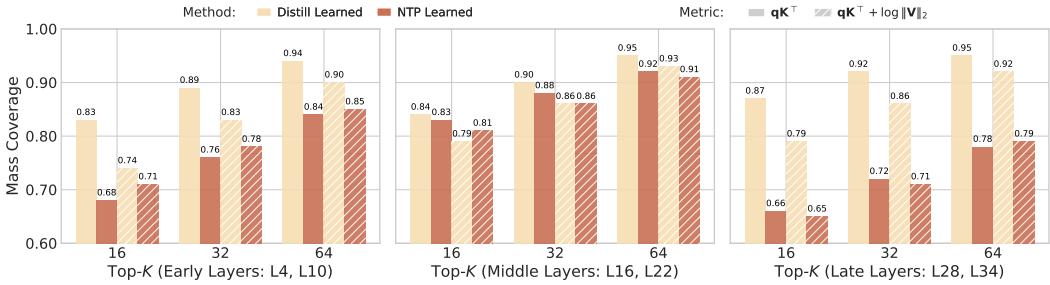
OLMo3-7B; train backbone + a retrieval Q/K selector jointly.
About 50B tokens, 8192 context, 13000 steps, batch 512.

Model	General average	LongBench average
OLMo3-Base	43.88	29.0
OLMo3-512SWA	43.24	28.0
HiLS-Attn-RoPE	41.68	30.0
SAS-RoPE	43.28	30.0

SAS nearly matches SWA on the general average and ties HiLS on LongBench.
Task-level tradeoff: HellaSwag 50.63 is strong, but CRUX 19.25
is below both dense base (24.62) and HiLS (25.62).

Sec. 5.3 and Tables 6–7, PDF pp. 11–12. Selector architecture differs from the main AttnGate setting.

Better task performance can coexist with lower layerwise coverage



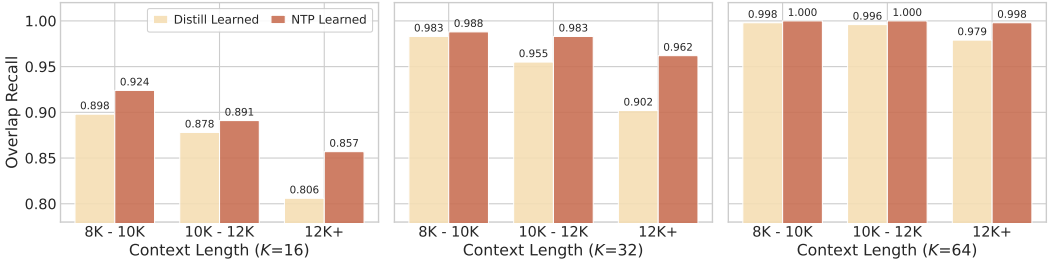
Coverage weights: e^{z_i} or $e^{z_i}\|\mathbf{v}_i\|_2$.

Across early, middle and late layers, SAS generally retains less mass.

Matching dense attention mass is therefore an incomplete selection objective.

Original Fig. 5(a), Eq. (10), PDF p. 13. Distill = Seer-R; NTP = SAS.

Cross-layer recall supports a complementarity explanation



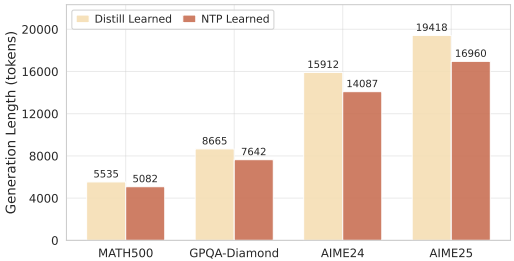
$$\text{Recall} = \frac{|\mathcal{I}_{\text{all}} \cap \mathcal{I}^{\star}_{\text{all}}|}{|\mathcal{I}^{\star}_{\text{all}}|}, \qquad \mathcal{I}_{\text{all}} = \bigcup_{j=1}^{L} \mathcal{I}_j.$$

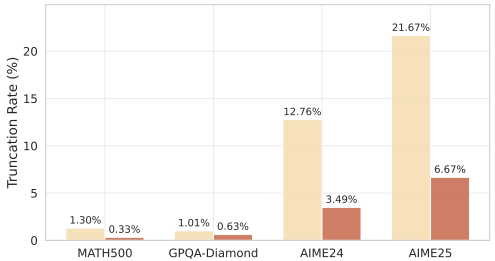
Higher union recall is consistent with complementary choices across layers.

Reading caveat: recall alone does not control union size or prove causality; the oracle derives from dense attention on long training sequences.

Original Fig. 5(b), PDF p. 13; Eq. (11) and protocol, PDF pp. 13–14. GQA oracle uses within-group voting.

Shorter reasoning traces also hit the length cap less often





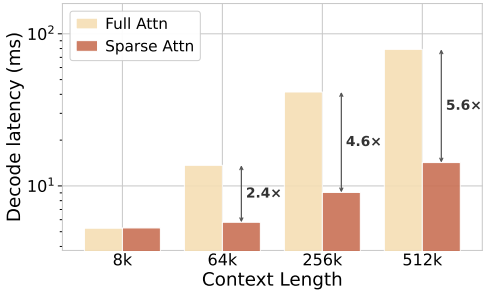
Qwen3-4B, budget 4096, generation cap 32768.

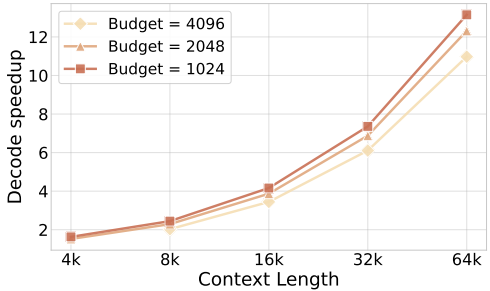
AIME25: 19418 $\rightarrow$ 16960 tokens; truncation 21.67% $\rightarrow$ 6.67%.

Interpret alongside accuracy: shorter output alone is not better reasoning.

Original Fig. 6, Sec. 6.2, PDF p. 14.

Steady-state decode is faster as context grows





Batch 1: 2.4× at 64K, 4.6× at 256K, 5.6× at 512K.

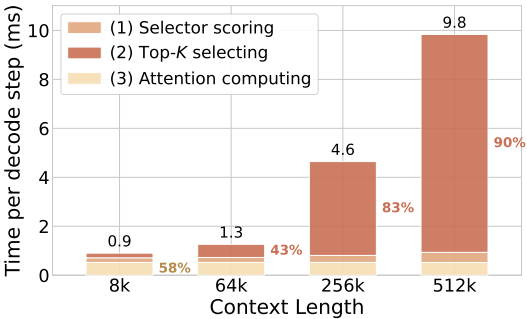
Batch 8: approximately 13× at 64K.

Qwen3-4B, single GPU, SGLang, CUDA graphs; time 512 decode tokens.

Prefill excluded; GPU model not specified. These are not quality tests at 512K.

Original Fig. 7(a,b), PDF p. 15; measurement protocol, Sec. 6.3, PDF p. 14.

Selection becomes the bottleneck at very long context



- Attention reads a fixed KV budget.
- Scoring still scans all block summaries.
- Top-K grows from 21% of this stage at 8K to 90% at 512K.

Fixed selected-attention cost does not imply constant total decode cost.

Original Fig. 7(c), Sec. 6.3, PDF p. 15. Breakdown covers the selector + sparse-attention stage.

Long-context retrieval remains a major limitation

RULER: separate 0.5B-token selector training on packed 64K sequences.
Non-thinking; YaRN extends evaluation to 128K.

Qwen3-14B, budget 4096	Full	Seer-R	SAS
16K	92.91	76.77	83.35
32K	92.15	64.96	66.82
64K	92.19	42.34	45.92
128K	82.23	22.36	29.95

At 128K: $+7.59$ over Seer, but -52.28 relative to full attention.
Authors suggest pooled block summaries lose localized needle-like signals.
This explanation is plausible; the mechanism is not isolated experimentally.

Appendix B, Table 8, PDF p. 21. This is not the main math-only selector training protocol.

The contribution is a better learning signal for sparse routing

1. **Mechanism:** log-space softmax gates give the selector an LM-loss path while retaining Top-K sparsity.
2. **Evidence:** strong tight-budget reasoning gains, plus transfer to long-context and agent tasks.
3. **Boundary:** no universal win, no full recovery of long retrieval, and no constant-cost routing at arbitrary context length.

Discussion

Can richer block summaries improve retrieval without making selection the dominant cost? Can we reduce the train–inference gate mismatch?

Synthesis of Secs. 4–6 and Appendix B. Discussion questions are reading analysis.

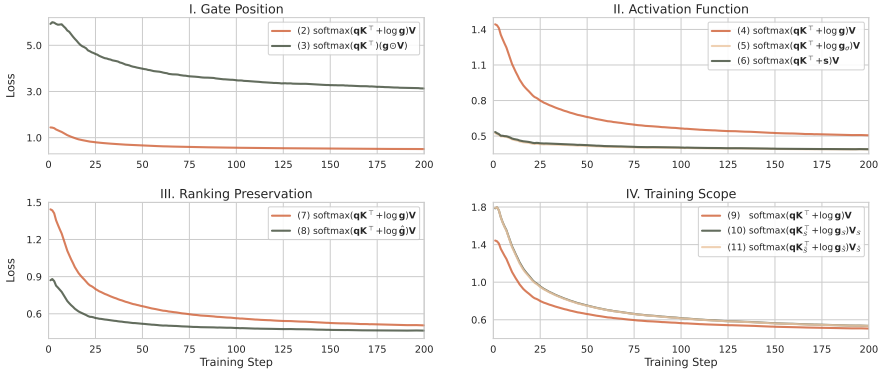
Reference and material provenance

Primary source
Zhiwei Li et al.
SAS: Simple Attention Sparsification via End-to-End Optimization of Context Ranking.
arXiv:2609.13141v1, submitted 11 September 2026.

Read the matching author TeX and 26-page PDF, including Appendices A–E.
Figures are original author assets; numeric tables are selected transcriptions.
The normalization example and methodological caveats are this reading's analysis.

Companion: `reading-notes.zh.md`.
Editable slides: `slides.en.tex`; local build: pdfLaTeX.
No independent model training or benchmark reproduction was performed.

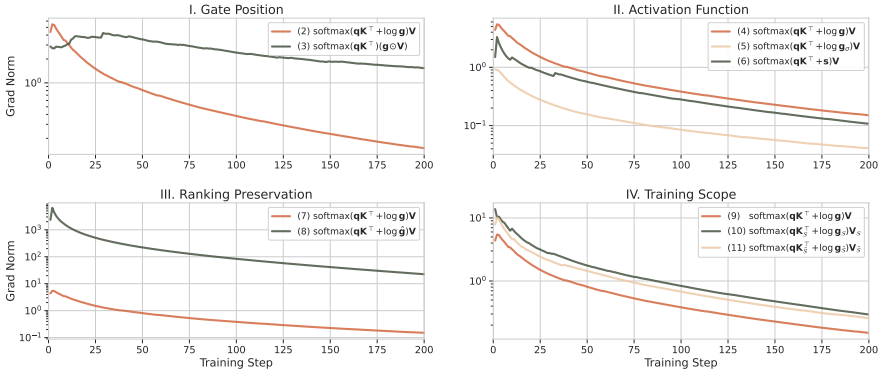
Backup: lower training loss can hide poor discrete rankings



Sigmoid/raw-logit variants can lower loss by approaching ungated attention.
Evaluate the final ungated sparse inference model, not only training loss.

Original Fig. 8(a), Appendix A, PDF p. 20; final ablation results, Table 1, PDF p. 6.

Backup: STE has much larger gradient norms



Panel III: binary STE produces much larger measured gradient norms.
Bounded soft attention probabilities alone are not a global bound on all gate or model gradients.

Original Fig. 8(b), Appendix A, PDF p. 20; gradient analysis, Appendix D, PDF pp. 21–24.

Backup: sparse scope is comparable, not mathematically equivalent

Budget	Backbone / task	Full scope	Sparse scope
1024	8B / GPQA	55.62 (2.8)	53.17 (2.9)
2048	4B / GPQA	54.40 (2.9)	54.86 (2.8)
2048	14B / AIME24	76.67 (6.7)	77.76 (6.6)
4096	14B / MATH500	94.12 (1.0)	95.23 (0.8)
4096	8B / AIME25	64.24 (7.1)	63.41 (7.2)

Most final results are close, with differences in both directions.
This supports the practical sparse-scope choice; it is not a convergence proof.
For unselected blocks, full scope supplies an own-content gradient,
whereas sparse scope supplies only a normalization-mediated update.

Appendix C, Table 9, PDF p. 21. Parentheses: reported standard deviations.

Backup: comparison details that change the interpretation

- Quest at budget 1024: digitized from a prior paper's figure.
- Seer-R and Seer-R#: original-report and reproduced rows; only budget 4096 has a separately listed reproduced row.
- Main reasoning sampling settings are referenced to Seer-R; the SAS text does not fully restate them.
- AIME at budget 1024 is unreported, not zero.
- A *Kb* historical budget coexists with an always-kept current block.
- The abstract page lists Leo Liang; the PDF lists Leowei Liang.

Table 2 and Sec. 5.1, PDF p. 10; Algorithms 1–2, PDF p. 9; identity, PDF p. 1 and arXiv abstract page.