NEURAL CONTEXT COMPILER: LEARNING KV CACHES FOR LONG-CONTEXT LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Documents that are queried many times can be compiled once into a compact key–value (KV) cache that a frozen language model reads at every query. We show that such compaction must alter attention at almost every query and that a training objective by itself bounds the loss only on the inputs it supervises. The leading learned method, self-study, supervises answers to synthetic questions that must first be generated for each document, and hence only the facts those answers state. We introduce *Neural Context Compiler* (NCC), whose refinement supervises a prediction of every document token and generates no text: starting from an Attention Matching (AM) cache, it refines only the compact keys and values by distilling the full-context teacher on teacher-forced continuations of the document. With the hybrid Qwen3.6-27B on 64 held-out papers, a 512-row NCC cache raises constrained recall of masked source numbers from 62.3% to 79.8% over AM and Qasper F1 from 38.1 to 44.2, 28–44 points above StreamingLLM, SnapKV and KVzip. It is statistically indistinguishable from self-study on constrained recall and QA, 11 points better under greedy decoding and 13–25 points better on numbers self-study's answers never state, with 92% less refinement time. On three of four LongBench QA tasks it matches the full-context reader with a 5–11$\times$ smaller state, and on 34K–97K-token documents at 11–30$\times$ compression it retains 74–84% of full-context QA F1 (AM: 19–49%). A fully generation-free variant does as well, an FP4 reader with a 4-bit state keeps its accuracy with a 14$\times$ smaller stored state, and the recall gain replicates on three further hybrids, fading where the teacher misses many numbers.

1 INTRODUCTION

Many applications ask a language model many questions about the same document (e.g., Kočiský et al., 2018; Dasigi et al., 2021; Pang et al., 2022). Serving them either repeats the document's prefill for every question or keeps its key–value (KV) cache resident, whose size grows with the document (64 KiB per token for our main model), even when serving systems page and reuse it (Kwon et al., 2023; Zheng et al., 2024; Gim et al., 2024; Yao et al., 2025) or quantize and encode it (Hooper et al., 2024; Liu et al., 2024a). A frequently queried document can instead be *compiled* once (as in Latent Context Compilation; Li et al., 2026) into a fixed-size state that the frozen model reads at every query (Eyuboglu et al., 2026; Zweiger et al., 2026).

Two lines of work build such states; each leaves a gap. Training-free compaction selects cached entries (Zhang et al., 2023; Xiao et al., 2024; Li et al., 2024; Kim et al., 2025) or, as in Attention Matching (AM; Zweiger et al., 2026), refits compact keys and values to reproduce each layer's attention. It does not consult the model's outputs and can lose the detail a later question needs (Figure 1; Tang et al., 2024; Chen et al., 2025). Learned caches are instead trained through those outputs (Li & Liang, 2021). The leading method, Cartridges (Eyuboglu et al., 2026), distills a full-context teacher on *self-study*: synthetic questions about the document and the teacher's answers. Each document then needs its own question bank, generated before the first gradient step, and only the facts those answers state are directly supervised (Figure 4).

Our key idea is that the document can supervise its own compact cache. *Neural Context Compiler* (NCC) initializes a compact cache with AM and refines only its keys and values by distilling the full-context teacher on continuations of the document (Figure 3). Teacher and student read the same

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

cue, up to 32 source tokens preceding a document span, and are teacher-forced through that span: every position yields a full-vocabulary teacher distribution (Hinton et al., 2015) for a forward-KL loss (Kim & Rush, 2016), and no text is generated. The compiled state is *native*: it keeps the model's own cache format and leaves the model unchanged.

Our analysis motivates this design (Section 3). Compaction alters attention at almost every query (Proposition 1), so the objective decides where the state stays accurate, yet by itself it bounds the loss only on inputs its supervision covers (Proposition 2): source continuations supervise a prediction of every document token, self-study only its generated answers. Below a sharp, machine-checked KL margin, the student provably keeps the teacher's top next token (Theorem 1).

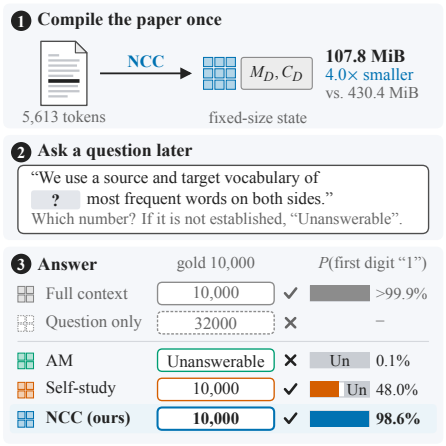


Figure 1: **A number-recall question.** AM abstains under greedy decoding; NCC recovers the number. Bars: first-token probability of the gold digit (color) and of "Un" (gray).

Our main model is Qwen3.6-27B (Qwen Team, 2026b), a hybrid (De et al., 2024; Lieber et al., 2024; Ren et al., 2025; NVIDIA et al., 2025b) of linear-recurrent Gated DeltaNet (Gu & Dao, 2024; Yang et al., 2024a;b; 2025b) and gated attention layers (Qiu et al., 2025). Its KV cache is the only state that grows with the document and carries exact retrieval (Afendulev et al., 2026; Arora et al., 2024a;b). On 64 held-out Qasper papers (Dasigi et al., 2021), an NCC cache of 512 rows per KV head raises recall of masked source numbers under number-constrained decoding from 62.3% for AM to 79.8%, 28–44 points above training-free selection. It reaches self-study's accuracy at 8% of its refinement time (Figure 2) and, as Proposition 2 suggests, leads it by 13–25 points on numbers that self-study's answers never state. On 34K–97K-token documents at 11–30× compression, it retains 74–84% of full-context QA F1, against 19–49% for AM.

In summary, we train a native compact cache on the document's own text rather than on generated conversations, explain with standard arguments where each kind of supervision can help, and show that NCC improves on AM at every tested budget (128–2048 rows) and document length, on four LongBench QA tasks (Bai et al., 2024) and, in constrained recall, on three of four further hybrids.

2 Neural Context Compiler

Problem. A context compiler maps a document D to a reusable artifact $A(D)$ of fixed size; a frozen reader f answers any later question q as $f(A(D), q)$. Compilation never sees evaluation questions, answers or annotations. Qwen3.6-27B has 16 attention layers with 4 KV heads each (Ainslie et al., 2023) and 48 Gated DeltaNet layers with a fixed-size recurrent and convolutional state (Schlag et al., 2021; Yang et al., 2024b). The artifact, which we call the compact state, is a native hybrid state

$$A_\theta(D) = \big(K_\theta,\, V_\theta,\, \beta,\, M_D,\, C_D\big), \tag{1}$$

where K_θ, V_θ hold 512 rows (key–value pairs) per KV head in each attention layer, β holds AM's per-row biases, which correct attention mass, and M_D, C_D are the recurrent and convolutional states after reading D: 107.8 MiB in total (32 MiB of KV rows), against 225–648 MiB for the full context of our papers. All compact states we compare share M_D, C_D, so differences isolate the KV rows.

Stage 1: Attention Matching. AM fits compact keys, values and biases β so that attention outputs over the compact cache reproduce those over the full cache on a set of reference queries (Zweiger et al., 2026). We run its query-agnostic recipe with the authors' OMP code at 512 rows per KV head (Appendix O.2); its reference queries come from 9.7K model-generated tokens per document, shared by every refined state. With AM's repeat-prefill variant, whose reference queries come from the document itself, nothing is generated and NCC scores the same (78.6% constrained recall and 45.5 F1, against 78.6% and 46.2; Appendix D).

Stage 2: source-continuation distillation. We split the document into 128 contiguous spans that cover every token. For span b with target tokens $t_{b,1:L_b}$, the request x_b quotes a cue of up to 32

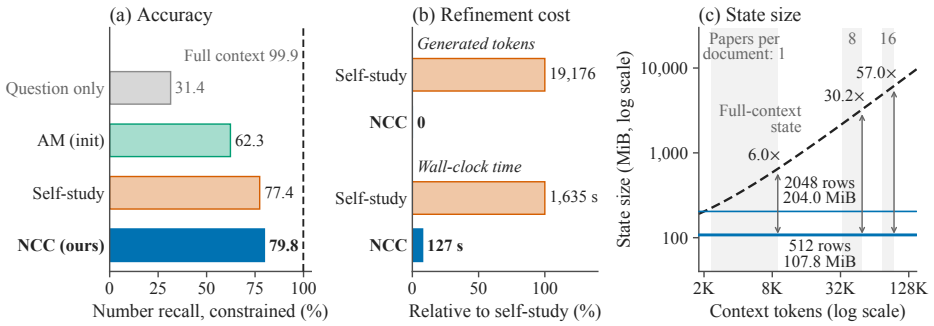


Figure 2: **NCC reaches self-study's accuracy at 8% of its refinement time, in a state that does not grow with the document.** (a) Constrained recall of masked source numbers on 64 held-out papers; the question-only reader sees no document. (b) Refinement cost per paper, relative to self-study (labels: absolute values); NCC generates no text. (c) The full-context state grows by 64 KiB per token; the compact state is fixed at 107.8 MiB (512 rows per KV head). Bands: tested lengths; arrows: compression at the longest input.

preceding source tokens and asks the model to continue the document. Teacher and student receive the same history $h_{b,t} = (x_b, t_{b,<t})$; the teacher reads the full-context state and the student the compact state. With next-token distributions $P_D(\cdot \mid h)$ and $Q_\theta(\cdot \mid h)$ over the vocabulary, we minimize

$$\mathcal{L}_b(\theta) = \frac{1}{L_b} \sum_{t=1}^{L_b} D_{\mathrm{KL}}(P_D(\cdot \mid h_{b,t}) \,\|\, Q_\theta(\cdot \mid h_{b,t})) \tag{2}$$

over the 451 rows per KV head that hold document tokens; the 61 template rows (system prompt and delimiters), the backbone, β, positions and M_D, C_D stay fixed. We train for two epochs (256 AdamW updates (Loshchilov & Hutter, 2019), learning rate 0.01).

Self-study control. Our main comparison generates 256 factual questions from eight 2,048-token document windows, lets the full-context teacher answer each, and distills on the conversations with Equation (2), a form of context distillation (Askell et al., 2021; Snell et al., 2022) in the style of Cartridges. It shares the initialization, trainable rows, optimizer and reader with NCC and takes one conversation per update; we report its 256-update checkpoint and an *exposure-matched* one (256–466 updates) whose input tokens, target tokens and attention operations match or exceed those of NCC.

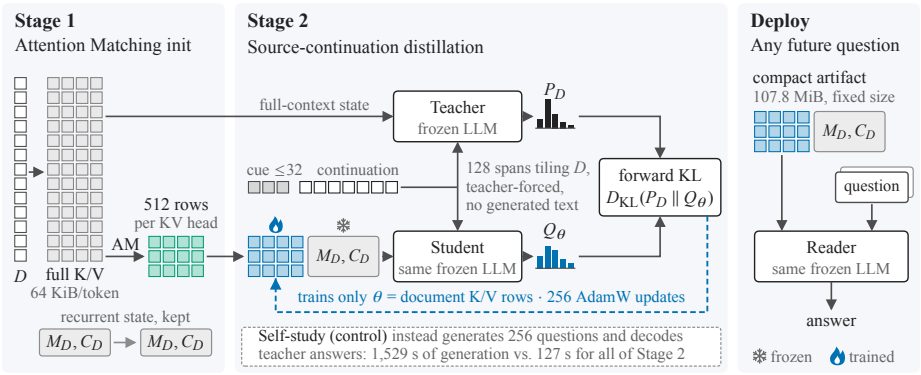


Figure 3: **Neural Context Compiler.** *Stage 1:* Attention Matching compacts each attention layer to 512 rows per KV head; the recurrent state M_D, C_D is carried over. *Stage 2:* a frozen full-context teacher and the compact-cache student read the same cue from the source and are teacher-forced through the next source tokens; forward KL between their next-token distributions updates only the compact keys and values, and no text is generated. *Deploy:* the compact state answers future questions with the unchanged model.

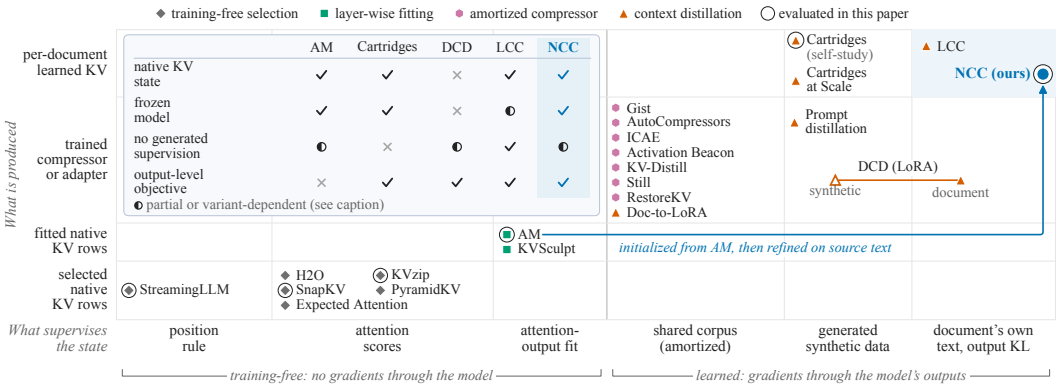


Figure 4: **NCC carries Attention Matching's native KV rows into end-to-end training on the document's own text.** Methods are placed by what supervises the compact state (columns) and what is produced (rows); the darker line separates methods fitted without the model's outputs from those trained through them. Ringed: run here. *Inset:* closest designs (half marks: variant-dependent). Citations: Section 6 and Table 41.

3 What a compact cache can preserve

A compact state affects an answer only through the next-token distributions it induces in the frozen reader; the results below make predictions that Section 5 tests. Proofs, numerical checks and further discussion are in Appendices T and U.

Setup. A *history* $h \in \mathcal{H}$, the finite set of token sequences up to a maximum length, is the token sequence read after the document: a request followed by any tokens already produced. For an artifact A (Equation (1)), $Q_A(\cdot \mid h)$ is the reader's next-token distribution over the vocabulary $\mathcal{V}$, $P_D(\cdot \mid h)$ that of the full-context state, and $\ell_A(h) = D_{\mathrm{KL}}(P_D(\cdot \mid h) \| Q_A(\cdot \mid h))$ the loss at h. A *supervision distribution* ν over histories defines the objective $L_\nu(A) = \mathbb{E}_{h\sim\nu}\ell_A(h)$, and a *deployment distribution* μ is the law of the histories at which the deployed reader chooses tokens.

Equation (2) puts positive probability on the history before every source token; self-study puts it on the answers of its generated conversations.

Proposition 1 (Compaction is lossy almost everywhere)**.** *Let an attention head's full cache have n rows with pairwise distinct keys and a compact cache $m < n$ rows, each row with a finite bias b_i (0 in an ordinary cache, β in AM). The set of queries at which their attention masses $Z(q) = \sum_i e^{\langle q, k_i\rangle + b_i}$ agree is closed and Lebesgue-null, and, for key dimension $d \geq 2$, also null on every ellipsoid $\{Mv : \|v\| = 1\}$ with M invertible, on which RMS-normalized queries lie. At every other query, given the same appended rows, the share of attention that the head places on the cached rows differs from the full cache's; under selection the compact mass is strictly smaller at every query.*

Proposition 2 (Transfer by change of measure; Shimodaira, 2000; Donsker & Varadhan, 1975)**.** *Fix A and let μ, ν be distributions on $\mathcal{H}$. (i) If $\operatorname{supp}\mu \subseteq \operatorname{supp}\nu$, then $\mathbb{E}_\mu \ell_A \leq \rho\, L_\nu(A)$ with $\rho = \max_{h:\nu(h)>0} \mu(h)/\nu(h)$. (ii) For every $\lambda > 0$, $\mathbb{E}_\mu \ell_A \leq \lambda^{-1}[D_{\mathrm{KL}}(\mu\|\nu) + \log \mathbb{E}_\nu e^{\lambda \ell_A}]$. (iii) Over unrestricted predictors $h \mapsto Q_h$, if $\mu(S) > 0$ for a set S with $\nu(S) = 0$, no bound of this kind holds: some predictor has $L_\nu = 0$ and arbitrarily large $\mathbb{E}_\mu \ell$.*

Evaluation prompts are supervised by neither method, so Proposition 2 guarantees no test answer; applied to facts rather than histories (a heuristic step), it suggests that a method gains least on facts its supervision never states, which Section 5.4 tests.

Lemma 1 (Chain rule for KL divergence; Cover & Thomas, 2006)**.** *For strictly positive P, Q on $\mathcal{V}$ and a nonempty proper subset $G \subset \mathcal{V}$ with $p = P(G)$, $q = Q(G)$ and conditionals P^G, Q^G,*

$$D_{\mathrm{KL}}(P\|Q) = d(p\|q) + p\, D_{\mathrm{KL}}(P^G\|Q^G) + (1-p)\, D_{\mathrm{KL}}(P^{G^c}\|Q^{G^c}), \tag{3}$$

where d is the binary KL divergence.

Theorem 1 (Sharp decision margin)**.** *Let P be strictly positive with unique maximizer y and runner-up probability b, and $a = P_y$. Define*

$$\tau(P, y) = a \log \frac{2a}{a+b} + b \log \frac{2b}{a+b}. \tag{4}$$

	Single papers 2.3K–9.1K tokens			8-paper documents 34K–51K tokens				16-paper documents 76K–97K tokens			
Rows per KV head	512			512		2048		512		2048	
Compression, total	3.7×			26.1×		13.8×		50.5×		26.7×	
Compression, K/V	10.3×			85.5×		21.4×		167.6×		41.9×	
State	Greedy	Constr.	QA	Constr.	QA	Constr.	QA	Constr.	QA	Constr.	QA
Full context	99.7	99.9	45.9	99.6	46.2	99.6	46.2	99.9	44.1	99.9	44.1
Question only	31.2	31.4	–	–	–	–	–	–	–	–	–
Training-free compact states											
Recurrent state only†	0.9	22.3	10.0	23.2	8.7	23.2	8.7	21.4	8.7	21.4	8.7
StreamingLLM‡	14.6	35.7	22.9	23.1	9.3	29.1	11.0	22.3	8.7	26.0	10.3
SnapKV	11.3	36.8	21.8	23.7	11.5	28.8	14.0	22.3	10.4	25.9	10.9
KVzip	9.4	52.2	18.2	21.6	8.7	32.0	8.7	20.5	8.7	20.9	8.7
Layer-wise fitting (initialization)											
AM	16.7	62.3	38.1	26.3	9.0	41.1	22.7	25.8	8.7	32.7	11.5
AM + learned refinement											
Self-study, 256 updates	65.1	77.4	43.4	36.5	26.1	50.1	37.2	30.9	**23.3**	39.4	31.0
Self-study, exp.-matched	65.8	77.9	43.3	–	–	–	–	–	–	–	–
NCC	**76.4**	**79.8**	**44.2**	**37.9**	**34.2**	**57.6**	**38.6**	**33.0**	21.8	**44.0**	**34.0**
NCC − AM	+59.7	+17.5	+6.1	+11.6	+25.2	+16.5	+15.8	+7.3	+13.1	+11.4	+22.5

Table 1: **Results across all document lengths.** Paper-averaged number recall (%) under greedy and number-constrained (Constr.) decoding, and Qasper F1 (QA), on the 64 held-out papers read singly or inside 8- and 16-paper documents. All compact states keep the same rows per KV head and the recurrent state; compression: median full over compact bytes, for the whole state (total) or the KV rows (K/V). **Bold**, underline: best and second compact state; –: not run. Last row: paired gain of NCC over AM; all eleven 95% bootstrap intervals exclude zero. †No document rows. ‡Attention sinks (Xiao et al., 2024) are the template rows.

(i) If $D_{\mathrm{KL}}(P\|Q) < \tau(P,y)$*, then* $Q_y > Q_j$ *for every* $j \neq y$*. (ii)* $\tau(P,y)$ *is the minimum of* $D_{\mathrm{KL}}(P\|Q)$ *over the* Q *under which some* $j \neq y$ *ties or beats* y*. (iii)* $(a-b)^2/(2(a+b)) \leq \tau(P,y) \leq (a-b)^2/(a+b)$*, and* $\tau(P,y) < \log 2$*.*

Corollary 1 (Whole answers)**.** *If* $\ell_A(h_t) < \tau(P_{h_t}, y_t)$ *at every history* h_t *along the teacher's greedy answer* $y_{0:T}$ *(each* y_t *a strict maximizer), the student's greedy answer is identical. Under constrained decoding with grammar-permitted sets* G_t*, along the teacher's constrained answer (strict maximizers within each* G_t*), it suffices that* $D_{\mathrm{KL}}(P^{G_t}_{h_t}\|Q^{G_t}_{h_t}) < \tau(P^{G_t}_{h_t}, y_t)$ *at every* t *with* $|G_t| \geq 2$*.*

Lemma 1 separates the mass a student moves out of the permitted set, for example onto an abstention, from its ranking within the set, which is all a constrained decoder reads. Theorem 1 is tight; (i), the minimum in (ii) and the induction behind Corollary 1 are machine-checked in Lean 4 (Appendix T.10).

4 Experimental setup

We use 64 papers from the Qasper training release (Dasigi et al., 2021) (2,329–9,101 tokens), disjoint from our 12 development papers. *Number recall*, a cloze-style probe (Petroni et al., 2019), masks one of up to 16 numbers per paper in its sentence and asks for it (916 questions; Figure 1), scored under greedy decoding and under a number-constrained grammar (Geng et al., 2023; Dong et al., 2025); the reader may answer `Unanswerable`. Number recall resembles NCC's source continuations, *document QA* (all 234 Qasper questions, answer F1) resembles self-study's question–answer conversations, and QuALITY's multiple-choice answers share neither format. We compare NCC with AM, the two self-study checkpoints, three training-free selections, question-free as compile-once requires, at the same uniform per-head budget (StreamingLLM (Xiao et al., 2024), SnapKV (Li et al., 2024) and KVzip (Kim et al., 2025)), and a recurrent-state-only control without document rows; full-context and question-only readers serve as references. Scores are paper-averaged, with

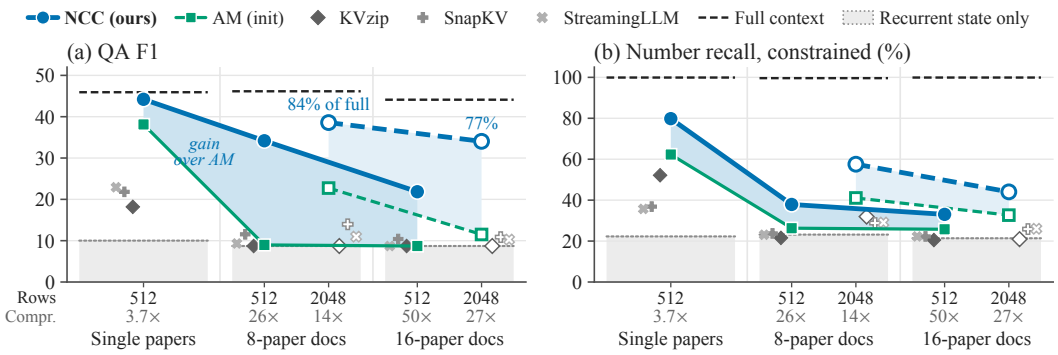


Figure 5: **Training-free selection falls to near the recurrent-state-only control on long documents; NCC stays far above it.** Solid, filled: 512 rows per KV head; dashed, hollow: 2048 rows; blue shading: NCC's gain over AM. Under each column: rows and median total-state compression.

paired 95% bootstrap intervals (Efron, 1979) over papers, 8-paper documents, QuALITY articles or LongBench stories and documents (10,000–20,000 resamples). Details are in Appendices D, H, O and Q; Table 4 guides the additional studies.

5 Results

5.1 Source continuations are enough supervision

Table 1 summarizes all settings. NCC improves its AM initialization on every metric: constrained recall rises by 17.5 points [14.7, 20.5], greedy recall by 59.7, and QA F1 by 6.1 [2.4, 9.7] (Appendix L.1). A question-only reader recalls 31.4% of the numbers; on the 636 it misses, NCC still gains 23.0 points over AM (Appendix Q.5). Against self-study, NCC is statistically indistinguishable under constrained decoding ($+2.4$ $[-0.5, 5.3]$ and $+1.9$ $[-1.2, 4.9]$ against its two checkpoints) and on QA, and better by 10.5–11.2 points under greedy decoding, where self-study abstains on 211–215 of the 916 questions and NCC on 65; its refinement generates no tokens (self-study: 19.2K per paper) and takes 7.7% of self-study's time (Appendix L.4).

The KV rows carry the document. With the recurrent state alone the model recalls 22.3% of the numbers under constrained decoding and reaches 10.0 F1, as exact retrieval passes through attention (Jelassi et al., 2024). Selecting rows is not enough either: StreamingLLM, SnapKV and KVzip reach 35.7–52.2% recall and 18.2–22.9 F1, 28–44 points below NCC on recall, and their adaptive-budget versions (Feng et al., 2025; Cai et al., 2025) stay 23–47 points below (Appendix H.1); selected rows under-count attention mass at every query (Proposition 1).

5.2 Gains across budgets, models and precisions

At 128, 256, 512 and 1024 rows per KV head, NCC improves on AM by 18.4, 18.6, 17.5 and 10.1 points of constrained recall and by 14.3, 13.6, 6.1 and 3.6 F1, all with intervals above zero (Figure 17 in Appendix N). At 1024 rows NCC reaches 93.7% recall and 46.3 F1, on par with full context in F1 (45.9). The recipe also improves on AM in three further hybrids (Table 2b), by 10.2 points of constrained recall on Qwen3.5-9B (Qwen Team, 2026a) and by 7.8 and 9.7 on the Mamba-2 hybrids Nemotron 3 Nano 4B and Nano 9B v2 (NVIDIA et al., 2025a; NVIDIA, 2025). On Olmo Hybrid 7B (Merrill et al., 2026), whose teacher recalls only 54% of the numbers, it reduces abstentions but not errors (Table 7). Stored in NVFP4, the whole NCC state is 13.7× smaller than the BF16 full state (the full state in NVFP4: 3.6×) with no significant change in any score, also when NVIDIA's NVFP4 Qwen3.6-27B (emulated) reads it; that reader keeps NCC's gain over AM ($+17.0$ points; Appendices E and E.1).

5.3 Gains persist on long documents

We concatenate the 64 papers, in a fixed order, into 8 documents of 8 papers (34K–51K tokens) and 4 of 16 papers (76K–97K), reuse every question, and keep the recipe unchanged (Appendix I). Training-free selection collapses (Figure 5); NCC keeps 84% and 77% of full-context QA F1 at 2048

(a) LongBench QA, official F1

State	NarrQA	MFQA	HotpotQA	2Wiki
Full context	27.6	57.6	55.7	58.3
StreamingLLM	4.9	9.1	6.1	1.0
SnapKV	8.3	8.6	14.8	3.0
KVzip	1.5	6.6	3.3	1.0
Attention Matching	23.6	31.6	33.2	24.5
Self-study	27.8	–	–	–
NCC	**28.3**	**44.0**	**54.6**	**57.6**
State vs. full	11.4×	5.1×	6.5×	5.1×

(b) Other hybrids, constr. recall

Model	Full	AM	NCC
Qwen3.6-27B	99.9	62.3	**79.8**
Qwen3.5-9B	99.0	59.3	**69.6**
Nemotron 3 Nano 4B	67.4	25.8	**33.6**
Nemotron Nano 9B v2	67.0	29.6	**39.3**

(c) Serving, 16 papers

	Prefill	Cached	NCC
TTFT (s)	11.1	0.22	**0.12**
State (GiB)	–	5.3	**0.11**
States per GPU	–	7	**361**

Table 2: **Beyond single Qasper papers.** (a) Four LongBench tasks (Bai et al., 2024) (NarrativeQA: 200 questions on 20 stories of 8.7K–67.7K tokens; the others: 335 questions on 100 documents each); states are compiled once without the question, at 512 rows per KV head (2048 from 16K tokens); last row: full over compiled state bytes. (b) Constrained recall on the 64 papers at 512 rows (Gated DeltaNet: Qwen; Mamba-2: Nemotron; Appendix B); on Olmo Hybrid 7B, whose full-context reader recalls 54%, and on the pure-attention Qwen3-8B, NCC does not significantly improve on AM (Table 7). (c) A 16-paper document on one GH200 (Appendix F): time to first token with per-query prefill, the cached full state or the 512-row NCC state; resident state; copies that fit next to the weights.

rows and 74% on 8-paper documents at 512 rows, against 19–49% for AM, and even at 45–57× (16-paper documents, 512 rows) it keeps 50% against AM's 20%. It gains 7.3–16.5 points of recall and 13.1–25.2 F1 over AM in every setting. Self-study, run unchanged on the long documents, remains strong; NCC is never significantly worse, and is better on recall at 2048 rows (+7.5 and +4.6) and on QA for 8-paper documents at 512 rows (+8.1 F1), at 13–17% of its refinement time (Appendix J). On four standard LongBench QA tasks (Table 2a, Figure 9), NCC is statistically indistinguishable from the full-context reader on NarrativeQA, HotpotQA and 2WikiMQA with a 5.1–11.4× smaller state (and from self-study on NarrativeQA at a tenth of its refinement time), and improves on AM by 4.7–33.1 F1 on every task (Appendices A and A.1). A compiled 16-paper document answers after 0.12 s instead of 11.1 s, 361 compiled states fit on one GPU against 7 full ones (Table 2c), and compilation repays its GPU time within 224 queries (Appendix F).

5.4 Why source continuations work

Section 3 separates two failure modes, and the data show both. The first is *coverage* (Proposition 2). Although self-study's eight windows span every token of a single paper, 292 of the 916 masked numbers appear in none of its 256 teacher answers. On these, self-study gains only 3.8 points of constrained recall over AM, against 20.7 on the other 624, whereas NCC gains 16.8 and 18.3; NCC leads self-study by 25.0 points [17.0, 33.5] under greedy and 13.0 [7.4, 18.6] under constrained decoding, against +4.5 and −2.4 where self-study states the number (Figure 6d). Long documents give a test whose predictions we fixed before any self-study output: inside self-study's eight windows self-study matches or beats NCC, outside them NCC leads by 4–12 points, and the difference-in-differences (DiD) is +9.1 to +15.1 points with every interval above zero, while NCC's gain over AM changes by at most 4.5 points (Figure 7).

The second is *abstention* (Lemma 1). AM puts a mean first-token probability of 0.80 on "Un", the first token of `Unanswerable`, against 0.24 for self-study and 0.08 for NCC; this mass lies outside the permitted number tokens, so a constrained decoder discards it and lifts AM from 16.7% to 62.3%, and every answer it repairs was a greedy abstention (Appendix L.2). Whether a repaired answer is right depends on the ranking within the number tokens, where NCC agrees with the teacher's first token on 90.5% of questions against 79.4% for AM. An exact number is a path of such rankings (Corollary 1), and NCC's lead over AM grows with its length (post hoc; Appendix L.2).

5.5 Which parts of NCC matter

A local teacher is harmful (Table 3): a teacher that reads only a local window, whose top token is the source token only 34–45% of the time (97.6% for the full-context teacher), trains the cache to abstain

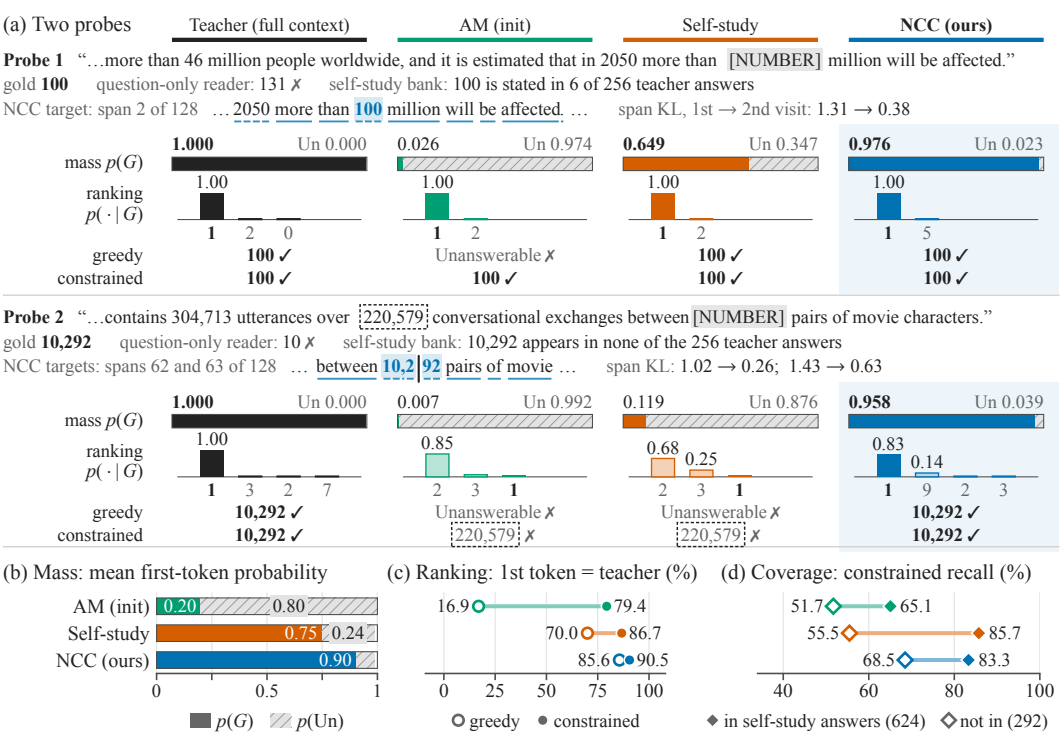


Figure 6: **A compact cache fails on a number by moving first-token mass onto abstention or by ranking the wrong number first.** (a) Two probes from one paper (hatched: mass on "Un"; span KL: NCC's loss on the span holding the number, epochs 1 and 2). (b–d) All 916 questions: first-token mass, agreement with the teacher, and constrained recall on numbers that self-study's answers state (624) or not (292).

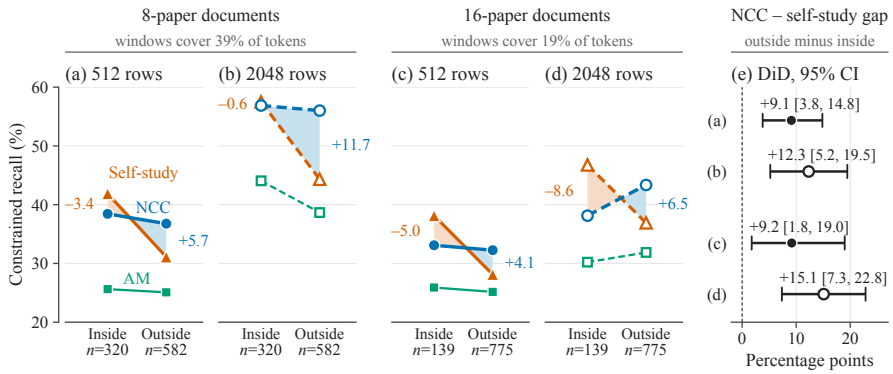


Figure 7: **On long documents, self-study helps where its generated questions go; NCC helps everywhere.** (a–d) Constrained recall inside and outside the eight 2,048-token windows from which self-study generates its questions (solid: 512 rows; dashed: 2048); shading marks the leader, numbers give NCC minus self-study. (e) Difference-in-differences, (NCC − self-study) outside minus inside, as fixed in advance, with 95% bootstrap intervals over papers (Appendix J).

and ends below AM. *The document itself is the supervision*: with a full-context teacher the targets are nearly one-hot, and cross-entropy on the source tokens, which needs no teacher, reaches the same accuracy as KL. *Initialization is what makes this supervision work*: the same refinement from StreamingLLM or KVzip rows stays 37 and 7 points below NCC. From StreamingLLM rows it ends below AM, echoing Eyuboglu et al. (2026), who found next-token training from a corpus's first tokens uncompetitive. At 1,024 updates NCC reaches 84.1%, 5.2 points [2.0, 8.6] above self-study trained for as many updates, which does not exceed NCC at 256 (Appendix L.1); halving or doubling the spans, or doubling the cue, moves the default's constrained recall by at most 1.3 points (Appendix G).

Variant	Change	Accuracy ↑ Greedy	Constr.	QA F1	Δ constr. vs. NCC	
NCC	–	76.6	79.3	45.9		
AM, no refinement	–	16.4	61.7	38.5	−17.6	
Hard-label targets	CE on source tokens	77.2	78.7	45.3	−0.7	[−2.6, 1.2]
Local teacher	cue + preceding span	0.0	20.8	21.6	−58.5	[−63.4, −53.7]
Local teacher	≥1,024 preceding tokens	1.1	40.7	31.6	−38.7	[−42.8, −34.6]
StreamingLLM init	instead of AM	36.3	42.6	27.5	−36.8	[−41.0, −32.7]
KVzip init	instead of AM	59.4	72.6	34.8	−6.7	[−10.1, −3.3]
1,024 updates	4×	80.1	84.1	46.1	+4.8	[2.2, 7.5]

Table 3: **Ablations** (64 papers, 512 rows). Each variant changes one component; differences are paired against NCC rebuilt in the same run (95% bootstrap intervals). Local teachers see a local window, like that of Deep Context Distillation (DCD; Caccia et al., 2025).

5.6 Single decisions largely survive compaction

On QuALITY (1,155 questions on 64 articles; Appendix K), all fitted 512-row states keep most comprehension with a letter-constrained decoder (AM 84.9%, NCC 85.1%, full context 90.6%; NCC−AM +0.2 [−1.6, 2.0]). Under greedy decoding NCC again removes AM's abstentions (14.0% to 1.4%; +8.8 points [6.0, 11.8]). Theorem 1 explains why single decisions survive: a letter is one decision, and every fitted state's letter distribution lies within the teacher's margin on 84% of questions, which provably keeps that letter; an exact number needs every token right (Corollary 1).

6 Related work

KV cache compression. Training-free methods keep entries chosen by position (Xiao et al., 2024), attention statistics (Zhang et al., 2023; Liu et al., 2023; Li et al., 2024; Cai et al., 2025; Devoto et al., 2025), per-head budgets (Ge et al., 2024a; Feng et al., 2025) or context reconstruction (Kim et al., 2025), quantize them (Liu et al., 2024b; Hooper et al., 2024), or select entries per query (Tang et al., 2024; Chen et al., 2025); AM and KVSculpt fit compact keys and values to each layer's attention (Zweiger et al., 2026; Jiang & Jin, 2026).

Learned context representations and context distillation. Soft-token and KV compressors (Mu et al., 2023; Chevalier et al., 2023; Ge et al., 2024b; Zhang et al., 2025; Chari et al., 2025; O'Neill et al., 2026; Baek et al., 2026) train shared modules, and prompt compression and retrieval shorten or select the text (Jiang et al., 2023; Xu et al., 2024; Lewis et al., 2020). Context distillation transfers a context-conditioned teacher into a student (Askell et al., 2021; Snell et al., 2022; Kujanpää et al., 2025); test-time training and continued pretraining instead write documents into weights (Wang et al., 2026; Yang et al., 2025c; Tan et al., 2024). Eyuboglu et al. (2026) distill self-study conversations into a per-corpus KV cache, and Hardalov et al. (2026) suggest compacted caches as its initialization, which our self-study control and NCC adopt. Caccia et al. (2025) distill a local-chunk teacher into per-document LoRA modules (Hu et al., 2022), a teacher that hurts a compact cache (Table 3); Li et al. (2026) distill a full-context teacher on the document's text into buffer tokens written by a LoRA compiler; NCC refines AM's native rows, the best start we tested.

7 Conclusion

Because an objective bounds the loss only where its supervision reaches, NCC refines an AM cache on the document's own continuations, a signal that covers every source token and needs no generated text. Never significantly worse overall than self-study at 8–17% of its refinement time, it leads where self-study's supervision does not reach and improves on AM across budgets, documents of up to 97K tokens, four LongBench QA tasks, an FP4 reader with a 4-bit state and, in constrained recall, three of four further hybrids. The compiled state is native: the unchanged model reads it at every query, answers a 16-paper document after 0.12 s instead of 11.1 s, and repays the compilation within 224 queries. Compiling a document once and serving it many times is thus a practical alternative to reading it again for every question.

AI USE STATEMENT

The research idea is the authors' own, and the authors directed the experiments. We used generative AI tools to polish the writing, to assist with part of the literature search, and, under the authors' direction, as coding assistants for implementing, running and analyzing experiments and for the Lean formalization of our proofs. We have reviewed all AI-assisted work, verified the reported results against the original model outputs, and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix O specifies the model, native state, Attention Matching setup, refinement objective, self-study control, decoding and statistics; Appendix P gives the cost and memory accounting; Appendix Q describes task construction; Appendix H defines the training-free baselines and their adaptive-budget variants; Appendix T contains the proofs and numerical checks; Appendices A–G describe the LongBench, cross-model, generation-free, low-precision, serving and sensitivity studies; and Appendix V lists every per-paper score. All scores use the first read of each question (Appendix O.5), averaged within and then across papers.

ETHICS STATEMENT

This study uses public research papers from Qasper, public articles from QuALITY, the public LongBench test sets and model outputs; it collects no human-subject data. Agreement with a full-context model does not guarantee factual truth, and compiled states inherit any errors of the model that reads them.

REFERENCES

Kirill Afendulev, Alexey Dontsov, Elena Tutubalina, and Anton Korznikov. What Attention Recalls and Recurrence Controls in Hybrid Language Models. arXiv preprint arXiv:2609.04434, 2026. To appear in Findings of EMNLP 2026.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *The Twelfth International Conference on Learning Representations*, 2024.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Josh Alman and Zhao Song. Fast Attention Requires Bounded Entries. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Joshua D. Angrist and Jörn-Steffen Pischke. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press, 2009.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and Improving Recall in Efficient Language Models. In *The Twelfth International Conference on Learning Representations*, 2024a.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 2024b.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, et al. A General Language Assistant as a Laboratory for Alignment. arXiv preprint arXiv:2112.00861, 2021.

Changwoo Baek, Seungjun Shin, and Kyeongbo Kong. RestoreKV: Recovering Full-Cache Behavior Under Aggressive Query-Agnostic KV Cache Eviction. arXiv preprint arXiv:2608.01247, 2026.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, et al. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A Theory of Learning from Different Domains. *Machine Learning*, 79(1–2), 2010.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Lucas Caccia, Alan Ansell, Edoardo Ponti, Ivan Vulić, and Alessandro Sordoni. Training Plug-and-Play Knowledge Modules with Deep Context Distillation. In *Second Conference on Language Modeling*, 2025.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, et al. PyramidKV: Dynamic KV Cache Compression based on Pyramidal Information Funneling. In *Second Conference on Language Modeling*, 2025.

Brian J. Chan, Chao-Ting Chen, Jui-Hung Cheng, and Hen-Hsen Huang. Don't Do RAG: When Cache-Augmented Generation is All You Need for Knowledge Tasks. In *Companion Proceedings of the ACM on Web Conference 2025*, 2025.

Rujikorn Charakorn, Edoardo Cetin, Shinnosuke Uesaka, and Robert Tjarko Lange. Doc-to-LoRA: Learning to Instantly Internalize Contexts. In *Forty-third International Conference on Machine Learning*, 2026.

Vivek Chari, Guanghui Qin, and Benjamin Van Durme. KV-Distill: Nearly Lossless Learnable Context Compression for LLMs. arXiv preprint arXiv:2503.10337, 2025.

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, et al. MagicPIG: LSH Sampling for Efficient LLM Generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting Language Models to Compress Contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning Bounds for Importance Weighting. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, second edition, 2006.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 2024.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A Dataset of Information-Seeking Questions and Answers Anchored in Research Papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

Soham De, Samuel L. Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, et al. Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models. arXiv preprint arXiv:2402.19427, 2024.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 Theorem Prover and Programming Language. In *Automated Deduction – CADE 28*, volume 12699, 2021.

Alessio Devoto, Maximilian Jeblick, and Simon Jégou. Expected Attention: KV Cache Compression by Estimating Attention from Future Queries Distribution. arXiv preprint arXiv:2510.00636, 2025.

Yixin Dong, Charlie F. Ruan, Yaxing Cai, Ziyi Xu, Yilong Zhao, Ruihang Lai, and Tianqi Chen. XGrammar: Flexible and Efficient Structured Generation Engine for Large Language Models. In *Proceedings of Machine Learning and Systems*, volume 7, 2025.

M. D. Donsker and S. R. S. Varadhan. Asymptotic Evaluation of Certain Markov Process Expectations for Large Time, I. *Communications on Pure and Applied Mathematics*, 28(1), 1975.

B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1), 1979.

Sabri Eyuboglu, Ryan Saul Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Ruoyu Liu, Atri Rudra, James Zou, Azalia Mirhoseini, and Christopher Ré. Cartridges: Lightweight and general-purpose long context representations via self-study. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou. Ada-KV: Optimizing KV Cache Eviction by Adaptive Budget Allocation for Efficient LLM Inference. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model Tells You What to Discard: Adaptive KV Cache Compression for LLMs. In *The Twelfth International Conference on Learning Representations*, 2024a.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context Autoencoder for Context Compression in a Large Language Model. In *The Twelfth International Conference on Learning Representations*, 2024b.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt Cache: Modular Attention Reuse for Low-Latency Inference. In *Proceedings of Machine Learning and Systems*, volume 6, 2024.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *First Conference on Language Modeling*, 2024.

Momchil Hardalov, Gonzalo Iglesias, and Adrià de Gispert. Cartridges at Scale: Training Modular KV Caches over Large Document Collections. arXiv preprint arXiv:2606.04557, 2026.

Themistoklis Haris and Krzysztof Onak. Compression Barriers in Autoregressive Transformers. In *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291, 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. arXiv preprint arXiv:1503.02531, 2015.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*, 2020.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat After Me: Transformers are Better than State Space Models at Copying. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 2024.

Bo Jiang and Sian Jin. KVSculpt: KV Cache Compression as Distillation. arXiv preprint arXiv:2603.27819, 2026.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Divyansh Kaushik and Zachary C. Lipton. How Much Reading Does Reading Comprehension Require? A Critical Investigation of Popular Benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W. Lee, Sangdoo Yun, and Hyun Oh Song. KVzip: Query-Agnostic KV Cache Compression with Context Reconstruction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Yoon Kim and Alexander M. Rush. Sequence-Level Knowledge Distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA Reading Comprehension Challenge. *Transactions of the Association for Computational Linguistics*, 6, 2018.

Kalle Kujanpää, Pekka Marttinen, Harri Valpola, and Alexander Ilin. Efficient Knowledge Injection in LLMs via Self-Distillation. *Transactions on Machine Learning Research*, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified Adversarial Robustness with Additive Noise. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM Knows What You are Looking for Before Generation. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Zeju Li, Yizhou Zhou, and Qiang Xu. Latent Context Compilation: Distilling Long Context into Compact Portable Memory. In *Third Conference on Language Modeling*, 2026.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, et al. Jamba: A Hybrid Transformer-Mamba Language Model. arXiv preprint arXiv:2403.19887, 2024.

Marc Lipsitch, Eric Tchetgen Tchetgen, and Ted Cohen. Negative Controls: A Tool for Detecting Confounding and Bias in Observational Studies. *Epidemiology*, 21(3), 2010.

Yuhan Liu, Hanchen Li, Yihua Cheng, Siddhant Ray, Yuyang Huang, Qizheng Zhang, Kuntai Du, Jiayi Yao, et al. CacheGen: KV Cache Compression and Streaming for Fast Large Language Model Serving. In *Proceedings of the ACM SIGCOMM 2024 Conference*, 2024a.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the Persistence of Importance Hypothesis for LLM KV Cache Compression at Test Time. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 2024b.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019.

Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, Seungyeon Kim, and Sanjiv Kumar. A Statistical Perspective on Distillation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 2021.

William Merrill, Yanhong Li, Tyler Romero, Anej Svete, Caia Costello, Pradeep Dasigi, Dirk Groeneveld, David Heineman, et al. Olmo Hybrid: From Theory to Practice and Back. arXiv preprint arXiv:2604.03444, 2026.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to Compress Prompts with Gist Tokens. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

NVIDIA. NVIDIA Nemotron Nano 2: An Accurate and Efficient Hybrid Mamba-Transformer Reasoning Model. arXiv preprint arXiv:2508.14444, 2025.

NVIDIA, Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, et al. Nemotron 3 Nano: Open, Efficient Mixture-of-Experts Hybrid Mamba-Transformer Model for Agentic Reasoning. arXiv preprint arXiv:2512.20848, 2025a. 4B model card: https://huggingface.co/nvidia/NVIDIA-Nemotron-3-Nano-4B-BF16.

NVIDIA et al. Nemotron-H: A Family of Accurate and Efficient Hybrid Mamba-Transformer Models. arXiv preprint arXiv:2504.03624, 2025b.

Charles O'Neill, Alex Sandomirsky, Harry Partridge, Mudith Jayasekara, and Max Kirkby. Still: Amortized KV Cache Compaction in a Single Forward Pass. arXiv preprint arXiv:2606.07878, 2026.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, et al. QuALITY: Question Answering with Long Input Texts, Yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.

Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, 1993.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

George Pólya and Gabor Szegő. *Problems and Theorems in Analysis II: Theory of Functions. Zeros. Polynomials. Determinants. Number Theory. Geometry*. Springer Berlin Heidelberg, 1998.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, et al. Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Qwen Team. Qwen3.5-9B, 2026a. URL https://huggingface.co/Qwen/Qwen3.5-9B. Model card.

Qwen Team. Qwen3.6-27B: Flagship-Level Coding in a 27B Dense Model, 2026b. URL https://qwen.ai/blog?id=qwen3.6-27b. Model card: https://huggingface.co/Qwen/Qwen3.6-27B.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple Hybrid State Space Models for Efficient Unlimited Context Language Modeling. In *The Thirteenth International Conference on Learning Representations*, 2025.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, 2011.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear Transformers Are Secretly Fast Weight Programmers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 2021.

Hidetoshi Shimodaira. Improving Predictive Inference under Covariate Shift by Weighting the Log-Likelihood Function. *Journal of Statistical Planning and Inference*, 90(2), 2000.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by Distilling Context. arXiv preprint arXiv:2209.15189, 2022.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing*, 568, 2024.

Sijun Tan, Xiuyu Li, Shishir G Patil, Ziyang Wu, Tianjun Zhang, Kurt Keutzer, Joseph E. Gonzalez, and Raluca Ada Popa. LLoCO: Learning Long Contexts Offline. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. QUEST: Query-Aware Sparsity for Efficient Long-Context LLM Inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023. GitHub repository.

The mathlib Community. The Lean Mathematical Library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, 2020.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer Dissection: An Unified Understanding for Transformer's Attention via the Lens of Kernel. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau,

Evgeni Burovski, Pearu Peterson, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 2020.

Zixuan Wang, Xingyu Dang, Rui-Jie Zhu, Zixin Wen, Hengyu Fu, Wenhao Chai, and Jason D. Lee. Learning What to Remember: Test-Time Training via Context Distillation. arXiv preprint arXiv:2608.01672, 2026.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *The Twelfth International Conference on Learning Representations*, 2024.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient Long-Context LLM Inference with Retrieval and Streaming Heads. In *The Thirteenth International Conference on Learning Representations*, 2025.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving Retrieval-Augmented LMs with Context Compression and Selective Augmentation. In *The Twelfth International Conference on Learning Representations*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, et al. Qwen3 Technical Report. arXiv preprint arXiv:2505.09388, 2025a.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated Linear Attention Transformers with Hardware-Efficient Training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 2024a.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing Linear Transformers with the Delta Rule over Sequence Length. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated Delta Networks: Improving Mamba2 with Delta Rule. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the Softmax Bottleneck: A High-Rank RNN Language Model. In *International Conference on Learning Representations*, 2018.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. Synthetic continued pretraining. In *The Thirteenth International Conference on Learning Representations*, 2025c.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. CacheBlend: Fast Large Language Model Serving for RAG with Cached Knowledge Fusion. In *Proceedings of the Twentieth European Conference on Computer Systems*, 2025.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long Context Compression with Activation Beacon. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, et al. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, et al. SGLang: Efficient Execution of Structured Language Model Programs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Adam Zweiger, Xinghong Fu, Han Guo, and Yoon Kim. Fast KV Compaction via Attention Matching. In *Forty-third International Conference on Machine Learning*, 2026.

APPENDIX OVERVIEW

Table 4 lists the questions a reader may ask about the main results, the answer from the additional studies, and where each study is reported. Every study uses the 64 held-out papers unless stated otherwise; Figure 8 shows the gain of NCC over its AM initialization in each of them.

Compute. The studies ran on Grace Hopper (GH200, 96 GB) and Grace Blackwell (GB200 NVL4 and NVL72, GB300 NVL72) superchips, with H100 GPUs for the Nemotron and Qwen3-8B studies. At peak, 100 GPUs ran concurrently (68 GH200 and 32 GB300), and up to 48 GB200 and 32 GB300 GPUs worked on the paper at the same time. Each study reports its hardware and per-stage times.

Question	Answer	Where
Does it hold on a standard long-context benchmark?	On four LongBench QA tasks, NCC is statistically indistinguishable from the full-context reader on three, with a 5–11× smaller state, and improves on AM by 4.7–33.1 F1 on all four; on NarrativeQA it ties self-study at a tenth of its refinement time.	Appendix A
Does it hold on other models?	On three further hybrids (Qwen3.5-9B; Mamba-2 hybrids Nemotron 3 Nano 4B and Nano 9B v2), NCC improves on AM by 7.8–10.2 points of constrained recall; the gain tracks how well the full-context teacher knows the numbers, and it exceeds KVzip on every model tested.	Appendices B and C
Does the pipeline need generated text at all?	No: with AM fitted on a repeat-prefill of the document, no stage generates a token and NCC scores the same (78.6% constrained recall and 45.5 F1 against 78.6% and 46.2).	Appendix D
How stable is a single run?	Five runs from the same initialization: 79.4 ± 0.5 constrained recall and 45.1 ± 1.3 F1; the gain over AM holds in every run.	Appendices D and G
Does it survive low-precision serving?	With NVIDIA's NVFP4 reader, and with the rows and recurrent state stored in NVFP4 (30.4 MiB, 13.7× smaller than the BF16 full state; 180× on 16-paper documents), no single-paper score changes significantly.	Appendices E and E.1
What does serving cost?	No prefill: 0.12 s to the first token on 16-paper documents instead of 11.1 s, about 360 compiled states per GPU against 7 full ones, and 128 concurrent requests where full caches run out of memory at 8.	Appendix F
Were the settings tuned?	Each of the eight tested settings keeps 69–106% of the gain over AM in constrained recall, every interval above zero; halving or doubling the spans, or doubling the cue, moves constrained recall by at most 1.3 points.	Appendix G
Are the training-free baselines too weak?	With their own adaptive head and layer budgets, KVzip and SnapKV stay 23–47 points of constrained recall below NCC (which is higher on at least 56 of the 64 papers against each) and below AM in constrained recall and QA F1.	Appendix H.1
Is the self-study comparison fair?	At an equal 1,024 updates, NCC leads self-study by 5.2 points [2.0, 8.6] of constrained recall; its exposure-matched checkpoint and long-document runs are in the appendix.	Appendices J and L.1
Where are the proofs?	Proofs, numerical checks and the Lean 4 formalization; further discussion of each result.	Appendices T and U

Table 4: **Guide to the appendix.**

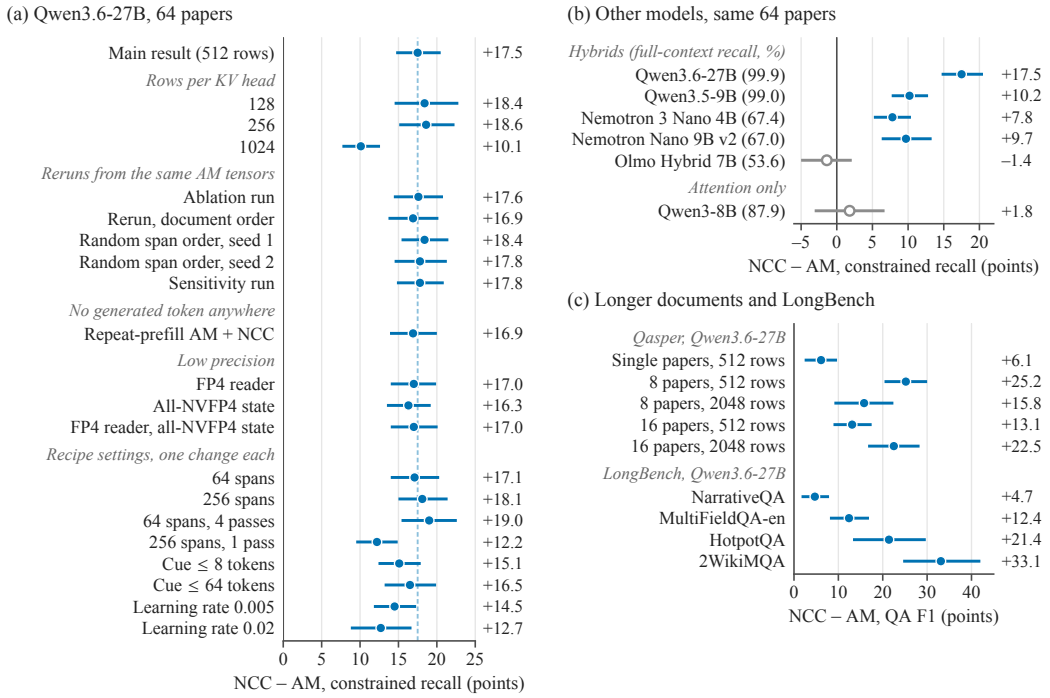


Figure 8: **The gain of NCC over its AM initialization in every study.** Paired differences with 95% bootstrap intervals, each against the AM state read in the same run; hollow gray: interval includes zero. (a) Qwen3.6-27B, 64 papers, constrained recall: row budgets (Appendix N), five runs of the default recipe from the same AM tensors (Appendices D and G), the generation-free pipeline (Appendix D), FP4 reading and NVFP4 storage (Appendices E and E.1) and one-at-a-time changes of the recipe (Appendix G); dashed: main result. (b) Other models, same papers and protocol (Table 7); in parentheses: the full-context reader's constrained recall. (c) QA F1 on single papers, 8- and 16-paper documents (Appendix I; intervals over documents for 8-paper documents, over papers otherwise) and four LongBench tasks (official F1; intervals over stories or documents; Appendices A and A.1).

A STANDARD LONG-CONTEXT BENCHMARKS: LONGBENCH

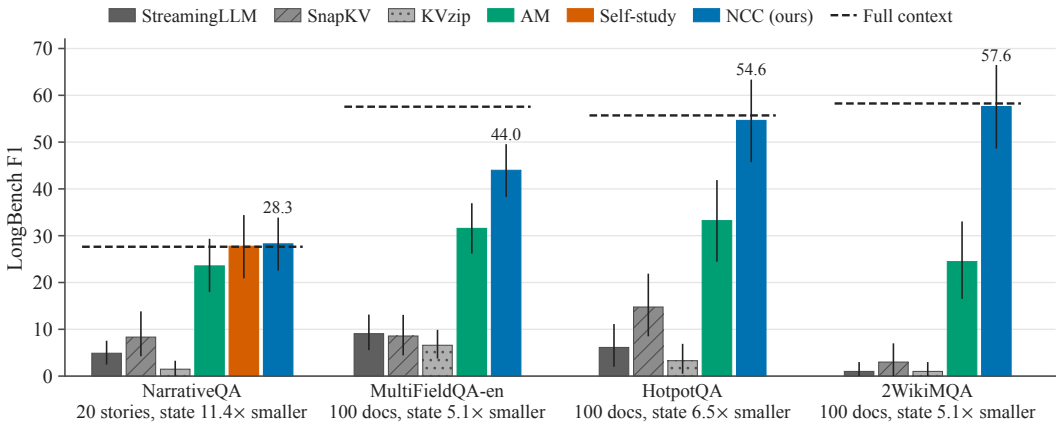


Figure 9: **On three of four LongBench QA tasks NCC is statistically indistinguishable from the full-context reader, while training-free selections fall to 1–15 F1.** Official LongBench F1 of Qwen3.6-27B reading each compact state (bars; whiskers: 95% bootstrap intervals over stories or documents) and the full context (dashed). Every compact state is compiled once without the question, at 512 rows per KV head (2048 for documents of at least 16K tokens); under each task: number of stories or documents and how much smaller NCC's state is than the full-context state. The recurrent-state-only control scores 0.0 on every task and is not drawn; self-study was run on NarrativeQA only. Paired differences are in Tables 5 and 6.

We first evaluate on the NarrativeQA subset (Kočiský et al., 2018) of LongBench (Bai et al., 2024): all 200 official questions, which ask about 20 stories (novels and movie scripts) of 8,694–67,682 tokens, so one compiled state per story serves all of its 5–19 questions. Each story is compiled once, without seeing any question, with the unchanged recipe of the main study (AM initialization, then 256 source-continuation updates on 128 spans); stories of at least 16K tokens use 2,048 rows per KV head and the others 512. KVzip, SnapKV and StreamingLLM keep the same number of rows, chosen without the question, and the recurrent-state-only control keeps the Gated DeltaNet state and the template rows. Questions use the LongBench NarrativeQA instruction inside our chat template and are answered by greedy decoding with LongBench's limit of 128 new tokens (first read only, no thinking); the score is the official LongBench F1. The benchmark subset, arms and analysis were fixed before any evaluation read. Intervals are paired bootstrap intervals over the 20 stories (20,000 resamples); question-level bootstrap intervals, which ignore that questions share a story, lead to the same conclusion for every contrast reported here (e.g., NCC−AM +4.7 [1.0, 8.5], NCC−full context +0.6 [−2.8, 3.9], NCC−self-study +0.5 [−3.6, 4.6]).

	Budget by length (512 / 2048 rows)				512 rows for every story		
State	MiB	F1	− NCC		F1	− NCC	
Full context	1996	27.6	−0.6	[−4.7, 3.3]	27.6	+5.5	[0.1, 11.7]
Recurrent state only	80	0.0	−28.3	[−33.7, −22.7]	0.0	−22.1	[−27.5, −17.0]
StreamingLLM	175	4.9	−23.4	[−28.6, −18.3]	3.8	−18.3	[−22.9, −14.1]
SnapKV	175	8.3	−19.9	[−25.7, −14.1]	2.9	−19.2	[−24.7, −14.0]
KVzip	175	1.5	−26.8	[−32.7, −20.6]	0.5	−21.7	[−27.3, −16.1]
Attention Matching	175	23.6	−4.7	[−7.9, −1.7]	14.6	−7.6	[−10.1, −5.1]
Self-study (256 questions)	175	27.8	−0.5	[−5.3, 3.7]	–		
NCC	175	**28.3**			**22.1**		

Table 5: **LongBench NarrativeQA** (200 questions on 20 stories of 8.7K–67.7K tokens; Qwen3.6-27B; official LongBench F1). MiB: mean stored state per story under the length-dependent budget, including the 75.75 MiB recurrent and convolutional state; the full-context state averages 1996 MiB, 11.4× the compiled state. −NCC: the row's state minus NCC, with 95% story-level bootstrap intervals.

Self-study, run with its unchanged recipe from the same AM state (256 generated questions from eight 2,048-token windows, 256 updates), reaches 27.8 F1, statistically indistinguishable from NCC (-0.5 [-5.3, 3.7]), while generating 14.1K tokens per story and spending 9.5× NCC's refinement-stage GPU time (22.9 against 2.4 GPU-minutes per story on GH200, of which self-study's 256 updates take 3.0; the shared AM initialization is excluded for both). With the length-dependent budget, NCC scores 28.3 F1 against 27.6 for the full-context reader ($+0.6$ [-3.3, 4.7]; higher on 11 stories, lower on 7) while storing 11.4× less state, and it improves on AM by 4.7 F1 [1.7, 7.9], higher on 16 stories and lower on 2. The training-free selections fall to 1.5–8.3 F1 and answer `Unanswerable` to 154–191 of the 200 questions, close to the recurrent-state-only control, which answers `Unanswerable` to every question. As in all our experiments, the reader's instructions allow `Unanswerable`; the full-context reader gives it to 39 of the 200 questions, NCC to 40 (28 of them the same questions), self-study to 35 and AM to 91; of the 161 questions the full-context reader answers, NCC abstains on 12 and AM on 53. With 512 rows for every story (60× fewer K/V rows), NCC keeps 22.1 F1, 7.6 points [5.1, 10.1] above AM and higher on all 18 stories where the two differ. On the 14 stories of at least 16K tokens at 2048 rows, NCC reaches 29.0 F1 against 27.9 for full context and 24.0 for AM ($+4.9$ [0.7, 9.6]).

A.1 Three more LongBench tasks

We repeat the comparison on the other English QA tasks of LongBench scored by F1: MultiFieldQA-en (single-document QA) and HotpotQA and 2WikiMultihopQA (multi-hop QA over concatenated passages). For each task we use a random subset of 100 documents fixed before any read (MultiFieldQA-en: 100 of its 112 documents, with 135 of its 150 questions; HotpotQA and 2WikiMQA: 100 of 200 documents, one question each; prefixes of 1.3K–17.8K tokens), the official instructions inside our chat template, greedy decoding with LongBench's limits of 64 (MultiFieldQA-en) and 32 new tokens, and the official metric. Each document is compiled once without its question, at 512 rows per KV head, or 2048 for the 44 documents of at least 16K tokens; the compiled states are on average 5.6× smaller than the full-context states. Intervals are paired bootstrap intervals over documents, stratified by task (10,000 resamples). Documents ran on GB200 (two processes per GPU) or GH200 GPUs, with all states of a document in one process, so every paired difference is within one device; per document, NCC's teacher targets and 256 updates took 71 s and 276 s on average at 512 rows, against 781 s to generate AM's fit sequences.

State	MultiFieldQA-en	HotpotQA	2WikiMQA	Average	Average − NCC
Full context	57.6	55.7	58.3	57.2	$+5.1$ [1.4, 8.8]
Recurrent state only	0.0	0.0	0.0	0.0	-52.1 [-56.7, -47.4]
StreamingLLM	9.1	6.1	1.0	5.4	-46.7 [-51.4, -41.9]
SnapKV	8.6	14.8	3.0	8.8	-43.3 [-48.2, -38.5]
KVzip	6.6	3.3	1.0	3.6	-48.4 [-53.3, -43.6]
Attention Matching	31.6	33.2	24.5	29.8	-22.3 [-26.6, -18.1]
NCC	**44.0**	**54.6**	**57.6**	**52.1**	
NCC − full context	-13.6 [-18.9, -8.7]	-1.1 [-7.0, 4.7]	-0.7 [-9.1, 7.7]		
NCC − AM	$+12.4$ [8.1, 16.9]	$+21.4$ [13.3, 29.7]	$+33.1$ [24.6, 42.0]		

Table 6: **Three more LongBench tasks** (100 documents each, 335 questions; Qwen3.6-27B; official LongBench F1; Average: mean of the three task scores, as in LongBench). Intervals: 95% paired bootstrap over documents.

On the two multi-hop tasks NCC is statistically indistinguishable from the full-context reader (54.6 against 55.7 and 57.6 against 58.3 F1, with the same F1 on 79 and 70 of the 100 documents), and on all three tasks it improves on AM, by 12.4–33.1 F1, scoring higher on 125 of the 300 documents and lower on 16 (MultiFieldQA-en 55/13, HotpotQA 31/2, 2WikiMQA 39/1); the training-free selections fall to 1.0–14.8 F1 and answer `Unanswerable` to 278–302 of the 335 questions, against 44 for the full-context reader, 159 for AM and 39 for NCC. On MultiFieldQA-en, where 99 of the 100 documents are shorter than 16K tokens and so get 512 rows, NCC trails the full-context reader by 13.6 F1. With 512 rows for every document (15–27× fewer K/V rows than the full cache), NCC still averages 50.4 F1 against 25.4 for AM ($+25.0$ [20.7, 29.4]), remains statistically indistinguishable from the full-context reader on HotpotQA (51.5; -4.2 [-11.3, 3.1]) and 2WikiMQA (56.4; -1.8 [-9.9, 6.4]), and scores higher than AM on every HotpotQA and 2WikiMQA document where the

two differ (35 and 40 documents). Across the four LongBench tasks at the length-dependent budget, NCC is statistically indistinguishable from the full-context reader on three and significantly above AM and every training-free selection on all four.

B Other hybrid models

All main results use one hybrid model, Qwen3.6-27B. This section repeats the 64-paper comparison on four further hybrids, of another scale, another family and another recurrent mechanism, with the unchanged protocol; Table 7 summarizes them together with the pure-attention model of Appendix C.

		Constrained recall			NCC − AM					
Model	Recurrent	Full	AM	NCC	Constr.		Greedy		QA F1	
Qwen3.6-27B	GDN	99.9	62.3	79.8	+17.5	[14.7, 20.5]	+59.7	[55.3, 64.1]	+6.1	[2.4, 9.7]
Qwen3.5-9B	GDN	99.0	59.3	69.6	+10.2	[7.7, 12.8]	+21.9	[18.7, 25.4]	+2.8	[−1.6, 7.3]
Nemotron Nano 9B v2	Mamba-2	67.0	29.6	39.3	+9.7	[6.3, 13.3]	+17.0	[13.6, 20.5]	−0.8	[−4.6, 2.5]
Nemotron 3 Nano 4B	Mamba-2	67.4	25.8	33.6	+7.8	[5.2, 10.4]	+9.0	[6.3, 11.9]	−0.4	[−4.4, 3.6]
Olmo Hybrid 7B	GDN	53.6	25.4	24.0	−1.4	[−5.0, 2.1]	+8.0	[4.1, 11.9]	−6.8	[−11.5, −2.3]
Qwen3-8B	none	87.9	43.0	44.8	+1.8	[−3.1, 6.7]	+5.2	[0.2, 10.1]	−5.5	[−10.9, −0.1]

Table 7: **NCC against AM on six models** (64 papers, 512 rows per KV head, paper-averaged first reads; 95% paper-level bootstrap intervals). Full: the full-context reader, which is also NCC's teacher; GDN: Gated DeltaNet. Learning rate: 0.01 on the two Qwen hybrids and the scale-matched rule of Appendix C on the other models (for the Nemotron and Olmo models fixed in their plans before any evaluation read). Among the five hybrids the gain follows the full-context reader's recall (post hoc and descriptive). On every model NCC exceeds KVzip in constrained recall, by 8.5–29.0 points, with every interval above zero.

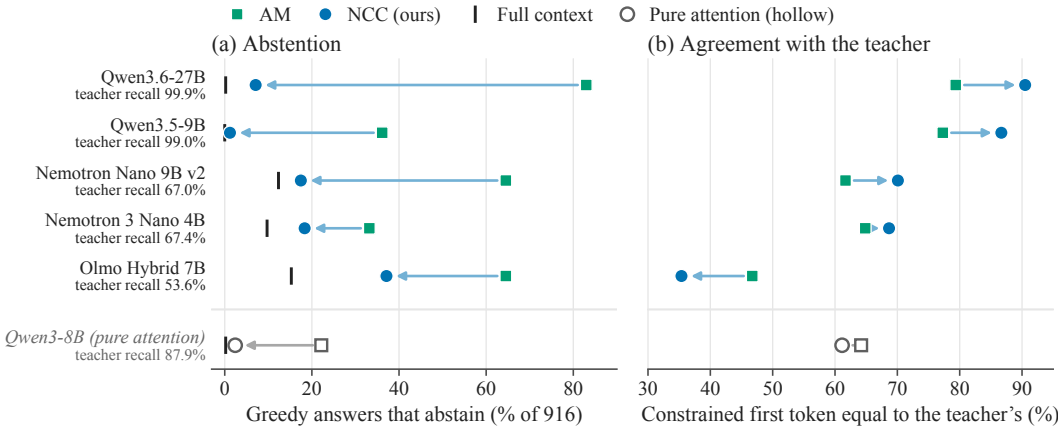


Figure 10: **What refinement changes on six models** (916 number-recall questions, 64 papers, 512 rows per KV head, first reads). (a) Share of greedy answers that abstain for AM (square), NCC (circle) and the full-context reader (tick): NCC lowers it on every model, by 14.8–75.9 points, with every 95% paper-cluster interval below zero. (b) Share of questions whose first number token under constrained decoding equals the full-context teacher's: it rises on the four hybrids where NCC improves recall, falls on Olmo Hybrid 7B and does not rise on Qwen3-8B (logged aggregates; no intervals). Teacher recall: full-context constrained recall. Hollow gray: the pure-attention Qwen3-8B.

B.1 Qwen3.5-9B

This subsection repeats the comparison on Qwen3.5-9B (Qwen Team, 2026a), a hybrid of a different scale from the same family. It has 32 layers: 24 Gated DeltaNet layers, each with a recurrent state, and 8 attention layers, each with a KV cache. The protocol, the model choice and the learning rate were fixed before any evaluation paper was read.

		State per paper (MiB)			Reduction		Time per paper (s)	
Model	Layers×KV	Full	Compact	Recurrent	KV rows	Total	AM	Refine
Qwen3.6-27B	16×4	415.6	107.8	75.8	10.6×	3.9×	Figure 18	127
Qwen3.5-9B	8×4	195.4	41.5	25.5	10.6×	4.7×	617	75
Nemotron Nano 9B v2	4×8	223.1	145.5	137.5	10.7×	1.5×	380	85
Nemotron 3 Nano 4B	4×8	165.9	88.3	80.3	10.7×	1.9×	511	99
Olmo Hybrid 7B	8×30	679.1	113.0	52.7	10.4×	6.0×	1,954	81
Qwen3-8B	36×8	764.3	72.3	–	10.6×	10.6×	2,281	145

Table 8: **State size and compile cost on six models** (512 rows per KV head; means over the 64 papers). Layers×KV: attention layers times KV heads per layer. Recurrent: the fixed recurrent and convolution state that every compact state keeps. Reduction: full over compact, for KV rows and for the whole state. AM: fit-bank generation plus OMP; Refine: teacher targets plus 256 updates (Qwen3.5-9B with compiled kernels; 160 s on the first call). Hardware: GH200 for the Qwen hybrids and Olmo, H100 for Nemotron and for 62 of the 64 Qwen3-8B papers, with 1–3 processes per GPU, so times are comparable within a model, not across models. Refinement takes 4–22% of AM's time on the five models with a listed AM time.

Setup. We use the instruction-tuned Qwen3.5-9B in BF16, with no thinking and greedy decoding. Its 8 attention layers have 16 query and 4 KV heads of dimension 256. Its tokenizer and chat format match Qwen3.6-27B's, so every prompt, source span and template/document partition is identical to the main comparison.

The model was chosen by a preregistered rule on the 12 development papers: the full-context reader must reach at least 75% constrained number recall. Qwen3.5-9B reaches 99.4% and Qwen3.5-4B 99.1%; the larger model was preferred.

The papers, the 916 number-recall and 234 Qasper questions, the prompts, decoders, scorers and first-read protocol are those of the main comparison. Every compressed state keeps 512 rows per KV head in all 8 attention layers: 61 template rows and 451 document rows. It also keeps the full-context recurrent and convolution state, as on Qwen3.6-27B. On the 2.4K–9.2K-token papers this stores 10.6× fewer KV rows than the full cache on average (4.7–17.9×). That is 16 MiB of KV rows against a mean of 170 MiB. The fixed recurrent state adds 25.5 MiB to every state, so the whole state shrinks 4.7× on average (41.5 against 195 MiB; 2.4–7.5×).

AM uses the same recipe, with a fit bank generated by Qwen3.5-9B itself. NCC uses the unchanged recipe of the main comparison: AM initialization, 128 source spans with cues of at most 32 tokens, the full-context teacher, forward KL, and 256 AdamW updates at learning rate 0.01, with BF16 export.

Learning rate. We kept the learning rate of 0.01 because the two models come from the same family, and their cache scales are indeed close. The per-layer RMS of the retained keys is 1.35–1.56 on Qwen3.5-9B and 1.24–1.57 on Qwen3.6-27B. With learning rate 0.01, the relative updates $\|\Delta K\|/\|K\|$ are 0.10–0.13 and $\|\Delta V\|/\|V\|$ are 0.04–0.25 per layer (medians over papers). On Qwen3.6-27B they are 0.11–0.15 and 0.04–0.30.

Two further variants were fixed in advance and are reported next to, not instead of, the primary result. The first uses the scale-matched learning rate of Appendix C. The second runs 1,024 updates instead of 256.

Hardware and cost. All states of a paper ran in one process on one GH200, with two processes sharing a GPU. Per paper, AM takes 617 s: 464 s to generate the fit bank and 153 s of orthogonal matching pursuit (OMP) key selection and fitting over 32 KV heads. The 256 refinement updates take 64 s once kernels are compiled (149 s on the first call, which includes compilation), and 1,024 updates take 216 s. The whole study used 22.6 allocated GPU hours.

NCC improves on AM on the second hybrid model. In constrained recall the gain is $+10.2$ points [7.7, 12.8], and NCC is better on 45 of the 64 papers and worse on 4. In greedy recall the gain is $+21.9$. Both AM and NCC are far above KVzip at the same budget.

State	Qwen3.5-9B (hybrid) Greedy	Constrained	QA F1	Qwen3.6-27B (hybrid) Greedy	Constrained	QA F1
Full context	98.9	99.0	46.8	99.7	99.9	45.9
Attention Matching	47.4	59.3	36.9	16.7	62.3	38.1
NCC	**69.4**	**69.6**	**39.7**	**76.4**	**79.8**	**44.2**
KVzip	13.9	40.6	15.7	9.4	52.2	18.2
Floor (template rows only)	2.0	20.9	8.9	0.9	22.3	10.0
NCC, scale-matched learning rate	71.8	72.2	37.9			
NCC, 1,024 updates	76.5	76.7	40.2			

Table 9: **Two hybrid models on the 64 held-out papers** (512 rows per KV head; paper-averaged first reads). Greedy and constrained: number recall (%). Every compressed state, including the floor, keeps the full-context recurrent state. Bold: best of the five primary states. The last two rows are variants fixed in advance; they were run on Qwen3.5-9B only.

Difference	Greedy recall		Constrained recall		QA F1	
NCC minus AM	+21.9	[18.7, 25.4]	+10.2	[7.7, 12.8]	+2.8	[−1.6, 7.3]
NCC minus KVzip	+55.5	[51.1, 59.7]	+29.0	[24.3, 33.7]	+24.0	[17.5, 30.1]
AM minus KVzip	+33.6	[29.0, 38.2]	+18.8	[14.1, 23.5]	+21.2	[16.3, 26.0]
Full context minus NCC	+29.5	[25.3, 34.0]	+29.4	[25.2, 33.8]	+7.2	[3.5, 11.0]
NCC with 1,024 updates minus AM	+29.0	[25.6, 32.4]	+17.3	[14.5, 20.2]	+3.4	[−1.3, 7.8]
NCC with 1,024 minus 256 updates	+7.1	[4.5, 9.6]	+7.1	[4.6, 9.6]	+0.6	[−3.3, 4.4]
Scale-matched minus fixed learning rate	+2.4	[0.1, 4.7]	+2.6	[0.3, 5.0]	−1.7	[−4.5, 0.8]
Qwen3.6-27B: NCC minus AM	+59.7	[55.3, 64.1]	+17.5	[14.7, 20.5]	+6.1	[2.4, 9.7]

Table 10: **Paired differences on Qwen3.5-9B** (points; 95% paper-level bootstrap intervals, 20,000 resamples). The last row repeats the Qwen3.6-27B result for comparison.

The gain is smaller than on Qwen3.6-27B (+17.5 constrained), and full context stays 29 points ahead. Measured as a share of the gap between the floor and full context in constrained recall, AM recovers 49% and NCC 62%; the shares on Qwen3.6-27B are 52% and 74%. Running 1,024 instead of 256 updates adds a further 7.1 points and brings the share to 71% (+17.3 [14.5, 20.2] over AM, about as large as the 256-update gain on Qwen3.6-27B).

On QA the overall difference is not significant (+2.8 F1 [−1.6, 7.3]). On the 216 answerable questions NCC gains +5.4 F1 [1.9, 9.2] (question-weighted, paper-cluster bootstrap). On the 18 unanswerable questions it loses −22.2 [−43.8, −5.3]: there F1 rewards abstention, and NCC abstains less than AM (28% against 50%), as on Qwen3.6-27B and Qwen3-8B (Tables 16 and 48).

Model	State	Greedy answers Corr.	Wrong	Abst.	Constrained correct	P(Un)	Agreement (%) Greedy	Constr.
Qwen3.5-9B	Full context	905	11	0	906	≤0.002	100	100
	Attention Matching	414	171	331	527	≤0.34	57.1	77.3
	NCC	621	284	11	623	≤0.02	86.0	86.7
Qwen3.6-27B	Full context	913	1	2	915	0.003	100	100
	Attention Matching	141	15	760	557	0.80	16.9	79.4
	NCC	688	163	65	720	0.08	85.6	90.5

Table 11: **Outcomes and first-token statistics on the 916 number-recall questions** (first reads, question counts). P(Un) is the mean first-token probability of `Unanswerable`. For Qwen3.5-9B only the top-5 logits were logged, so its values are normalized over the top 5 and are upper bounds. Agreement: first token equal to the full-context teacher's, under greedy and under constrained decoding.

The mechanism matches Qwen3.6-27B rather than Qwen3-8B (Table 11). AM on Qwen3.5-9B abstains on 331 questions (P(Un) ≤ 0.34). NCC removes this abstention (11 questions left). It also ranks the correct number higher: with abstention forbidden, NCC answers 623 questions correctly

against 527 for AM, and first-token agreement with the full-context teacher under constrained decoding rises from 77% to 87%. On Qwen3-8B, by contrast, constrained recall barely moved (386 to 397) and agreement did not rise (Appendix C).

Scope. Qwen3.5-9B and Qwen3.6-27B share the attention and Gated DeltaNet design and the tokenizer, so this result shows that the gain is not specific to one checkpoint or scale; Appendix B.2 turns to another family and recurrent mechanism.

B.2 Mamba-2 hybrids from another family

Nemotron 3 Nano 4B (NVIDIA et al., 2025a) interleaves 21 Mamba-2 layers (Dao & Gu, 2024) with only 4 attention layers (40 query and 8 KV heads of dimension 128, without positional encoding) and 17 MLP layers. It differs from the Qwen models in family, tokenizer and recurrent mechanism. We use the post-trained checkpoint with reasoning off and greedy decoding, and run the protocol of the main comparison unchanged: the same 64 papers and questions, 512 rows per KV head, AM with a fit bank written by this model, KVzip at the same uniform budget, the template-rows-only control, and NCC from the AM initialization with 128 source spans and 256 updates. The model, the arms and the learning-rate rule were fixed before any evaluation paper was read. Because this family has no query–key normalization, the primary NCC arm uses the scale-matched learning rate of Appendix C; learning rate 0.01 runs as a declared secondary arm. On the development papers the full-context reader recalls 73.0% of the numbers, below the 75% threshold of the plan, so every compact state here is compared against a weaker ceiling than on the Qwen models; the larger Nemotron Nano 9B v2, the pre-registered fallback, reached only 70.1%.

State (64 papers, 512 rows)	Greedy	Constr.	QA F1
Full context	63.0	67.4	28.2
Recurrent state only	1.7	6.0	10.4
KVzip	4.2	9.2	14.0
Attention Matching	20.4	25.8	23.1
NCC (scale-matched learning rate)	**29.5**	**33.6**	22.7
NCC (learning rate 0.01, secondary)	28.0	31.6	**25.7**
Nemotron Nano 9B v2 (declared secondary model)			
Full context	61.8	67.0	42.0
Recurrent state only	0.9	7.1	9.0
KVzip	3.2	15.4	19.9
Attention Matching	18.8	29.6	36.5
NCC (scale-matched learning rate)	**35.8**	39.3	35.6
NCC (learning rate 0.01, secondary)	35.3	**40.1**	**39.0**
Paired differences, Nemotron 3 Nano 4B			
NCC − AM	+9.0 [6.3, 11.9]	+7.8 [5.2, 10.4]	−0.4 [−4.4, 3.6]
NCC − KVzip	+25.3 [21.4, 29.4]	+24.4 [20.6, 28.4]	+8.7 [3.2, 14.4]
NCC (lr 0.01) − AM	+7.6 [4.9, 10.4]	+5.8 [3.4, 8.3]	+2.6 [−0.8, 6.1]
Full context − NCC	+33.5 [28.9, 38.2]	+33.8 [29.1, 38.6]	+5.5 [1.8, 9.3]
Paired differences, Nemotron Nano 9B v2			
NCC − AM	+17.0 [13.6, 20.5]	+9.7 [6.3, 13.3]	−0.8 [−4.6, 2.5]
NCC (lr 0.01) − AM	+16.5 [13.3, 19.6]	+10.5 [7.3, 13.8]	+2.5 [0.7, 4.5]
NCC − KVzip	+32.6 [28.5, 36.8]	+23.9 [19.4, 28.6]	+15.8 [10.2, 21.0]

Table 12: **Two Mamba-2 hybrids**: Nemotron 3 Nano 4B (21 Mamba-2 and 4 attention layers) and, as a declared secondary model, Nemotron Nano 9B v2 (paper-averaged first reads). Greedy and constrained: number recall (%). Bottom: paired differences in points with 95% paper-level bootstrap intervals (20,000 resamples).

NCC improves on AM by 7.8 points of constrained recall [5.2, 10.4] and 9.0 greedy [6.3, 11.9], and on KVzip by 24.4, with no significant change in QA F1 (-0.4 [-4.4, 3.6]). Measured as a share of the gap between the recurrent-state-only control and full context, AM recovers 32% of constrained recall and NCC 45%. The refinement loss falls on all 64 papers (mean source-span KL 1.33 over the first 32 updates and 0.64 over the last 32). The fixed learning rate of the main study gives a

similar recall gain (+5.8 [3.4, 8.3]) and a nonsignificant QA gain (+2.6 [−0.8, 6.1]). With only four attention layers, the 512-row K/V state is 8.0 MiB against 85.6 MiB for the full cache (10.7× fewer rows on average), while the 80.3 MiB Mamba-2 state carried by every compact state dominates the total, which therefore shrinks only 1.9×. The declared secondary model, Nemotron Nano 9B v2 (NVIDIA, 2025), run with the same protocol on all 64 papers, shows the same pattern with larger gains: NCC improves on AM by 9.7 points of constrained recall [6.3, 13.3] and 17.0 greedy [13.6, 20.5], and with learning rate 0.01 also by 2.5 F1 [0.7, 4.5] (full context: 67.0% constrained recall; AM recovers 38% and NCC 54% of the gap from the recurrent-state-only control).

As on the Qwen hybrids, refinement both removes abstention and improves the ranking of numbers (Figure 10): greedy abstentions fall from 304 to 168 of the 916 questions on the 4B model and from 591 to 160 on the 9B model, and constrained first-token agreement with the full-context teacher rises from 64.8% to 68.7% and from 61.7% to 70.1%. NCC is better than AM on 39 and 45 of the 64 papers and worse on 6 and 9. On the numbers the full-context reader recalls under constrained decoding, the gain is +8.6 [5.5, 11.8] and +12.8 [8.6, 17.2] points (question-pooled, paper-cluster intervals). Under the scale-matched rule, keys and values move by about a tenth of their norm in every layer (0.10–0.11, medians over papers). The 9B model's attention layers have the same shape (8.0 against 85.6 MiB of K/V rows), and its 137.5 MiB Mamba-2 state limits the total reduction to 1.5×. On H100 GPUs shared by two or three processes, AM takes 511 s per paper on the 4B model (181 s fit bank, 330 s OMP) and 380 s on the 9B model; teacher targets take 8 s and the 256 updates 91 s and 77 s. The two studies used 4.9 and 6.0 allocated GPU hours.

B.3 A Gated DeltaNet hybrid from another family, and when the gain fades

Olmo Hybrid 7B (Merrill et al., 2026) interleaves 24 Gated DeltaNet layers with 8 attention layers that use multi-head attention (30 KV heads of dimension 128) without positional encoding. We use the instruction-tuned checkpoint (Olmo-Hybrid-Instruct-DPO-7B) and the protocol of the main comparison unchanged, with 58 template and 454 document rows per KV head; the checkpoint rule, the arms and the learning-rate rule (the scale-matched rule of Appendix C as primary, learning rate 0.01 as secondary) were fixed before any evaluation paper was read. On the development papers the full-context reader recalls only 58.1% of the numbers under constrained decoding (55.7% for the supervised fine-tuned checkpoint), below the 75% threshold of the plan, so the better checkpoint was used. The 512-row K/V state is 60.2 MiB against a mean of 626 MiB for the full cache (10.4× fewer rows; 4.6–17.7×), and every state carries the 52.7 MiB Gated DeltaNet state, so the whole state shrinks 6.0× on average (2.9–9.9×). Per paper, on GH200 GPUs shared by one or two processes, AM takes 1,954 s (360 s fit bank, 1,594 s OMP over 240 KV heads), teacher targets 9 s and the 256 updates 72 s. Some least-squares fits in AM's OMP code were rank-deficient, so we added a minimum-norm solver after the authors' fallbacks; 64 of the 15,360 fitted heads needed a fallback.

State (64 papers, 512 rows)	Greedy	Constr.	QA F1
Full context	48.7	53.6	33.5
Recurrent state only	0.0	3.0	8.7
KVzip	2.4	15.5	20.2
Attention Matching	15.3	**25.4**	**27.3**
NCC (scale-matched learning rate)	**23.3**	24.0	20.5
NCC (learning rate 0.01, secondary)	15.4	9.9	14.5
NCC − AM	+8.0 [4.1, 11.9]	−1.4 [−5.0, 2.1]	−6.8 [−11.5, −2.3]
NCC − KVzip	+20.9 [17.2, 25.0]	+8.5 [5.5, 11.5]	+0.3 [−6.0, 6.0]
Full context − NCC	+25.5 [21.1, 29.8]	+29.6 [25.3, 34.1]	+12.9 [7.8, 18.3]

Table 13: **Olmo Hybrid 7B** (24 Gated DeltaNet and 8 attention layers; paper-averaged first reads). Greedy and constrained: number recall (%). Bottom: paired differences in points with 95% paper-level bootstrap intervals (20,000 resamples).

Here NCC does not improve on AM in constrained recall (−1.4 [−5.0, 2.1]), so the pre-registered prediction is not supported, and it loses 6.8 QA F1 [−11.5, −2.3]. It still removes much of AM's abstention: greedy abstentions fall from 591 to 340 of the 916 questions and greedy recall rises by 8.0 points [4.1, 11.9]. The refinement itself behaves as on the other models (the mean source-span KL falls from 1.03 in the first pass over the 128 spans to 0.42 in the second, on all 64 papers).

A weak full-context reader limits what distillation can supply. NCC distills the full-context teacher, and all preservation results of Section 3 concern the teacher's decisions; they preserve a wrong answer as faithfully as a right one (Appendix T.6). Olmo's full-context reader answers 430 of the 916 number questions wrongly under constrained decoding, so on nearly half of the questions the teacher's top number is wrong (on these, AM and NCC recall only 4.2% and 8.4% of the numbers). Across the five hybrids we ran, the gain of NCC over AM follows how well the full-context reader knows the numbers (Table 7): it is largest where the reader recalls nearly all of them (Qwen), intermediate at two thirds (Nemotron) and absent at one half (Olmo). With five models this is a descriptive pattern, not a test, and the teacher is not the only factor: on the 486 questions that Olmo's full-context reader answers correctly under constrained decoding, NCC recalls 35.6% against 42.0% for AM (-6.4 $[-11.9, -1.0]$, question-pooled, paper-cluster intervals), so other differences between the models also contribute; candidates include attention without positional encoding, Olmo's 30 KV heads and the learning-rate rule.

C A PURE-ATTENTION MODEL

All main results use Qwen3.6-27B, a hybrid model in which 48 of 64 layers keep a recurrent state and only 16 layers keep a KV cache. This section repeats the 64-paper comparison on Qwen3-8B (Yang et al., 2025a), a standard Transformer (Vaswani et al., 2017) in which every layer is attention and the KV cache is the only document state. The analyses in the second half were chosen after the scores were known and are descriptive.

Setup. Qwen3-8B has 36 attention layers with 32 query and 8 KV heads of dimension 128 (BF16, no thinking, greedy decoding). The papers, the 916 number-recall and 234 Qasper questions, prompts, decoders, scorers and first-read protocol are those of the main comparison. AM uses the same recipe, with fit sequences generated by Qwen3-8B itself, and keeps 512 rows per KV head in every layer: 58 template rows and 454 document rows. On the 2.4K–9.1K-token papers this stores 10.6× fewer KV rows than the full cache on average (4.7–17.9×; 72 MiB against a mean of 764 MiB). NCC uses the same 128 source spans with 32-token cues, the same full-context teacher and forward KL, 256 AdamW updates and BF16 export.

Learning rate. The learning rate is matched to the scale of each layer's cache. Each layer l has one AdamW step size for keys, $c_K\, s_{l,K}$, and one for values, $c_V\, s_{l,V}$. Here $s_{l,K}$ is the RMS of the keys that AM keeps in layer l. $s_{l,V}$ is the RMS of the full-cache values at the positions AM keeps; AM's own fitted values occasionally have much larger norms. The constants $c_K = 0.0069$ and $c_V = 0.0066$ make the layer-averaged step equal to the learning rate of 0.01 used on Qwen3.6-27B, computed on the Qwen3.6-27B AM caches of the long-document study. The rule was fixed before any of its outcomes were observed.

The resulting updates are uniform across layers: $\|\Delta K\|/\|K\|$ is 0.06–0.10 and $\|\Delta V\|/\|V\|$ is 0.07–0.11 in every layer (medians over papers). On Qwen3.6-27B the same quantities are 0.11–0.15 and 0.04–0.30.

Hardware and cost. Sixty-two papers ran on H100 and two on GH200, with all states of a paper on one GPU. With two processes sharing a GPU, AM takes 2,281 s per paper: 364 s to generate its fit bank and 1,917 s of orthogonal matching pursuit (OMP) key selection and fitting over 288 KV heads. Teacher targets take 11 s and the 256 refinement updates 134 s. The whole study used 25.5 allocated GPU hours.

On Qwen3-8B, AM retains a large part of the document: it scores 25 points above the floor on constrained recall and 29 points above it on QA. Training-free KVzip selection at the same budget is clearly below AM. NCC stays well above KVzip. Its advantage over AM, however, is small: $+5.2$ points of greedy recall, with an interval that barely excludes zero, and $+1.8$ of constrained recall, with an interval that includes zero. On QA it loses 5.5 F1. The gain over AM that NCC shows on Qwen3.6-27B therefore does not carry over to this model.

C.1 WHERE THE DIFFERENCE ARISES

Document QA. Table 16 splits the Qasper questions by the answer type of their first reference, as in Appendix Q.6.

	Qwen3-8B (pure attention)			Qwen3.6-27B (hybrid)		
State	Greedy	Constrained	QA F1	Greedy	Constrained	QA F1
Full context	87.5	87.9	42.4	99.7	99.9	45.9
Attention Matching	39.6	43.0	**37.9**	16.7	62.3	38.1
NCC	**44.7**	**44.8**	32.5	**76.4**	**79.8**	**44.2**
KVzip	24.9	30.8	19.4	9.4	52.2	18.2
Floor (template rows only)	15.1	18.2	8.8	0.9	22.3	10.0

Table 14: **Both models on the 64 held-out papers** (512 rows per KV head; paper-averaged first reads). Greedy and constrained: number recall (%). The floor state keeps only the template rows: on Qwen3-8B this removes all document information; on Qwen3.6-27B the full-context recurrent state remains.

Difference	Greedy recall		Constrained recall		QA F1	
NCC minus AM	$+5.2$	$[0.2, 10.1]$	$+1.8$	$[-3.1, 6.7]$	-5.5	$[-10.9, -0.1]$
NCC minus KVzip	$+19.8$	$[14.0, 25.6]$	$+14.0$	$[8.2, 19.7]$	$+13.1$	$[5.7, 20.0]$
AM minus KVzip	$+14.6$	$[10.7, 18.7]$	$+12.1$	$[8.4, 16.1]$	$+18.5$	$[13.3, 23.6]$
Full context minus NCC	$+42.8$	$[37.6, 48.1]$	$+43.1$	$[37.9, 48.3]$	$+9.9$	$[4.8, 15.2]$
Qwen3.6-27B: NCC minus AM	$+59.7$	$[55.3, 64.1]$	$+17.5$	$[14.7, 20.5]$	$+6.1$	$[2.4, 9.7]$

Table 15: **Paired differences on Qwen3-8B** (points; 95% paper-level bootstrap intervals, 20,000 resamples). The last row repeats the Qwen3.6-27B result for comparison.

		F1				Abst.		NCC − AM		Qwen3.6-27B	
Answer type	n	Full	AM	NCC	KVzip	AM	NCC				
Extractive	126	44.6	40.0	36.4	13.8	11.1	7.9	-3.7	$[-9.0, 1.5]$	$+7.1$	$[3.6, 10.4]$
Abstractive	57	19.0	17.1	15.1	4.5	35.1	24.6	-2.0	$[-8.9, 5.0]$	$+4.2$	$[-1.0, 10.0]$
Yes/no	33	60.6	45.5	51.5	27.3	45.5	18.2	$+6.1$	$[-15.2, 25.0]$	$+24.2$	$[11.1, 39.4]$
Unanswerable	18	56.5	61.1	33.5	88.9	61.1	33.3	-27.6	$[-50.0, -9.4]$	-16.7	$[-33.3, 0.0]$
All answerable	216	40.3	34.8	33.1	13.4	22.7	13.9	-1.7	$[-6.7, 3.2]$	$+8.9$	$[5.8, 12.2]$

Table 16: **Qwen3-8B answer F1 by Qasper answer type** (question-weighted; Abst.: share of answers that begin with `Unanswerable`, %). Intervals: 95% paper-cluster bootstrap. Last column: the same difference on Qwen3.6-27B.

The significant part of the QA loss lies on the 18 unanswerable questions. There, F1 rewards abstention, and NCC lowers AM's abstention rate (61% to 33%), as it does on Qwen3.6-27B (78% to 61%). Without the unanswerable questions, NCC and AM do not differ significantly: -1.7 F1 $[-6.7, 3.2]$ question-weighted, and -2.9 $[-8.8, 2.7]$ paper-averaged. The difference from Qwen3.6-27B, where NCC gains 8.9 F1 $[5.8, 12.2]$ on the same answerable questions, is therefore mainly a missing gain on answerable questions rather than a specific harm.

Number recall. Table 17 decomposes the greedy answers and gives first-token statistics.

		Greedy answers			Constrained		Agreement (%)	
Model	State	Correct	Wrong	Abstain	correct	$P(\text{Un})$	Greedy	Constr.
Qwen3.6-27B	Full context	913	1	2	915	0.003	100	100
	Attention Matching	141	15	760	557	0.80	16.9	79.4
	NCC	688	163	65	720	0.08	85.6	90.5
Qwen3-8B	Full context	804	110	2	807	$\leq$0.002	100	100
	Attention Matching	353	360	203	386	$\leq$0.22	55.0	64.2
	NCC	397	497	22	397	$\leq$0.03	59.9	61.1

Table 17: **Outcomes and first-token statistics on the 916 number-recall questions** (first reads, question counts). $P(\text{Un})$: mean first-token probability of the first token of `Unanswerable`. On Qwen3.6-27B it is computed from the full-vocabulary normalizer, which was logged. The Qwen3-8B reads logged only the top-5 logits, so its values are normalized over the top 5 and are upper bounds; on Qwen3.6-27B the top 5 carry 99.9% of the mass. Agreement: first token equal to the full-context teacher's, under greedy and under constrained decoding.

The two models' AM states fail in different ways. On Qwen3.6-27B, AM abstains on 760 of 916 questions under greedy decoding. When the number grammar forbids abstention, it answers 557 correctly, so the right number is usually present but hidden behind abstention mass. NCC removes that mass ($P(\text{Un})$ from 0.80 to 0.08) and also improves the ranking of numbers (constrained first-token agreement from 79% to 91%).

On Qwen3-8B, AM abstains on only 203 questions. Its errors are mostly wrong numbers (360), and forbidding abstention adds only 33 correct answers. NCC again removes almost all abstention (203 to 22), but the freed questions mostly receive wrong numbers. Constrained recall barely moves (386 to 397 questions), and constrained first-token agreement with the teacher does not rise (64% to 61%). In both models, then, refinement suppresses abstention. It improves the ranking of the correct number on Qwen3.6-27B, as on Qwen3.5-9B (Table 11), but not on Qwen3-8B.

Content and position. Table 18 splits the constrained difference by content role and by position in the paper. On Qwen3-8B, NCC gains on quantities ($+5.8$ $[1.0, 10.7]$; 718 of 916 questions) and loses on identifiers and on references to sections, figures or equations (-26.9 $[-38.7, -13.7]$). No position quarter shows a significant difference.

Group	n	Qwen3-8B		Qwen3.6-27B	
Quantity	718	$+5.8$	[1.0, 10.7]	$+20.9$	[17.5, 24.3]
Identifier	79	-11.4	[-25.0, 2.6]	$+8.9$	[3.7, 14.9]
Reference or structure	78	-26.9	[-38.7, -13.7]	$+3.8$	[-3.2, 13.6]
Constant	21	-19.0	[-52.9, 10.5]	$+0.0$	[-15.0, 13.0]
Event time	15	$+20.0$	[0.0, 38.9]	$+13.3$	[0.0, 35.7]
First quarter	227	-4.0	[-12.2, 4.1]	$+18.1$	[13.0, 23.4]
Second quarter	226	$+3.1$	[-5.0, 10.9]	$+19.9$	[13.9, 26.1]
Third quarter	234	$+1.7$	[-5.3, 8.7]	$+20.9$	[15.7, 26.3]
Last quarter	229	$+3.9$	[-4.0, 12.2]	$+12.2$	[6.9, 18.0]

Table 18: **NCC minus AM in constrained recall by content role and position** (question-weighted points; 95% paper-cluster bootstrap intervals). Roles as in Appendix Q.4; unlike Table 46, scores here are question-weighted.

Structural differences between the models. Three candidate explanations can be checked against the logged data.

- *The recurrent state does not hold the numbers.* On Qwen3.6-27B, the refined KV rows sit beside a fixed full-context recurrent state, so one could suppose that this state carries the facts that refinement then exposes. The floors argue against this. With no document rows, the Qwen3.6-27B state that keeps the full recurrent state reaches 22.3% constrained recall and 10.0 F1. The Qwen3-8B state that keeps nothing of the document reaches 18.2% and 8.8. The recurrent state alone thus holds little of the tested detail.
- *The teacher's targets are less precise on Qwen3-8B.* This is the clearest difference. On the source continuations that NCC is trained to match, the Qwen3.6-27B full-context teacher predicts digit tokens almost deterministically (mean negative log-likelihood of the true digit 0.004 nats, top-1 99.9%). The Qwen3-8B teacher does not (0.30 nats, top-1 97.4%; over all tokens 0.20 against 0.58 nats). Its full-context number recall is also lower (87.9% against 99.9% constrained). A forward-KL student of this teacher is therefore trained toward less certain number distributions.
- *The objective fits equally well; the state is larger.* The objective itself fits equally well on both models. The median ratio of second-pass to first-pass loss on the same span is 0.44 on Qwen3-8B and 0.41 on Qwen3.6-27B. What differs is the size of the refined state. On Qwen3-8B the refined state spans all 288 KV heads (33.5M coordinates), against 64 heads (14.8M) on Qwen3.6-27B, trained with the same 128 spans and 256 updates. Relative to the full-context state, AM already recovers a smaller share of the recall on Qwen3-8B: 36% of the gap between floor and full context in constrained recall, against 52% on Qwen3.6-27B. After refinement the shares are 38% and 74%.

These comparisons are correlational and do not identify a single cause. With the recipe unchanged except for the scale-matched learning rate, NCC's advantage over AM on number recall does not carry over to this pure-attention model: on Qwen3-8B it suppresses abstention without improving which number is ranked first. Its advantage over training-free KVzip selection holds on both models.

D GENERATION-FREE INITIALIZATION AND RUN-TO-RUN VARIATION

A pipeline without generated text. The refinement of NCC generates nothing, but the AM recipe we use fits its reference queries on model-generated sequences (9.7K generated tokens per paper on average, 6.8K–11.7K). AM also has a repeat-prefill variant (Zweiger et al., 2026): the model is asked to repeat the document and the document itself is teacher-forced as the answer, so the reference queries come from a prefill. We refit the 512-row AM state on this sequence alone, with the fitting code, query sampling, the seeds of AM's key selection by orthogonal matching pursuit (OMP; Pati et al., 1993) and template rows unchanged, and ran the unchanged refinement from it; no token is generated anywhere in this pipeline. Because the document is shorter than the query-token cap on 60 of the 64 papers, the variant uses fewer reference queries (32.4K instead of 49.9K query vectors per KV head on average). All states in Table 19 were read in one run; the standard initialization is the AM state of Table 3.

Repeat-prefill AM is slightly weaker than standard AM (-2.5 points $[-4.8, -0.3]$ greedy, -2.5 $[-5.3, 0.4]$ constrained, -0.3 F1 $[-2.4, 1.9]$); its per-layer attention error is slightly lower on repeat-prefill queries but higher on queries from generated text (relative output error 0.218 against 0.175). Refinement removes this gap. The generation-free pipeline reaches 74.1% greedy and 78.6% constrained recall and 45.5 F1, against 75.4%, 78.6% and 46.2 for NCC from the standard initialization in the same run (paired differences -1.3 $[-3.7, 1.2]$, 0.0 $[-2.3, 2.4]$ and -0.6 $[-3.1, 1.9]$), and improves on standard AM by 16.9 points [13.9, 20.0] of constrained recall and 7.1 F1 [2.8, 11.5]. The gains of NCC therefore do not depend on generated reference queries in this setting. Fitting the repeat-prefill AM takes 114 s per paper, whereas standard AM first generates its reference sequences (730 s per paper) and then fits them (158 s when rebuilt from its stored sequences in the ablation run; 179 s of query extraction and fitting in Table 44). End to end, the generation-free pipeline (prefill, repeat-prefill AM, teacher targets and 256 updates) compiles a paper in 242 s, against 1,226 s for the standard pipeline (medians over the 64 papers; Table 24).

Run-to-run variation. Apart from the order in which the 128 spans are visited, the refinement makes no random choice, but the GPU stack is not bitwise deterministic: retraining NCC in document order from the same AM tensors reproduced the earlier run bit for bit on 19 of the 64 papers, and on the others small forward- or backward-pass differences grew over 256 Adam updates into a different cache. We therefore compare four runs from identical initial tensors: the run of Table 3, a document-order rerun, and two runs that visit the spans in a fresh random order in each epoch. Paper-averaged scores are stable at 75.7 ± 0.6 greedy, 79.4 ± 0.6 constrained and 45.0 ± 1.4 F1 (mean $\pm$ SD; ranges 75.2–76.6, 78.6–80.1 and 43.1–46.2). No run differs from the document-order rerun on number recall, and one random-order run is 3.1 F1 lower on QA $[-6.0, -0.5]$. Individual answers vary more: runs with different span orders agree on 77–79% of constrained answers and 47–48% of QA answers, against 99.5% and 96.2% when one AM state is read twice. On papers where the rerun diverged, it changes per-paper constrained recall by as much as a new span order does (5.4 against 5.5–6.5 points on average). The gain over AM holds in every run, at $+16.9$ to $+18.4$ points of constrained recall and $+4.7$ to $+7.8$ F1, with every interval above zero. A fifth document-order run from the same tensors, the in-run default of Appendix G (76.0%, 79.5% and 45.6), leaves all three ranges unchanged: over five runs the scores are 75.7 ± 0.6 greedy, 79.4 ± 0.5 constrained and 45.1 ± 1.3 F1. That run gains $+17.8$ points [14.8, 20.9] of constrained recall and $+7.0$ F1 [3.1, 10.9] over AM and matches the Table 3 run bit for bit on 25 of the 64 papers. Single-run QA comparisons should be read against this run-to-run standard deviation of 1.3–1.4 F1.

State (64 papers, 512 rows)	Greedy	Constr.	QA F1
AM, standard reference queries (9.7K generated tokens)	16.4	61.7	38.4
AM, repeat-prefill reference queries (none generated)	13.9	59.2	38.1
NCC from standard AM	75.4	78.6	46.2
NCC from repeat-prefill AM (none generated)	74.1	78.6	45.5
NCC, four runs from the same AM tensors (mean $\pm$ SD)	75.7 ± 0.6	79.4 ± 0.6	45.0 ± 1.4
document order (Table 3)	76.6	79.3	45.9
document order, rerun	75.4	78.6	46.2
random span order, seed 1	75.4	80.1	43.1
random span order, seed 2	75.2	79.5	44.7

Paired difference	Greedy		Constr.		QA F1	
Repeat-prefill AM − AM	−2.5	[−4.8, −0.3]	−2.5	[−5.3, 0.4]	−0.3	[−2.4, 1.9]
Generation-free NCC − NCC	−1.3	[−3.7, 1.2]	0.0	[−2.3, 2.4]	−0.6	[−3.1, 1.9]
Generation-free NCC − AM	+57.8	[53.3, 62.3]	+16.9	[13.9, 20.0]	+7.1	[2.8, 11.5]
Seed 1 − document order	+0.0	[−2.7, 2.8]	+1.5	[−0.5, 3.6]	−3.1	[−6.0, −0.5]
Seed 2 − document order	−0.2	[−2.6, 2.2]	+0.9	[−1.2, 3.0]	−1.5	[−3.9, 1.1]
Rerun − Table 3 run	−1.2	[−3.3, 0.9]	−0.7	[−2.4, 1.0]	+0.2	[−1.5, 1.8]

Table 19: **Generation-free initialization and run-to-run variation** (64 papers, 512 rows, first reads). Top: paper-averaged scores; all states except the Table 3 run were read in one run. Bottom: paired differences in points with 95% paper-level bootstrap intervals (20,000 resamples); NCC, AM and "document order" refer to the document-order rerun and the standard AM state read in the same run.

E Low-precision serving and storage

FP8 and 4-bit formats are now standard for serving (Liu et al., 2024b; Hooper et al., 2024). We test whether the compiled state survives them on the 64 held-out papers (916 number-recall and 234 QA questions, first reads, unchanged prompts, decoders and scorers). The FP4 reader is NVIDIA's official NVFP4 release of Qwen3.6-27B, produced with TensorRT Model Optimizer: NVFP4 weights with BF16 activations in the MLPs and output head, and FP8 weights and activations in the attention and Gated DeltaNet projections. We emulate it by writing its dequantized weights into the BF16 model and quantizing the FP8 layer inputs, so we report accuracy and stored bytes, not speed. *Compile once in BF16, serve in FP4*: the AM and NCC states are the BF16 ones, read by the FP4 model. Stored-row quantization rounds the 512 K/V rows to FP8 (e4m3, one scale per row) or NVFP4 (e2m1 with an e4m3 scale per 16 values) and dequantizes them at read time; the KV-quantization baselines apply the same formats, or KIVI-style INT4 and INT2 (Liu et al., 2024b), to the full-context cache.

The recurrent and convolutional states stay BF16 throughout. All states of a paper were read in one process on one GPU (GH200 for 22 papers, GB200 for 26 and GB300 for 16), so every paired comparison is within one GPU, and the plan was fixed before any evaluation read.

State	Reader	K/V MiB	K/V ratio	Greedy	Constr.	QA F1
Full context	BF16	340	1.0×	99.7	99.9	46.0
Full context, KV FP8	BF16	173	2.0×	99.7	99.9	45.7
Full context, KV NVFP4	BF16	96	3.6×	99.5	99.9	45.4
Full context, KV KIVI-4	BF16	106	3.2×	99.6	99.8	45.7
Full context, KV KIVI-2	BF16	64	5.3×	99.2	99.7	44.7
AM	BF16	32	10.6×	16.2	61.7	38.4
AM, rows NVFP4	BF16	9	37.8×	15.7	61.9	37.7
NCC	BF16	32	10.6×	76.5	79.3	45.7
NCC, rows FP8	BF16	16.3	20.9×	76.2	79.4	44.6
NCC, rows NVFP4	BF16	9	37.8×	75.0	78.7	44.3
Full context	FP4	340	1.0×	99.4	99.8	45.3
AM (compiled in BF16)	FP4	32	10.6×	16.9	62.6	37.9
NCC (compiled in BF16)	FP4	32	10.6×	77.5	79.6	46.3
NCC (compiled in BF16), rows NVFP4	FP4	9	37.8×	76.2	79.2	44.3
NCC refined with the FP4 student	FP4	32	10.6×	77.8	79.5	44.1

Paired difference	Greedy		Constr.		QA F1	
FP4 reader: NCC − AM	+60.6	[55.9, 65.2]	+17.0	[14.0, 19.9]	+8.4	[4.2, 12.6]
NCC: FP4 reader − BF16 reader	+0.9	[−0.5, 2.3]	+0.3	[−0.9, 1.4]	+0.6	[−0.7, 1.9]
NCC rows: FP8 − BF16	−0.4	[−1.0, 0.2]	+0.1	[−0.5, 0.6]	−1.1	[−3.2, 0.4]
NCC rows: NVFP4 − BF16	−1.6	[−2.5, −0.6]	−0.6	[−1.5, 0.3]	−1.4	[−3.2, 0.3]
FP4 reader, NVFP4 rows − BF16 NCC	−0.3	[−1.8, 1.1]	−0.1	[−1.4, 1.1]	−1.4	[−3.5, 0.5]
FP4 reader: FP4 student − BF16-compiled	+0.3	[−1.6, 2.2]	−0.1	[−1.9, 1.8]	−2.1	[−4.7, 0.3]
NVFP4: NCC rows − full-context KV	−24.6	[−29.1, −20.1]	−21.2	[−25.4, −17.1]	−1.0	[−4.5, 2.4]

Table 20: **Low-precision serving and storage** (Qwen3.6-27B, 64 papers, 512 rows, paper-averaged first reads). Top: scores and mean stored K/V bytes per paper; K/V ratio: mean full BF16 K/V bytes over mean stored K/V bytes (ratio of means; Table 45 gives medians). The 75.8 MiB recurrent and convolutional state is added to every state. Bottom: paired differences in points with 95% paper-level bootstrap intervals (20,000 resamples).

Compiled once in BF16, NCC keeps its accuracy under the official FP4 reader: its scores move by $+0.9$, $+0.3$ and $+0.6$ points (all intervals include zero), and its gain over AM is unchanged ($+17.0$ points of constrained recall [14.0, 19.9] and $+8.4$ F1 [4.2, 12.6]). Refining with the FP4 model as the student gives statistically indistinguishable accuracy, so compilation need not be repeated per serving precision. The compiled rows also quantize well: FP8 rows change no score significantly, and NVFP4 rows cost 1.6 points of greedy recall [0.6, 2.5] and nothing significant under constrained decoding. With both the FP4 reader and NVFP4 rows, every difference from the BF16 pipeline includes zero, while the K/V rows take 9 MiB, 37.8× fewer bytes than the BF16 full-context cache (16.6–63.6× across papers).

Quantizing the full-context cache is nearly lossless on these papers, down to 2-bit KIVI, but it reduces K/V bytes only 2–5.3×, since every token keeps a row. Compaction and quantization therefore act on different axes, and they compose: compaction removes rows and quantization shrinks each row (Figure 11). At 37.8×, NCC with NVFP4 rows trails NVFP4 full context by 21 points of constrained recall, about the gap between the BF16 states, and matches it on QA (-1.0 F1 [-4.5, 2.4]). Because the recurrent state stays in BF16 here, the total state shrinks 4.9× with NVFP4 rows, against 3.9× with BF16 rows.

E.1 Quantizing the recurrent state

The 75.75 MiB recurrent and convolutional state, which every compact state carries, limits the total compression on single papers to 3.9×. We therefore also round it to FP8 (39.0 MiB with scales) or

NVFP4 (21.3 MiB), with the same formats as above, and read every state with the BF16 reader on GH200 GPUs; the plan and the analysis were fixed before any evaluation read.

K/V rows	Recurrent state	Total MiB	Total ratio	Greedy	Constr.	QA F1
Full context						
BF16	BF16	415.6	1.0×	99.7	99.9	45.8
FP8	FP8	211.6	2.0×	99.7	99.9	45.8
NVFP4	NVFP4	116.9	3.6×	99.5	99.9	46.2
NCC, 512 rows						
BF16	BF16	107.8	3.9×	76.6	79.3	45.8
BF16	FP8	71.1	5.9×	76.6	79.2	45.9
BF16	NVFP4	53.4	7.8×	76.6	79.6	46.0
FP8	FP8	55.3	7.5×	76.1	79.0	45.2
NVFP4	FP8	48.1	8.7×	75.2	78.8	44.7
NVFP4	NVFP4	30.4	13.7×	75.8	78.8	44.8
AM, 512 rows						
BF16	BF16	107.8	3.9×	16.2	62.0	38.4
NVFP4	NVFP4	30.4	13.7×	16.1	62.5	37.1

Table 21: **Quantizing the recurrent state** (Qwen3.6-27B, 64 single papers, BF16 reader, paper-averaged first reads). Total MiB: mean stored state (K/V rows, β, recurrent and convolutional state); total ratio: BF16 full-context state over stored state (ratio of means; 7.4–21.4× across papers for the all-NVFP4 state).

Quantizing the recurrent state alone changes no score of NCC significantly (FP8: -0.1 [-0.3, 0.0] constrained recall; NVFP4: $+0.2$ [-0.3, 0.9]). Storing both the rows and the state in FP8 (55.3 MiB, 7.5×) changes no score significantly either (-0.5 [-1.1, 0.0] greedy, -0.3 [-0.8, 0.1] constrained, -0.6 [-1.5, 0.0] F1). With both the rows and the state in NVFP4, the whole compiled state takes 30.4 MiB, 13.7× less than the BF16 full-context state, and every difference from the BF16 state includes zero (-0.8 [-1.8, 0.2] greedy, -0.5 [-1.5, 0.5] constrained, -1.1 [-2.8, 0.6] F1); read by the FP4 model, the same state scores 76.2, 79.0 and 45.0, again within noise of the all-BF16 pipeline (-0.4 [-2.0, 1.1] greedy, -0.3 [-1.7, 0.9] constrained, -0.8 [-2.7, 1.0] F1). NCC's gain over AM is unchanged in every format ($+16.3$ to $+17.3$ points of constrained recall, every interval above zero). On long documents the fixed-size state compresses further: the all-NVFP4 512-row state is 91× smaller than the BF16 full state of an 8-paper document and 180× smaller than that of a 16-paper document (72.7–202.5× across documents), with constrained recall of 37.4% and 31.9% against 37.8% and 32.7% for the BF16 compiled state; on 8-paper documents at 512 rows, NVFP4 rows cost 3.2 F1 [0.8, 6.2], whereas quantizing only the recurrent state costs no significant F1. With rows and state in NVFP4, NCC keeps its gain over AM on long documents at 512 rows: $+10.1$ and $+5.7$ points of constrained recall and $+21.4$ and $+14.0$ F1 on 8- and 16-paper documents, every interval above zero.

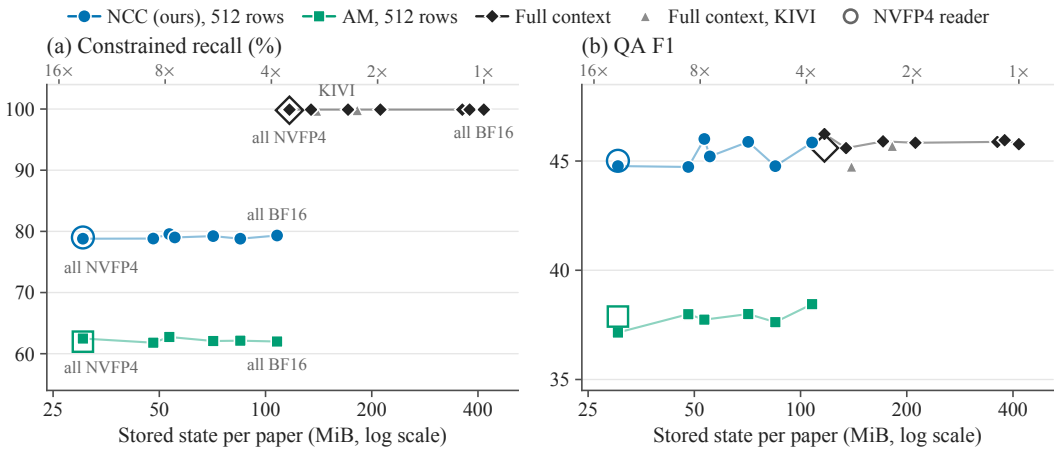


Figure 11: **Compaction and quantization compose** (Qwen3.6-27B, 64 papers, first reads). Each marker is one stored state; x: mean stored bytes per paper (K/V rows or cache, β, recurrent and convolutional state; log scale); top axis: compression of the BF16 full-context state. Filled markers, read by the BF16 model and joined in order of size: every storage format of the state-quantization run (K/V and recurrent state each in BF16, FP8 or NVFP4; Appendix E.1). Hollow: all-NVFP4 states read by NVIDIA's NVFP4 model. Triangles: full-context caches in KIVI-4 and KIVI-2 (Table 20). No NCC state differs significantly from BF16 NCC in constrained recall or QA F1, and on QA F1 each of the seven NCC states (30.4–107.8 MiB) is statistically indistinguishable from each of the seven full-context states (116.9–415.6 MiB; 49 paired comparisons, all 95% intervals include zero).

F SERVING COST AND BREAK-EVEN

We measure what serving from a compiled state costs, on one GH200 GPU (96 GB; one benchmark process per node) with the same Hugging Face implementation of Qwen3.6-27B (BF16, SDPA) for every state. Documents are the 10 development papers (single paper) and the 8- and 16-paper documents of Appendix I; only their text is used (timing only, no question or label is read). Every request asks one fixed synthetic question and greedily decodes exactly 128 tokens (64 in batch runs); each measurement is repeated 5 times after 1 warm-up. Absolute times are those of this research stack (eager decoding without CUDA graphs or paged attention), not of an optimized serving engine; the comparisons between states are the result. The per-row bias β of AM and NCC enters attention as an additive float mask on the keys (SDPA here); NCC refined from KVzip rows, which carry no bias, still reaches 72.6% constrained recall against 61.7% for AM (Table 3). At batch 1, TPOT barely depends on the state (59.5–64.3 ms for every state and document; Table 25), so the per-token cost comes from the model rather than the cache; the compact state matters for memory, which bounds the batch (Table 23 and Figure 12). Offloading full states to host memory would hold more of them per node, but a request still reads every row of its full state at each decoding step: served from host memory, a 16-paper full state reaches its first token after 0.239 s and still takes 7.31 GiB of GPU memory per request, against 0.120 s and 0.16 GiB for the compiled state served from host memory. Most of the standard compile time is AM's fit-sequence generation: the generation-free pipeline (Appendix D) compiles a paper in 242 s and pays for itself after 412 queries, and, by an upper bound on its cost, a 16-paper document after at most 91 queries (Table 24). At 2048 rows, where AM's fit takes 2,677–3,576 s per long document (Table 33), the standard pipeline pays for itself after a median of 770 queries on 8-paper and 472 on 16-paper documents; the 2048-row state is read as fast as the 512-row one (Table 22).

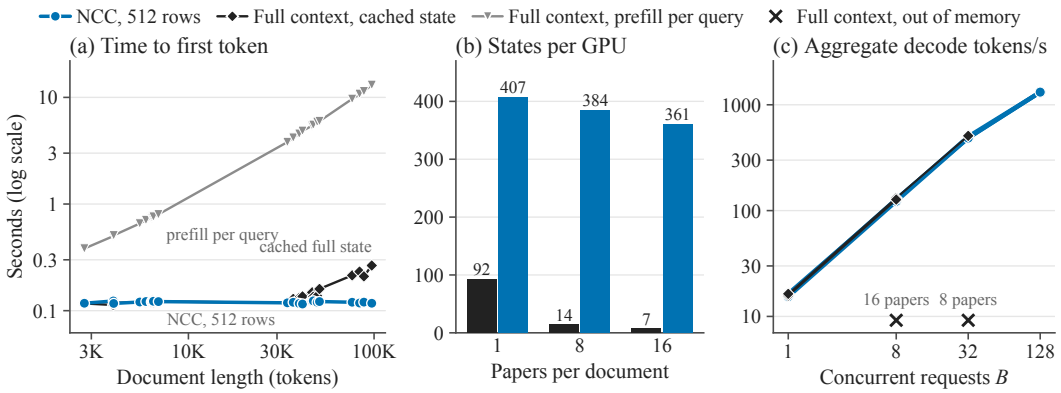


Figure 12: **Serving from a compiled state** (Qwen3.6-27B, BF16, one GH200; timing only). (a) Time to first token at batch 1 against document length, per-document median of 5 repetitions: prefilling for every query grows with length (0.38–13.1 s), a cached full state grows slowly (0.11–0.27 s), and the 512-row compiled state stays at 0.12 s. (b) Copies of one document's resident state that fit next to the weights (fill test; the NCC fill ran with the group's full-context state resident). (c) Aggregate decode throughput of B concurrent requests (64 tokens, median of 5): both states decode at nearly the same rate at equal B, but full-context batches run out of memory at B=8 (16-paper documents) and B=32 (8-paper documents), whereas compiled states reach B=128 (1,317 tokens/s); full context was not run at B=128 on single papers.

State	TTFT (s)	TPOT (ms)	tok/s	Peak (GiB)	Request (GiB)	State (MiB)
Single paper, 2.8K–6.9K tokens (10 documents)						
Full, prefill per query	0.726	62.1	16.1	51.7	1.61	–
Full, cached state	0.120	62.0	16.1	51.2	0.61	457.8
NCC, 512 rows	0.121	63.2	15.8	50.4	0.16	107.8
8-paper documents, 34.1K–50.8K tokens (8 documents)						
Full, prefill per query	5.141	61.6	16.2	61.1	11.00	–
Full, cached state	0.136	61.1	16.4	56.6	3.77	2,812
NCC, 512 rows	0.120	62.4	16.0	50.4	0.16	107.8
NCC, 2048 rows	0.119	62.3	16.0	50.7	0.36	204.0
16-paper documents, 76.2K–97.2K tokens (4 documents)						
Full, prefill per query	11.090	61.2	16.4	71.6	21.48	–
Full, cached state	0.223	60.9	16.4	62.7	7.31	5,440
NCC, 512 rows	0.119	62.3	16.0	50.4	0.16	107.8
NCC, 2048 rows	0.120	62.3	16.0	50.7	0.36	204.0

Table 22: **Per-request serving cost** (batch 1). Median over documents of the per-document median of 5 repetitions. TTFT: request arrival to first token, including the document prefill (prefill per query) or copying the resident state into a request buffer (cached states). TPOT: time per output token over tokens 2–128; decode speed (tok/s) $= 1/\mathrm{TPOT}$ per request. Peak: model weights + this resident state + the measured request memory, i.e. the peak of a GPU that serves only this state (the benchmark process held all states of a document at once; its measured peak is in the released results). Request: measured peak increase during the request; for the cached full state it includes a copy of the prefix keys and values, because the cache grows by concatenation. State: resident keys and values (and β for NCC) plus the Gated DeltaNet recurrent and convolution states.

State	States/GPU	Aggregate decode (tok/s) B=1	B=8	B=32	B=128	TTFT, B=32 (s)
Single paper						
Full context, cached state	92	16.2	131.4	509.0	–	0.53
NCC, 512 rows	407	16.4	125.8	500.7	1,317	0.32
8-paper documents						
Full context, cached state	14	15.7	127.2	OOM	–	OOM
NCC, 512 rows	384	15.4	122.5	486.6	1,317	0.31
16-paper documents						
Full context, cached state	7	16.4	OOM	OOM	–	OOM
NCC, 512 rows	361	16.0	128.2	485.3	1,317	0.32

Table 23: **Batch serving on one GH200.** States/GPU: copies of one document's resident state that fit next to the model weights (fill test, leaving no room for requests; the NCC fill test ran with the group's full-context state still resident, which lowers its count by 4, 28 and 52 NCC states, respectively). B: aggregate decode throughput of B concurrent requests decoded in lockstep (64 tokens; median of 5); per-request TPOT is 61–66 ms up to B=32 and 97 ms at B=128. Full-context batches hold B copies of the median-length document of the group (the cost does not depend on content); NCC batches use the states of different documents (cycled). OOM: the batch does not fit in GPU memory; –: not run.

Document	Compile (s) Standard	Gen.-free	Saved/query (s)	Break-even (queries) Standard	Gen.-free	Memory saved (MiB)
Single paper	1,226	242	0.59	2,086	412	350
8-paper documents	1,457	≤590	5.02	291	≤118	2,704
16-paper documents	2,468	≤991	10.95	224	≤91	5,333

Table 24: **Break-even of compilation** against prefilling the document for every query, in GPU-seconds on one GH200. Compile: per-document wall-clock cost recorded by the earlier runs on the same hardware and software. Single paper: median over the 64 evaluation papers; standard = prefill + AM fit-sequence generation + AM fit + teacher targets + 256 updates; generation-free (Appendix D) = prefill + repeat-prefill AM fit + teacher targets + 256 updates. Long documents: per document (median shown); generation-free was not run at this length, so it is bounded by the standard cost without fit-sequence generation (its AM fit replays only the repeat sequence, one of the standard fit sequences, under the same query budget); both include one-time kernel warm-up. Saved/query: prefill-per-query cost minus NCC cost for a 16-token answer (TTFT + 15 TPOT; single paper: median over the development papers). Break-even: number of queries after which compiling and serving the NCC state costs less GPU time than prefilling for every query. Memory saved: full resident state minus NCC state.

		TTFT (s)			TPOT (ms)	
Document	Tokens	Prefill per query	Cached full	NCC 512	Cached full	NCC 512
Single paper						
1904.11942	2,751	0.38 [0.38, 0.38]	0.117 [0.117, 0.118]	0.117 [0.117, 0.117]	59.7 [59.6, 59.8]	60.8 [60.8, 60.9]
1907.05664	3,941	0.50 [0.50, 0.50]	0.113 [0.095, 0.121]	0.123 [0.122, 0.124]	62.0 [61.8, 62.0]	64.3 [63.6, 65.8]
1908.09156	3,957	0.51 [0.51, 0.51]	0.118 [0.118, 0.119]	0.117 [0.116, 0.117]	60.6 [60.5, 61.5]	61.9 [61.5, 62.1]
1905.12260	5,470	0.66 [0.65, 0.66]	0.120 [0.119, 0.121]	0.121 [0.119, 0.122]	61.8 [61.7, 62.7]	62.8 [62.4, 63.4]
2004.03762	5,905	0.71 [0.71, 0.71]	0.120 [0.120, 0.123]	0.122 [0.121, 0.122]	62.5 [62.3, 63.4]	63.3 [63.2, 63.4]
1909.03242	6,322	0.75 [0.75, 0.75]	0.121 [0.119, 0.122]	0.121 [0.118, 0.122]	62.0 [61.3, 62.2]	63.1 [62.8, 63.3]
1911.00133	6,362	0.74 [0.74, 0.75]	0.121 [0.117, 0.121]	0.120 [0.120, 0.121]	61.6 [61.4, 61.7]	62.8 [61.9, 62.9]
1912.10806	6,404	0.76 [0.76, 0.76]	0.124 [0.123, 0.124]	0.123 [0.122, 0.123]	63.4 [62.5, 64.1]	64.3 [63.8, 64.6]
1807.07961	6,477	0.76 [0.76, 0.76]	0.120 [0.119, 0.123]	0.123 [0.121, 0.123]	62.3 [61.6, 63.5]	63.9 [63.2, 64.2]
2001.05493	6,894	0.80 [0.80, 0.80]	0.122 [0.121, 0.122]	0.122 [0.121, 0.122]	62.4 [62.3, 63.9]	63.3 [63.2, 64.3]
8-paper documents						
A08-1	34,128	3.81 [3.81, 3.81]	0.118 [0.118, 0.122]	0.118 [0.116, 0.120]	60.9 [59.3, 61.1]	62.1 [62.1, 62.2]
A08-3	36,707	4.19 [4.18, 4.19]	0.129 [0.128, 0.129]	0.120 [0.119, 0.120]	60.9 [59.8, 60.9]	61.8 [61.8, 62.7]
A08-2	39,521	4.56 [4.55, 4.56]	0.132 [0.132, 0.132]	0.117 [0.116, 0.117]	60.4 [60.3, 60.4]	61.2 [61.1, 61.3]
A08-6	41,159	4.85 [4.85, 4.85]	0.136 [0.136, 0.137]	0.115 [0.114, 0.116]	60.0 [59.8, 61.1]	62.0 [61.1, 62.5]
A08-4	46,407	5.43 [5.43, 5.44]	0.149 [0.148, 0.149]	0.122 [0.121, 0.122]	61.8 [61.6, 62.0]	63.0 [62.4, 63.9]
A08-7	47,042	5.52 [5.52, 5.53]	0.135 [0.135, 0.135]	0.124 [0.122, 0.124]	63.5 [63.3, 63.7]	62.8 [62.5, 62.9]
A08-0	49,412	5.89 [5.89, 5.89]	0.137 [0.137, 0.138]	0.120 [0.118, 0.121]	61.3 [61.0, 61.5]	62.6 [62.0, 62.6]
A08-5	50,819	5.99 [5.98, 5.99]	0.160 [0.160, 0.160]	0.122 [0.122, 0.122]	62.0 [61.9, 62.1]	63.2 [63.1, 63.3]
16-paper documents						
B16-1	76,191	9.67 [9.66, 9.67]	0.214 [0.214, 0.214]	0.120 [0.120, 0.123]	61.3 [60.8, 62.3]	62.9 [62.6, 63.1]
B16-0	83,503	10.77 [10.76, 10.78]	0.232 [0.232, 0.233]	0.118 [0.117, 0.119]	60.5 [60.2, 60.7]	61.8 [61.7, 61.8]
B16-3	88,164	11.40 [11.40, 11.41]	0.210 [0.209, 0.210]	0.120 [0.096, 0.121]	62.4 [61.5, 63.7]	64.0 [63.2, 64.2]
B16-2	97,189	13.08 [13.06, 13.10]	0.265 [0.265, 0.265]	0.117 [0.117, 0.117]	59.5 [59.4, 59.6]	60.7 [60.6, 60.8]

Table 25: **Per-document latency**: median [interquartile range] of 5 repetitions (batch 1, same implementation for every state).

G SENSITIVITY TO THE RECIPE'S SETTINGS

NCC has four settings: the number of source spans (128), the maximum cue length (32 tokens), the number of updates (256, two passes over the spans) and the learning rate (0.01). We change one at a time on the 64 papers, from the same AM tensors and with the same readers, and compare with the default retrained in the same run; the plan was fixed before any evaluation read. Because the GPU stack is not bitwise deterministic (Appendix D), we also compare each setting with the range of five default runs from the same AM tensors.

Setting	Value	Greedy	Constr.	QA F1	Constr. − default	
Default	128, ≤32, 256, 0.01	76.0	79.5	45.6		
Default, 5 runs	range	75.2–76.6	78.6–80.1	43.1–46.2		
Spans	64 (128 updates)	72.0	78.8	44.7	−0.7	[−2.7, 1.3]
	256 (512 updates)	78.4	79.8	43.8	+0.3	[−1.7, 2.3]
	64, 4 passes (256 updates)	74.3	80.7	45.2	+1.1	[−1.1, 3.4]
	256, 1 pass (256 updates)	72.9	73.9	42.7	−5.6	[−8.1, −3.2]
Cue	≤8 tokens	70.1	76.8	45.9	−2.7	[−4.7, −0.7]
	≤64 tokens	75.8	78.2	44.4	−1.3	[−3.6, 0.9]
Learning rate	0.005	65.2	76.2	43.9	−3.3	[−5.3, −1.4]
	0.02	74.3	74.4	39.7	−5.1	[−8.2, −2.1]
AM (no refinement)		16.4	61.7	38.6		

Table 26: **Sensitivity of NCC to its settings** (default: 128 spans, cue of at most 32 tokens, 256 updates, learning rate 0.01), changed one at a time on the 64 papers (512 rows, first reads, paper-averaged). All runs start from the same AM tensors; the default is retrained in the same run. Last columns: paired difference from the default with 95% paper-level bootstrap intervals (20,000 resamples). Greedy differences: -4.1, $+2.3$, -1.7, -3.1, -5.9, -0.2, -10.9 and -1.7 points; QA F1 differences: -0.9, -1.7, -0.4, -2.8, $+0.3$, -1.1, -1.7 and -5.9, in row order.

Every setting keeps a large gain over AM (greedy and constrained intervals above zero in all eight rows), retaining 71–102% of the default's gain in constrained recall when the two passes are kept, and 106% and 69% when 64 or 256 spans are visited within the same 256 updates (Figure 13). Halving or doubling the number of spans changes constrained recall by less than one point and doubling the

cue length by 1.3 points, all within or near the range of default runs, and 64 spans visited four times reach 80.7% (default 79.5%). The learning rate matters most: halving it slows greedy recall (−10.9) and doubling it costs constrained recall and QA F1, and the shortest cue (8 tokens) costs 2.7 points of constrained recall. With a single pass over 256 spans, constrained recall drops by 5.6 points, so revisiting each span helps more than finer spans. Along the number of updates, checkpoints of one 1,024-update trajectory in the ablation run reach 68.3, 74.4, 79.3 and 84.1% constrained recall after 64, 128, 256 and 1,024 updates. Even 64 updates, which visit only the first half of the spans once, improve on AM by 6.6 points [4.4, 8.9], and 1,024 updates add 4.8 points [2.2, 7.5] to the default (Table 3), so the default is a cheap setting rather than the best one.

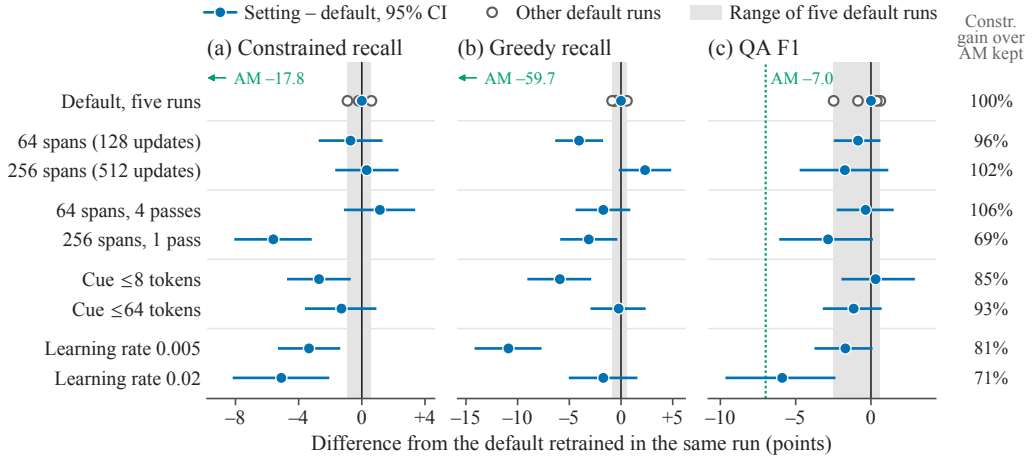


Figure 13: **Every setting keeps most of NCC's gain over AM in number recall.** Each row changes one setting of Table 26; markers give the paired paper-averaged difference from the default retrained in the same run, whiskers 95% paper-level bootstrap intervals (64 papers, 20,000 resamples). Top row and grey band: the five default runs from the same AM tensors (Appendix D), relative to the in-run default. Green: AM without refinement, off the axis in (a) and (b). Right: share of the default's constrained-recall gain over AM that each setting keeps.

H TRAINING-FREE BASELINES AND THE RECURRENT-STATE-ONLY CONTROL

Construction. Every state in this section keeps 512 KV rows per KV head in each of the 16 attention layers, the same template rows as Attention Matching (system prompt, document delimiters and question marker; 61 rows on every paper), the complete full-context recurrent and convolutional state M_D, C_D, the original post-RoPE keys and values, their original positions, and a zero mass correction. The remaining 451 rows per head hold document tokens chosen as follows. *Recurrent state only* keeps no document rows. *StreamingLLM* keeps the first 4 document tokens as attention sinks and the last 447 document tokens (Xiao et al., 2024). *SnapKV* keeps the last 64 document tokens as its observation window and, per head, the 387 earlier tokens that receive the most attention from that window during prefill (summed over the window and the 6 grouped query heads, average-pooled with kernel 7) (Li et al., 2024); no evaluation question is used. *KVzip* keeps, per head, the 451 document tokens that receive the highest maximum attention while the model is teacher-forced to reconstruct the document in chunks of at most 2,048 tokens from a fresh copy of the full-context cache (Kim et al., 2025); we use a uniform per-head budget, so KVzip does not use its non-uniform head allocation. All states are read with the same reader, prompts, decoders and scoring as the main comparison. After the shared prefill, building the states takes 0.02 s (recurrent state only), 0.03 s (StreamingLLM), 0.1 s (SnapKV) and 4.4 s (KVzip) per paper on average. We reimplemented both scorings in the per-head cache path shared by all states: KVzip's chunked reconstruction with the maximum attention each key receives, and SnapKV's pooled observation-window votes. Question-aware selection, as in the original SnapKV and H2O, keeps the full cache resident until each question arrives, so it saves neither prefill nor storage when one document serves many queries, and query-aware eviction degrades when one compressed cache serves several queries (Kim et al., 2025).

NCC minus	Greedy recall		Constrained recall		QA F1	
Recurrent state only	+75.5	[70.8, 80.0]	+57.5	[53.0, 62.1]	+34.2	[28.4, 39.7]
StreamingLLM	+61.8	[57.0, 66.6]	+44.1	[39.4, 48.8]	+21.3	[16.3, 26.3]
SnapKV	+65.0	[60.1, 70.0]	+43.0	[38.9, 47.1]	+22.4	[17.4, 27.4]
KVzip	+66.9	[62.3, 71.4]	+27.7	[23.3, 32.0]	+26.0	[20.7, 31.2]

Table 27: **Paired differences against the training-free states** (percentage points, 95% paper-level bootstrap intervals, 20,000 resamples). Positive values favor NCC.

Where selected rows succeed. Table 28 splits greedy and constrained recall by the position of the masked number within its paper (questions pooled over papers). Eviction keeps recent tokens and therefore answers questions about the end of a paper: StreamingLLM reaches 41.0% greedy recall in the last quarter but 3–9% in the first three. SnapKV shows the same bias through its end-of-document observation window. KVzip covers the document more evenly but thinly. NCC is uniformly strong across the document.

	Greedy				Constrained			
State	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Recurrent state only	1.8	0.0	0.4	0.9	26.4	23.0	17.9	19.2
StreamingLLM	4.0	3.1	9.0	41.0	27.3	26.5	33.3	52.8
SnapKV	3.5	5.3	10.3	25.8	34.8	28.3	35.0	44.1
KVzip	11.9	9.3	5.6	5.7	55.9	52.7	45.7	45.4
Attention Matching	15.9	10.2	14.1	21.4	62.6	57.1	60.3	63.3
NCC	**78.9**	**74.8**	**77.4**	**69.4**	**80.6**	**77.0**	**81.2**	**75.5**

Table 28: **Recall by position of the masked number** (quarters of the paper; questions pooled over papers; %).

H.1 Stronger training-free baselines with adaptive budgets

Our KVzip and SnapKV baselines give every KV head the same number of rows. Their original papers, and budget-allocation methods built on them, allocate rows non-uniformly, so we re-ran them with those rules at the same average of 512 rows per KV head: every arm keeps the 61 template rows in each head and $16 \times 4 \times 451 = 28{,}864$ document rows in total, the same 32,768 stored rows as AM and NCC, with the full-context recurrent state. *KVzip with its own rule* scores tokens by chunked context reconstruction and takes one global top-k over all attention layers and KV heads (Kim et al., 2025); *KVzip with adaptive heads* keeps an equal number of rows per layer and allocates them across heads. Both KVzip arms score rows as the KVzip repository does: the document is split into 2,000-token chunks, each reconstructed after a repeat prompt, and a row's score is the maximum attention it receives while its own chunk is reconstructed, with the softmax over the template rows, that chunk and the reconstruction. Our uniform KVzip of the main comparison instead normalizes over all rows and takes the maximum over every chunk, so we also report the repository scoring with uniform heads, which isolates the allocation rule. *SnapKV with Ada-KV* allocates each layer's rows across heads by a flattened top-k over the heads' pooled window scores, with the floor $\alpha = 0.2$ recommended for grouped queries (Feng et al., 2025); *SnapKV with PyramidKV* decreases layer budgets arithmetically with depth over the 16 attention layers (Cai et al., 2025). SnapKV variants use the question-free observation window of the main comparison, and every state is compiled once per paper. The arms were fixed before any evaluation read. The rules are far from uniform: under KVzip's global rule, on average 30 of the 64 KV heads keep fewer than 64 document rows and 5 keep none, and PyramidKV's per-head budgets fall from 818 document rows in the first attention layer to 84 in the last. Scoring by reconstruction takes 13.4 s per paper on average (at most 45.4 s), and each allocation rule adds 0.01–0.05 s on average.

Adaptive budgets help KVzip modestly in number recall. Against uniform heads with the same repository scoring, its own global rule adds 6.8 points of constrained recall [2.8, 11.0], though it loses 6.3 F1 [2.1, 10.4], and adaptive heads add 18.9 points of greedy recall [14.9, 23.3]; against our uniform KVzip the gains are $+4.3$ [0.2, 8.6] and $+18.0$ [13.5, 22.7]. They do not help SnapKV (Ada-KV: 0.0; PyramidKV: -4.4 [-6.5, -2.5]). Every adaptive variant stays below AM in constrained

State (average 512 rows per KV head)	Greedy	Constr.	QA F1	Constr. − NCC	
Full context	99.7	99.9	45.9		
NCC	**76.4**	**79.8**	**44.2**		
Attention Matching	16.7	62.3	38.1	−17.5	[−20.5, −14.7]
KVzip, uniform heads	9.4	52.2	18.2	−27.7	[−32.0, −23.3]
KVzip, uniform heads, repository scoring	8.5	49.7	21.4	−30.2	[−34.0, −26.3]
KVzip, global layer and head budgets	12.0	56.5	15.2	−23.3	[−27.1, −19.6]
KVzip, adaptive heads per layer	27.4	52.6	19.3	−27.3	[−31.4, −23.1]
SnapKV, uniform heads	11.3	36.8	21.8	−43.0	[−47.1, −38.9]
SnapKV + Ada-KV	11.4	36.8	21.5	−43.0	[−47.3, −38.8]
SnapKV + PyramidKV	9.6	32.4	19.1	−47.4	[−52.4, −42.5]
SnapKV + PyramidKV + Ada-KV	9.8	33.1	19.2	−46.7	[−51.4, −42.0]

Table 29: **Training-free selections with adaptive budgets** (Qwen3.6-27B, 64 papers, paper-averaged first reads). Every state stores the same number of K/V rows as NCC and keeps the full-context recurrent state. Last column: paired difference from NCC in constrained recall with 95% paper-level bootstrap intervals (20,000 resamples).

recall and QA F1, and 23–47 points of constrained recall and 23–29 F1 below NCC, with every interval excluding zero. Post hoc, NCC is higher in constrained recall on at least 56 of the 64 papers against each of them, and in greedy recall on all 64, and adaptive budgets leave SnapKV's recency bias in place: with Ada-KV its greedy recall is 3.5% on numbers in the first quarter of a paper and 26.2% in the last (Table 28), and NCC is above every adaptive variant in every quarter under both decoders.

I Long-document experiment

This appendix describes the long documents of Section 5.3, the states compared on them, and their complete scores and costs.

Documents and questions. The 64 held-out papers are sorted lexicographically by identifier. This fixed order is split into 8 documents of 8 consecutive papers and, separately, into 4 documents of 16 consecutive papers. Each paper keeps its exact text (Appendix Q.1), which begins with its title. A header line `[Paper k of n]` precedes each paper, and a blank line separates consecutive papers. Including the 61 template tokens, the cached input has 34.1K–50.8K tokens (8-paper documents) or 76.2K–97.2K tokens (16-paper documents; Table 33). All 916 number-recall and 234 QA questions of the main experiments are reused. Each question is read against the document that contains its paper, so each set of documents covers all 64 papers. Number-recall questions already state the paper title. Each QA question is prefixed with `Paper title:` followed by the title stored with the question and a blank line, because a bare question can be ambiguous among 8 or 16 papers. Single-paper and long-document QA scores are therefore not a matched comparison. Every state reads every question once, with the decoders of Appendix O.5.

Statistics. Scores are averaged within each paper and then over the 64 papers, as in Appendix O.6. All papers in a document share one compact state. Paired intervals for the 8-paper documents therefore resample whole documents (document-cluster percentile bootstrap over 8 clusters, 20,000 resamples). Four clusters are too few for a percentile interval, so intervals for the 16-paper documents resample the 64 papers, as in the main experiments; these intervals ignore correlation among papers that share a document. On the 8-paper documents, paper-level intervals would give the same answer to whether zero is excluded for every contrast in Table 31 except one: the QA F1 gain of NCC from 512 to 2048 rows has the paper-level interval $[-0.1, 8.7]$.

States. Each document is prefilled once to give the full-context state, the reference. Every compact state has a uniform budget of $R \in \{512, 2048\}$ rows per KV head in each of the 16 attention layers. These rows are the 61 template rows and $R - 61$ document rows (451 or 1,987). Each compact state also keeps the complete recurrent and convolutional state M_D, C_D of the whole document and the original positions, so a question is read at the position after the full document.

Table 30: **Long-document results for all states.** Paper-averaged number recall (%) with greedy and number-constrained decoding, QA F1, and the number of the 234 QA questions answered "Unanswerable" (Unans.), on 8-paper documents (34.1K–50.8K tokens) and 16-paper documents (76.2K–97.2K tokens). Every state reads all 916 number-recall and 234 QA questions. **Bold**: best compact state per column at each budget.

	8-paper documents				16-paper documents			
State	Greedy	Constr.	QA F1	Unans.	Greedy	Constr.	QA F1	Unans.
Full context	99.6	99.6	46.2	55	99.8	99.9	44.1	57
Recurrent state only	1.3	23.2	8.7	234	0.9	21.4	8.7	234
512 rows per KV head								
StreamingLLM	2.6	23.1	9.3	232	1.6	22.3	8.7	234
SnapKV	3.3	23.7	11.5	225	1.9	22.3	10.4	230
KVzip	0.6	21.6	8.7	234	1.0	20.5	8.7	234
Attention Matching	0.1	26.3	9.0	232	0.0	25.8	8.7	234
NCC	**37.8**	**37.9**	**34.2**	55	**32.5**	**33.0**	**21.8**	97
2048 rows per KV head								
StreamingLLM	7.3	29.1	11.0	225	3.9	26.0	10.3	231
SnapKV	4.8	28.8	14.0	212	4.7	25.9	10.9	226
KVzip	0.9	32.0	8.7	231	0.1	20.9	8.7	233
Attention Matching	9.3	41.1	22.7	170	2.5	32.7	11.5	227
NCC	**56.0**	**57.6**	**38.6**	36	**43.9**	**44.0**	**34.0**	29

- *Attention Matching (AM)* applies Appendix O.2 to the whole document. It uses the same seven fit sequences, at most 8,334 reference positions per query head, 50,000 reference vectors per KV head and on-policy replay, and it is fitted separately at each budget. Three implementation changes leave the arithmetic unchanged. On-policy query replays run in chunks of 1,024 tokens that continue the cache, because the verbatim-repetition sequence alone is as long as the document. The fitting step builds its FP32 score matrix and value targets in chunks of query rows; a CPU check confirmed identical selected keys, β and values. Finally, the diagnostic sequences, which the fit does not use, are not generated.
- *NCC* refines the AM state of the same budget with the recipe of Appendix O.3, unchanged: 128 spans that cover every document token exactly once, cues of at most 32 preceding tokens, and 256 updates with the same optimizer and learning rate. Spans therefore hold 266–397 target tokens in 8-paper documents and 594–759 in 16-paper documents, compared with 18–72 in single papers, so each update averages its loss over more positions. The update count and learning rate were not retuned. The FP32 full-vocabulary teacher targets (31.5–47.0 GiB per 8-paper document and 70.4–89.8 GiB per 16-paper document) are kept in host memory and serve both budgets. The hard-label variant, which matched KL on single papers (Table 3), needs no teacher pass and stores only the source token IDs.
- *StreamingLLM* keeps the template rows, whose leading system-prompt and delimiter tokens act as attention sinks, and the last $R-61$ document tokens. Unlike the single-paper version (Appendix H), it keeps no additional leading document tokens.
- *SnapKV* and *KVzip* are built as in Appendix H. SnapKV takes its observation window from the last 64 document tokens during the full prefill (pooling kernel 7). KVzip scores each token by the maximum attention it receives during chunked reconstruction of the document (chunks of at most 2,048 tokens). Neither uses a question.
- *Recurrent state only* keeps M_D, C_D and the 61 template rows, with no document rows.

The training-free states keep the original post-RoPE keys and values and use no mass correction.

Results. Table 30 lists every state, and Table 31 gives the paired differences. At both budgets and on both document lengths, NCC scores above AM and above every training-free state in greedy recall, constrained recall and QA F1, and every 95% interval lies above zero. NCC also has higher constrained recall than AM on each of the 12 documents at both budgets (Table 32). At 512 rows, AM is 3.1 (8-paper) and 4.4 (16-paper) points of constrained recall above the recurrent state alone. Most compact states rarely attempt a QA answer. The recurrent state alone answers "Unanswerable"

Table 31: **Paired differences on long documents** in percentage points (NCC minus the row at the same budget; the last two rows of each block compare a method at 2048 and 512 rows). 95% percentile bootstrap intervals with 20,000 resamples of the 8 documents (8-paper documents) or of the 64 papers (16-paper documents).

NCC minus	Rows	Greedy	Constrained	QA F1
8-paper documents (intervals over documents)				
Attention Matching	512	+37.7 [33.7, 41.5]	+11.6 [8.6, 14.4]	+25.2 [20.4, 30.0]
	2048	+46.7 [41.6, 52.7]	+16.5 [13.4, 19.6]	+15.8 [9.1, 22.4]
StreamingLLM	512	+35.2 [31.6, 38.8]	+14.8 [11.6, 18.0]	+24.9 [19.4, 30.0]
	2048	+48.7 [43.3, 55.7]	+28.5 [24.6, 33.8]	+27.6 [21.6, 32.4]
SnapKV	512	+34.4 [31.5, 37.4]	+14.2 [11.4, 17.0]	+22.6 [18.2, 26.8]
	2048	+51.1 [46.0, 57.8]	+28.8 [25.4, 32.2]	+24.6 [18.1, 30.9]
KVzip	512	+37.1 [33.5, 40.7]	+16.3 [13.3, 19.3]	+25.5 [20.5, 30.2]
	2048	+55.1 [49.5, 62.0]	+25.6 [20.5, 30.4]	+29.9 [23.0, 35.7]
Recurrent state only	512	+36.4 [33.2, 39.8]	+14.7 [11.8, 17.5]	+25.5 [20.5, 30.2]
	2048	+54.6 [49.8, 61.3]	+34.4 [30.2, 38.8]	+29.9 [23.0, 35.7]
Full context	512	−61.8 [−66.1, −57.7]	−61.7 [−66.1, −57.4]	−12.0 [−16.1, −7.3]
	2048	−43.7 [−49.3, −36.9]	−42.1 [−47.9, −35.3]	−7.6 [−11.1, −4.3]
NCC, 2048 minus 512 rows		+18.2 [14.5, 22.9]	+19.7 [16.2, 24.1]	+4.4 [0.2, 8.7]
AM, 2048 minus 512 rows		+9.2 [6.4, 12.5]	+14.8 [11.4, 18.3]	+13.8 [7.1, 20.6]
16-paper documents (intervals over papers)				
Attention Matching	512	+32.5 [27.8, 37.6]	+7.3 [4.1, 10.8]	+13.1 [8.9, 17.5]
	2048	+41.5 [36.8, 46.2]	+11.4 [7.8, 15.2]	+22.5 [16.7, 28.3]
StreamingLLM	512	+30.9 [26.6, 35.4]	+10.8 [7.4, 14.4]	+13.1 [8.9, 17.5]
	2048	+40.1 [34.2, 45.5]	+18.0 [13.1, 23.0]	+23.8 [18.0, 29.4]
SnapKV	512	+30.6 [26.2, 35.2]	+10.8 [7.2, 14.6]	+11.4 [7.5, 15.5]
	2048	+39.3 [33.8, 44.5]	+18.2 [13.6, 22.8]	+23.2 [17.4, 28.9]
KVzip	512	+31.5 [27.3, 35.9]	+12.5 [9.0, 16.4]	+13.1 [8.9, 17.5]
	2048	+43.8 [39.1, 48.7]	+23.1 [18.8, 27.5]	+25.3 [19.7, 30.9]
Recurrent state only	512	+31.6 [27.4, 36.1]	+11.7 [8.4, 15.1]	+13.1 [8.9, 17.5]
	2048	+43.0 [38.5, 47.5]	+22.7 [18.8, 26.7]	+25.3 [19.7, 30.9]
Full context	512	−67.3 [−72.0, −62.2]	−66.9 [−71.5, −61.9]	−22.3 [−28.5, −16.1]
	2048	−55.9 [−60.7, −51.0]	−55.9 [−60.7, −50.9]	−10.1 [−15.0, −5.1]
NCC, 2048 minus 512 rows		+11.4 [8.1, 14.8]	+11.0 [7.8, 14.3]	+12.2 [6.9, 17.7]
AM, 2048 minus 512 rows		+2.5 [1.1, 4.2]	+6.9 [4.4, 9.6]	+2.8 [0.6, 5.5]

to all 234 QA questions, which scores 8.7 F1. AM and the training-free states give this answer to 170–234 questions, compared with 29–97 for NCC and 55–57 for the full-context reader. Every compact state remains far below full context in number recall. At 2048 rows, NCC trails full context by 42.1 (8-paper) and 55.9 (16-paper) points of constrained recall.

Size and preparation time. Table 33 lists each document. The compact state has a fixed size, while the full-context state grows by 64 KiB per token. The full state is therefore 20.5–30.2× (8-paper) and 44.9–57.0× (16-paper) the size of the 512-row state, and 10.8–15.9× and 23.7–30.1× the size of the 2048-row state. AM dominates preparation, especially its fit at 2048 rows. Relative to AM at the same budget, NCC adds teacher targets and 256 updates. On average this takes 212 s at 512 rows and 115 s at 2048 rows per 8-paper document, and 292 s and 197 s per 16-paper document. Peak allocated GPU memory is 72.4 GiB for 8-paper documents and 92.4 GiB for 16-paper documents.

Recall by paper position. Constrained recall can also be split by the position of a paper within its document. Full context recalls at least 98.4% at every position. Selection by recency favors the last paper. At 2048 rows, StreamingLLM reaches 60.9% on the last paper of an 8-paper document, compared with 20.6% on the first; SnapKV reaches 52.2%. On 16-paper documents, the last paper gets 51.4% with StreamingLLM (15.6% on the first) and 42.0% with SnapKV. NCC shows no such recency effect. At 2048 rows its mean over the first half of the positions is 61.0%, against 54.1% over the second half (8-paper documents); on 16-paper documents it is 42.2% against 45.9%. For StreamingLLM the corresponding values are 21.8% against 36.4% and 23.6% against 28.4%. Each

Table 32: **Scores per long document.** Paper-averaged constrained number recall (%) and QA F1 of Attention Matching (AM) and NCC at 512 and 2048 rows; Full: full-context constrained recall. Document k holds papers $n(k-1)+1$ to nk of the fixed order, with $n = 8$ or 16.

		Constrained recall				QA F1			
		512 rows		2048 rows		512 rows		2048 rows	
Document	Full	AM	NCC	AM	NCC	AM	NCC	AM	NCC
8-paper documents									
1	100.0	25.9	29.6	32.7	45.3	0.0	23.7	8.3	25.0
2	100.0	30.2	41.2	46.6	56.3	10.4	35.3	25.5	46.5
3	100.0	22.6	31.2	37.2	51.9	15.3	45.9	46.2	46.1
4	100.0	29.3	43.6	53.3	76.8	18.8	35.1	25.9	29.2
5	100.0	22.7	31.2	32.8	53.9	5.0	40.0	19.8	41.6
6	97.7	30.1	44.8	42.5	60.7	7.7	22.6	5.0	37.5
7	99.2	29.5	43.2	43.6	63.3	8.3	30.8	20.2	35.1
8	100.0	20.1	38.2	39.8	52.2	6.2	40.0	30.9	47.6
16-paper documents									
1	100.0	25.4	30.6	31.0	38.6	4.2	15.0	8.9	27.1
2	100.0	26.1	32.1	34.1	46.8	17.0	33.6	17.0	39.9
3	99.6	26.0	34.5	33.9	46.0	6.3	18.6	9.7	31.0
4	100.0	25.6	35.0	31.7	44.8	7.3	20.2	10.4	38.2

Table 33: **State size and preparation time per long document**, on one GH200 GPU with the model resident. Tokens: document prefix including the 61 template tokens. Full: full-context state in MiB (64 KiB of keys and values per token plus 75.75 MiB of recurrent and convolutional state). Total: full state divided by the compact state (107.8 MiB at 512 rows, 204.0 MiB at 2048 rows, including β); KV: full KV rows divided by compact rows. AM (s): fit-sequence generation and query extraction (Fit seq., shared by both budgets), then fitting and on-policy replay at each budget. NCC (s): teacher targets (shared), then 256 updates at each budget; the first 512-row update includes a one-time kernel warm-up of 91–115 s. Not shown: full-context prefill (45–69 s) and KVzip scoring (31–148 s).

			512 rows (×)		2048 rows (×)		AM (s)			NCC (s)		
Document	Tokens	Full	Total	KV	Total	KV	Fit seq.	512	2048	Targets	512	2048
8-paper documents												
1	49,412	3,164.0	29.3	96.5	15.5	24.1	937	346	2,872	28	180	93
2	34,128	2,208.8	20.5	66.7	10.8	16.7	755	269	2,677	21	185	83
3	39,521	2,545.8	23.6	77.2	12.5	19.3	852	297	2,758	23	189	88
4	36,707	2,369.9	22.0	71.7	11.6	17.9	895	282	2,726	23	192	88
5	46,407	2,976.2	27.6	90.6	14.6	22.7	860	330	2,844	25	191	90
6	50,819	3,251.9	30.2	99.3	15.9	24.8	897	351	2,887	34	176	93
7	41,159	2,648.2	24.6	80.4	13.0	20.1	872	302	2,771	26	193	86
8	47,042	3,015.9	28.0	91.9	14.8	23.0	882	336	2,853	29	178	89
16-paper documents												
1	83,503	5,294.7	49.1	163.1	26.0	40.8	1,498	648	3,398	64	226	134
2	76,191	4,837.7	44.9	148.8	23.7	37.2	1,243	572	3,300	64	234	124
3	97,189	6,150.1	57.0	189.8	30.1	47.5	1,344	764	3,576	72	223	133
4	88,164	5,586.0	51.8	172.2	27.4	43.0	1,488	651	3,438	71	215	125

position holds different papers (8 per position in 8-paper documents and 4 in 16-paper documents), so these splits confound position with the difficulty of individual papers.

J SELF-STUDY ON LONG DOCUMENTS

This appendix adds self-study to the long-document comparison of Appendix I and reports the coverage test of Figure 7. Documents, questions, reader, scoring and statistics are those of Appendix I, and the predictions of every other state are reused unchanged.

Recipe. Self-study follows Appendix O.4 with its arithmetic unchanged. Each document gets one bank of 256 generated questions. The questions cycle through eight windows of up to 2,048 tokens spread evenly over the document, and the full-context state of the whole document answers them. The bank serves both budgets. Starting from the AM state of Appendix I at 512 or 2048 rows, which is also where NCC starts, the student takes 256 updates, one conversation per update, with the objective, trainable rows, optimizer and learning rate of NCC. One detail of self-study's question-generation prompt had to be ported. Its prompt identifies the source by a `Source identification:` line that repeats the document's first line, which for a single paper is the title. A long document begins with the header `[Paper 1 of` n`]`, so each window's prompt instead names the title line of the paper that contains the window's first token. Teacher answers average 31.6–45.9 tokens per document, so a bank supervises 8.1K–11.7K answer tokens, whereas the 256 updates of NCC supervise every document token twice. As in Appendix I, the FP32 full-vocabulary teacher targets are held in host memory. The exposure-matched checkpoint of Appendix O.4 was not run on long documents.

Results. Table 34 lists the scores and Table 35 the paired differences. Self-study is a strong baseline at long length: it exceeds AM on every metric at both budgets and on both document lengths, and every interval excludes zero. At 2048 rows NCC exceeds self-study in greedy and constrained recall, by $+6.8$ and $+7.5$ points on 8-paper documents and by $+4.7$ and $+4.6$ points on 16-paper documents, with every interval above zero. At 512 rows the recall differences are $+1.4$ to $+2.2$ points, and every interval includes zero. On QA, NCC is ahead only on 8-paper documents at 512 rows ($+8.1$ [4.8, 11.9]); the other three QA differences lie between -1.5 and $+3.0$ points, with intervals that include zero. For comparison, on single papers self-study reaches 77.4% constrained recall, 65.1% greedy recall and 43.4 QA F1, against 79.8%, 76.4% and 44.2 for NCC.

Coverage test. The definitions and predictions below were fixed before any self-study output existed. The eight windows cover 38.7% of the tokens of an 8-paper document (32.3–48.1%) and 19.2% of a 16-paper document (16.9–21.5%). In the 16-paper documents, 25 of the 64 papers receive no window. A number-recall question is *inside* if its masked sentence lies entirely within one window and *outside* if it overlaps none; window position is thus a positional proxy for coverage (Appendix T.3). The 14 and 2 questions whose sentence crosses a window boundary are left out of the contrasts. Recall is averaged over the questions of each group, and intervals resample the 64 papers. We predicted that (P1) NCC would match or exceed self-study in paper-averaged constrained recall at each budget and document length, (P2) self-study's gain over AM would be larger inside the windows than outside, and (P3) NCC's lead over self-study would be larger outside than inside. AM on the same questions is the baseline for both difference-in-differences (DiD) contrasts (Angrist & Pischke, 2009). NCC minus AM is the negative control (Lipsitch et al., 2010), because the source continuations of NCC cover every token.

Table 36 gives the results. P1 holds in all four settings, although at 512 rows its intervals include zero (Table 35). P2 and P3 hold in all four settings, and none of their intervals includes zero. Self-study's gain over AM is 12.2–16.5 points inside the windows and 3.0–6.0 points outside, which makes it 7.8–11.5 points smaller outside. Inside the windows self-study matches or exceeds NCC, by 0.6–8.6 points, while outside NCC leads by 4.1–11.7 points. The difference-in-differences for NCC minus self-study is therefore $+9.1$ to $+15.1$ points. NCC minus AM changes by -1.1 to $+4.5$ points between the two groups, and all four intervals include zero. Greedy decoding gives the same pattern: NCC minus self-study, outside minus inside, is $+9.5$, $+11.4$, $+9.4$ and $+14.4$ points, and every interval excludes zero. With intervals over the 8 documents instead of the papers, the 8-paper results keep their signs. The NCC minus self-study intervals become [2.8, 16.4] (512 rows) and [7.1, 17.7] (2048 rows), and the self-study minus AM interval at 512 rows includes zero ([-20.7, 1.2]). The NCC minus AM change at 2048 rows, $+4.5$, then has the interval [0.6, 9.4], so under this clustering NCC gains slightly more over AM outside the windows than inside.

The per-question outcomes agree with the averages. At 2048 rows on 8-paper documents, 33 of the 48 questions that self-study answers correctly and NCC incorrectly under constrained decoding lie

State	8-paper documents			16-paper documents		
	Greedy	Constr.	QA F1	Greedy	Constr.	QA F1
Full context	99.6	99.6	46.2	99.8	99.9	44.1
512 rows per KV head						
Attention Matching	0.1	26.3	9.0	0.0	25.8	8.7
Self-study, 256 updates	35.9	36.5	26.1	30.8	30.9	**23.3**
NCC	**37.8**	**37.9**	**34.2**	**32.5**	**33.0**	21.8
2048 rows per KV head						
Attention Matching	9.3	41.1	22.7	2.5	32.7	11.5
Self-study, 256 updates	49.1	50.1	37.2	39.2	39.4	31.0
NCC	**56.0**	**57.6**	**38.6**	**43.9**	**44.0**	**34.0**

Table 34: **Self-study on long documents.** Paper-averaged number recall (%) with greedy and number-constrained decoding and QA F1 on 8-paper documents (34.1K–50.8K tokens) and 16-paper documents (76.2K–97.2K tokens). Self-study and NCC both start from the AM state of the same budget and take 256 updates. Full-context, AM and NCC scores are those of Table 30. **Bold**: best compact state per column at each budget.

Difference	Rows	Greedy	Constrained	QA F1
8-paper documents (intervals over documents)				
NCC − self-study	512	+1.9 [−1.6, 4.3]	+1.4 [−2.0, 4.0]	+8.1 [4.8, 11.9]
	2048	+6.8 [3.1, 11.2]	+7.5 [4.2, 11.4]	+1.3 [−3.9, 7.4]
Self-study − AM	512	+35.8 [31.7, 39.6]	+10.2 [6.7, 13.2]	+17.1 [9.5, 24.1]
	2048	+39.8 [36.4, 43.5]	+9.0 [5.5, 13.2]	+14.5 [7.3, 21.3]
16-paper documents (intervals over papers)				
NCC − self-study	512	+1.8 [−1.3, 5.2]	+2.2 [−0.8, 5.5]	−1.5 [−7.5, 5.0]
	2048	+4.7 [1.8, 7.8]	+4.6 [1.8, 7.7]	+3.0 [−2.2, 8.2]
Self-study − AM	512	+30.8 [26.6, 35.3]	+5.1 [3.1, 7.2]	+14.6 [7.4, 21.2]
	2048	+36.7 [32.6, 41.1]	+6.7 [3.6, 9.9]	+19.5 [13.1, 25.7]

Table 35: **Paired differences with self-study on long documents** in percentage points at the same budget. 95% percentile bootstrap intervals with 20,000 resamples of the 8 documents (8-paper documents) or of the 64 papers (16-paper documents), as in Table 31.

inside windows, and 83 of the 114 questions with the opposite outcome lie outside. The facts behind some inside-window wins of self-study were stated in its generated conversations. For example, a question written from the third window of one document asks what percentage of the YouTube links shared by gang members relate to hip-hop music, gangster rap and the surrounding culture, and the teacher answers "76.58%". On the matching number-recall question at 2048 rows, self-study answers 76.58 and NCC answers 83.

For QA, a question counts as inside if any window overlaps its paper. Every paper of an 8-paper document overlaps a window, so the contrast exists only for 16-paper documents: 143 questions (39 papers) inside and 91 questions (25 papers) outside. Self-study's QA gain over AM is +19.9 points inside and +5.9 outside at 512 rows, a difference of +14.0 [1.8, 26.8], and +24.0 against +16.3 at 2048 rows, +7.7 [−4.3, 19.7] (Table 36, last block). AM itself scores higher on the papers without a window, so this split also differs in difficulty. A split by the location of the annotated evidence paragraphs has the same sign with wider intervals.

Cost. Self-study and NCC start from the same AM state, so the AM preparation in Table 33 is common to both. On one GH200 GPU with the model resident, self-study at 512 rows adds on average 736 s of question writing, 773 s of teacher answers and targets, and 164 s of updates, or 1,673 s per 8-paper document. NCC adds 27 s of targets and 185 s of updates, 212 s in total and 12.7% of the self-study time. On 16-paper documents the totals are 1,739 s and 292 s (16.8%). Both budgets share the bank. The 2048-row updates add 80 s (8-paper) and 76 s (16-paper) for self-study, against 89 s and 129 s for NCC. Wall-clock times include the warm-up of the first update. One 8-paper document reused part of its teacher generations from an earlier pass and is left out of these averages.

	Self-study		NCC		AM		Outside minus inside		
Rows	In	Out	In	Out	In	Out	NCC − SS	NCC − AM	SS − AM
8-paper documents: windows cover 38.7% of tokens; 320 questions inside, 582 outside									
512	41.9	31.1	38.4	36.8	25.6	25.1	+9.1 [3.8, 14.8]	−1.1 [−7.4, 5.4]	−10.2 [−17.8, −2.9]
2048	57.5	44.3	56.9	56.0	44.1	38.7	+12.3 [5.2, 19.5]	+4.5 [−1.0, 10.2]	−7.8 [−13.8, −2.0]
16-paper documents: windows cover 19.2% of tokens; 139 questions inside, 775 outside									
512	38.1	28.1	33.1	32.3	25.9	25.2	+9.2 [1.8, 19.0]	−0.1 [−7.2, 7.1]	−9.3 [−16.2, −3.6]
2048	46.8	36.9	38.1	43.4	30.2	31.9	+15.1 [7.3, 22.8]	+3.6 [−4.9, 12.0]	−11.5 [−19.5, −3.5]
QA F1 by paper, 16-paper documents: 143 questions inside, 91 outside									
512	24.8	18.0	20.1	21.7	5.0	12.1	+8.5 [−3.8, 20.6]	−5.5 [−14.2, 2.8]	−14.0 [−26.8, −1.8]
2048	31.6	29.5	32.9	33.0	7.6	13.2	+2.3 [−6.5, 11.8]	−5.4 [−15.2, 4.6]	−7.7 [−19.7, 4.3]

Table 36: **Coverage test on long documents.** Question-weighted constrained number recall (%) for questions whose masked sentence lies inside or outside self-study's eight generation windows, and QA F1 for questions whose paper does or does not overlap a window (16-paper documents only). The last six columns give difference-in-differences contrasts in points: (A minus B) outside the windows minus (A minus B) inside, with 95% paper-cluster percentile bootstrap intervals (20,000 resamples). SS: self-study.

Self-study needs much less host memory for its teacher targets: 7.5–10.9 GiB per document, against 31.5–47.0 GiB (8-paper) and 70.4–89.8 GiB (16-paper) for NCC, whose targets cover every document token. Self-study's peak GPU memory was at most 64.5 GiB.

Scope. The inside/outside split is observational. Inside and outside questions differ in position and difficulty; at 2048 rows on 8-paper documents, for example, AM itself recalls 5.4 points more inside than outside. The difference-in-differences contrasts use AM on the same questions as their baseline and NCC minus AM as a negative control, but coverage was not randomized. The labels of the already-scored AM and NCC predictions were visible while the analysis code was being tested, which happened after the definitions were fixed and before any self-study output existed. Each document has one sampled bank, and each question is read once. Both methods use the single-paper settings (256 updates, same learning rate) without retuning for long documents. The lack of coverage outside the windows is a property of the self-study recipe as specified, with eight windows of 2,048 tokens. The long documents concatenate unrelated papers. There are 8 document clusters for 8-paper documents, and the paper-cluster intervals for 16-paper documents ignore correlation among papers that share a document (Appendix I).

K QuALITY

Setup. We use QuALITY v1.0.1 (Pang et al., 2022) (HTML-stripped development split, CC BY 4.0) and sample 64 of its 115 development articles with a fixed seed, keeping all 1,155 of their questions. The articles have 2.5K–8.7K tokens (median 6.7K). Compilation is unchanged from the Qasper experiments: AM at 512 rows per KV head, NCC with 128 source spans and 256 updates, self-study with 256 generated questions and 256 updates, and question-free KVzip; questions and labels never enter compilation. The reader sees the question and the four options and answers with the option letter. The letter-constrained decoder, our primary one, restricts the first answer token to the four letters; we also parse greedy answers, which may be `Unanswerable` because the reader instructions, as for Qasper, allow it. Scores are article-averaged, and intervals are paired bootstrap intervals over the 64 articles (20,000 resamples).

Decision margins. For each question we form the teacher's and each student's distribution over the four letters from the logged first-token logits (a softmax over the four letter logits, as the letter decoder uses) and apply Theorem 1. The teacher is nearly deterministic: its median top-letter probability is 0.994 and its median margin τ is 0.666 nats. Table 39 reports the share of questions on which a student's within-letter KL divergence lies below τ, so that the student provably selects the teacher's letter; no such question changes letter.

State	Letter	Greedy	Abst.	Letter, HARD	MiB
Full context	90.6	89.0	28	87.8	453.5
Recurrent state only	40.2	10.7	916	33.3	79.6
KVzip	70.4	46.9	484	63.6	107.8
AM	84.9	75.8	164	80.2	107.8
Self-study, 256 updates	86.1	85.4	20	82.4	107.8
NCC	85.1	84.7	17	80.7	107.8

Table 37: **QuALITY accuracy** (%, article-averaged; chance 25%). *Abst.*: greedy `Unanswerable` answers out of 1,155. HARD: the benchmark's hard subset.

Contrast	Letter decoder		Greedy	
NCC − AM	+0.2	[−1.6, 2.0]	+8.8	[6.0, 11.8]
NCC − self-study	−1.0	[−2.9, 0.9]	−0.8	[−2.7, 1.1]
NCC − KVzip	+14.7	[11.3, 18.2]	+37.8	[32.5, 43.0]
Full context − NCC	+5.5	[3.6, 7.5]	+4.3	[2.3, 6.4]

Table 38: **Paired differences on QuALITY** (points; 95% article-cluster bootstrap intervals).

State	Certified (%)	Letter = teacher (%)	Median KL (nats)
AM	84.2	89.4	0.008
Self-study	83.9	89.3	0.014
KVzip	64.2	72.2	0.165
NCC	83.9	89.5	0.010

Table 39: **Certified letter decisions on QuALITY** (Theorem 1). *Certified*: within-letter $D_{\mathrm{KL}}(P^G \| Q^G) < \tau(P^G, y)$.

Commitment. On the 164 questions to which AM answers `Unanswerable` under greedy decoding, NCC commits to a letter on 89.6% (self-study 87.8%, full context 84.1%, KVzip 22.6%), and its greedy answer is correct on 64.0% of them (self-study 67.7%, full context 73.2%, AM 0%).

Exploratory strata. We split questions post hoc by their options, with a rule fixed before computing any stratum outcome: 229 whose options contain numbers or dates, 347 whose options contain named entities, and 579 others. With the letter decoder, NCC minus AM is not distinguishable from zero in any stratum (-4.0 [-9.8, 1.2], -0.5 and -0.4), nor on the HARD subset or across QuALITY's annotations of how much context a question needs (Pang et al., 2022); the greedy gain over AM ($+5.5$ to $+10.6$) appears in every stratum. Against self-study, NCC is lower on entity-option questions (-5.0 [-9.6, -1.2]), consistent with self-study's question-generation prompt asking for specific entities; with 16 stratum contrasts we treat this as a hypothesis.

Cost. Per article on one GH200, AM takes 835 s (682 s of which generate its fit sequences), NCC adds 120 s (15 s of teacher targets and 104 s of updates) and self-study adds 1,333 s (1,251 s of which generate its question bank).

L Additional single-paper results

L.1 Paired effects and per-paper results

Table 40 gives the paired effects of NCC against every state on single papers, and Figure 14 shows every paper. Under constrained decoding NCC beats AM on 57 of the 64 papers, ties on 6 and loses on one (greedy: all 64; QA F1: 35 wins, 15 ties, 14 losses). Against self-study, NCC is statistically indistinguishable under constrained decoding and on QA: at 95% confidence it is at most 1.2 recall points and 2.4 F1 below either checkpoint. On document QA, NCC comes within 1.7 F1 of the full-context reader (-1.7 [-4.6, 1.2]). On extractive questions, whose answers are spans of the paper (126 of the 234 questions), NCC leads both self-study checkpoints and AM (45.8 against 43.3, 42.7 and 38.7 F1; Appendix Q.6). With 1,024 updates (Table 3), under a quarter of self-study's refinement time (361 s against 1,635 s: 15 s of targets and 346 s of updates in the ablation run), NCC

reaches 84.1% constrained recall, 6.7 points [3.8, 9.7] above self-study after 256 updates and 6.2 [3.3, 9.2] above its exposure-matched checkpoint. Trained for the same 1,024 updates from the AM tensors on which this NCC trajectory starts (cycling its 256 conversations four times, with the reader and scoring of the ablation run), self-study reaches 78.9% constrained recall, 72.3% greedy recall and 44.3 F1: 5.2 points [2.0, 8.6] and 7.8 points [3.6, 12.4] below NCC at 1,024 updates, with QA F1 statistically indistinguishable (-1.8 [-5.6, 2.0]), and not significantly different from NCC at 256 updates in constrained recall (-0.4 [-3.6, 2.8]) while 4.3 points [0.2, 8.7] below it in greedy recall. Its 256-update checkpoint in the same run reproduces the original self-study (-0.2 [-2.4, 2.0] constrained). Self-study's 1,024 updates took 369 s per paper, on top of the 1,529 s of generating its conversations, which this run reused; the 1,024 updates of NCC took 346 s. Rebuilt from the same inputs in the ablation run, NCC reproduces Table 1 (-0.5 [-2.8, 1.7] points of constrained recall).

NCC minus	Greedy recall		Constrained recall		QA F1	
Recurrent state only	+75.5	[70.8, 80.0]	+57.5	[53.0, 62.1]	+34.2	[28.4, 39.7]
StreamingLLM	+61.8	[57.0, 66.6]	+44.1	[39.4, 48.8]	+21.3	[16.3, 26.3]
SnapKV	+65.0	[60.1, 70.0]	+43.0	[38.9, 47.1]	+22.4	[17.4, 27.4]
KVzip	+66.9	[62.3, 71.4]	+27.7	[23.3, 32.0]	+26.0	[20.7, 31.2]
Attention Matching	+59.7	[55.3, 64.1]	+17.5	[14.7, 20.5]	+6.1	[2.4, 9.7]
Self-study, 256 updates	+11.2	[7.0, 15.7]	+2.4	[−0.5, 5.3]	+0.8	[−2.4, 4.2]
Self-study, exposure-matched	+10.5	[6.0, 15.3]	+1.9	[−1.2, 4.9]	+0.9	[−2.2, 3.9]
Full context	−23.3	[−27.9, −18.8]	−20.1	[−24.3, −16.0]	−1.7	[−4.6, 1.2]

Table 40: **Paired effects of NCC** on single papers, in points (F1 points for QA), with 95% paper-level bootstrap intervals (20,000 resamples over the 64 papers).

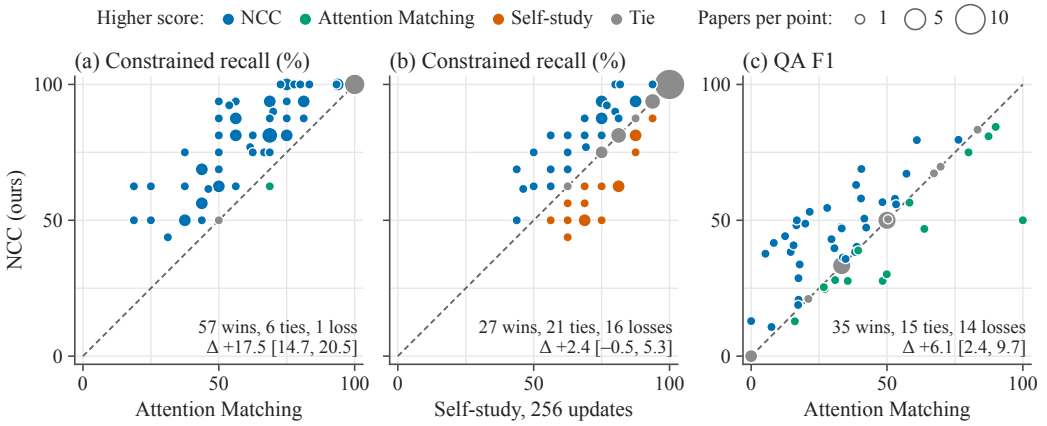


Figure 14: **Per-paper results on the 64 held-out papers.** Each marker is one paper, NCC (y) against a comparator (x); marker area counts papers with identical scores and color names the higher-scoring method. (a) Constrained recall against AM: NCC wins on 57 papers and loses on one. (b) Constrained recall against self-study after 256 updates: NCC is higher on 27 papers, tied on 21 and lower on 16. (c) QA F1 against AM. Appendix V lists every score.

L.2 First-token statistics

Figure 15 gives the distributions behind the first-token statistics of Section 5.4. Relative to self-study, NCC abstains on 150 fewer questions under greedy decoding and answers 101 more correctly; this accounts for most of the drop in its lead from 11.2 points under greedy decoding to 2.4 under constrained decoding (a paired difference of 8.8 points [5.5, 12.6]). For every state that reads the document, each question that constrained decoding repairs was a greedy abstention, and no wrong number becomes correct (Appendix S). A number is a path of token decisions (Corollary 1), and the lead of NCC over AM in constrained recall grows with the length of the teacher's answer: 10.9 points [7.5, 14.4] on numbers of one or two tokens (433 questions), 20.9 [15.6, 26.7] on three or four (335) and 31.1 [23.7, 39.3] on five or more (148). This split is post hoc and question-pooled, and part of the trend is headroom, since AM falls from 78.3% to 21.6% across the three groups.

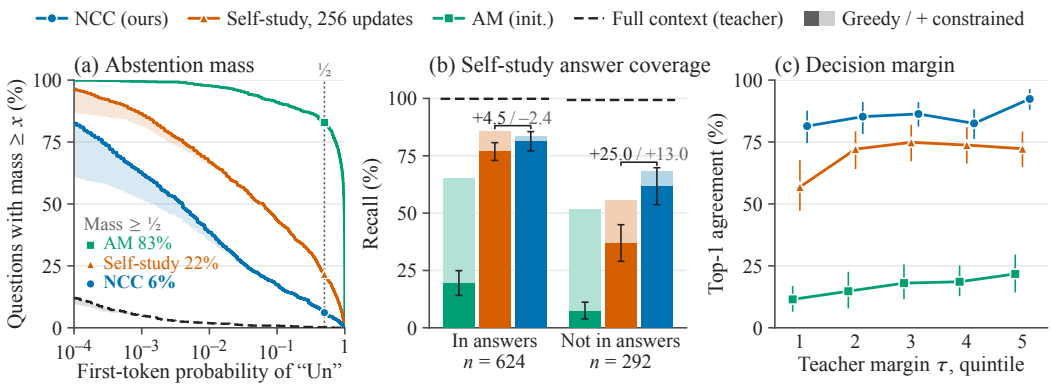


Figure 15: **Three first-token statistics behind the recall gap** (916 number-recall questions, first reads). (a) Share of questions whose first-token probability of "Un" is at least x; this mass lies outside the number tokens G and enters the binary term of Lemma 1. (b) Recall on the 624 numbers that occur in at least one of the paper's 256 self-study teacher answers and on the 292 that occur in none (dark: greedy; light: added by constrained decoding; whiskers: 95% paper-cluster intervals of greedy recall). Brackets: NCC minus self-study, greedy / constrained: $+4.5$ / -2.4 points on covered and $+25.0$ / $+13.0$ on uncovered numbers. (c) Agreement of the student's first token with the teacher's, by quintile of the teacher's decision margin τ (Theorem 1). Shading in (a): bounds for probabilities outside the logged top-5 tokens.

L.3 Where the gains come from

Figure 16 breaks the single-paper results down. NCC is uniformly strong across the paper, at 69.4–78.9% greedy and 75.5–81.2% constrained recall in every quarter, whereas StreamingLLM and SnapKV recover mainly the most recent tokens (StreamingLLM answers 41% of greedy questions about the last quarter of a paper but only 3–9% elsewhere). It is above AM on every role of number (Appendix Q.4) except mathematical constants (92.9 versus 96.4%; no state recovers the three author-markup items), and above AM on extractive, abstractive and yes/no questions. Its gain over AM shows no significant trend with paper length (constrained $r = -0.04$, $p = 0.73$; greedy $r = -0.12$, $p = 0.35$).

L.4 Preparation cost

For NCC, supervision costs 128 teacher-forced passes over 20.3K input tokens, 5.4K of which are source targets (18 s per paper). Self-study decodes 9.5K request tokens and 9.7K teacher-answer tokens one at a time, prefills 556K prompt tokens to write its questions, and still needs 256 teacher-forced passes (34.0K tokens) to extract its targets; generation alone takes 1,529 of its 1,635 refinement seconds. Both methods then train for similar times (108 s and 101 s for 256 updates). The refinement stage of NCC therefore takes 7.7% of the self-study refinement time, and the full pipeline, including the shared AM initialization, takes 45% (Figure 18). All generated supervision, for AM and for self-study, is produced with the same sequential decoding loop; batched decoding in serving engines (Kwon et al., 2023; Zheng et al., 2024) would shrink wall-clock gaps, but not the difference between zero and 19.2K generated tokens per document. The supervision of NCC is fixed by the document and two structural choices (128 spans, 32-token cues); it adds no question count, sampling temperature or answer length.

M Method comparison

Table 41 compares NCC with prior compression and document-memory methods on what they store, how the artifact is obtained and what supervises it. We do not run amortized compressors (Mu et al., 2023; Ge et al., 2024b; Zhang et al., 2025; Chari et al., 2025): none has released compressor weights for these hybrid readers, and training one needs large-scale training of its own. Per-document weight updates (Caccia et al., 2025; Kujanpää et al., 2025; Wang et al., 2026; Yang et al., 2025c; Tan et al., 2024) change the reader itself, whereas a compiled state leaves the model unchanged, so that hundreds of compiled states share one copy of it (Appendix F); Eyuboglu et al. (2026) also find a

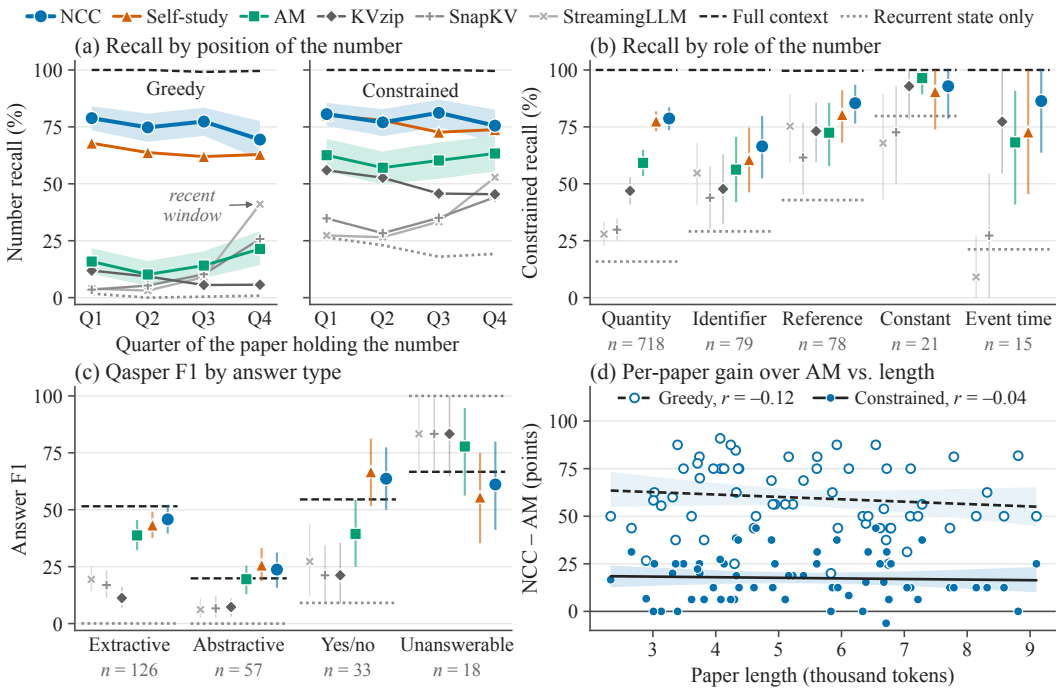


Figure 16: **NCC's gains over AM are not confined to one part of the paper, one kind of number or one paper length.** First reads on the 64 held-out papers; self-study is the 256-update checkpoint. (a) Number recall by the quarter of the paper that holds the masked number (question-weighted; shading: 95% paper-cluster bootstrap intervals for NCC and AM). StreamingLLM and SnapKV succeed mainly in the last quarter. (b) Constrained recall by the role of the masked number (Appendix Q.4). (c) Qasper answer F1 by answer type; unanswerable questions reward abstention, so the recurrent state alone scores 100 on them. (b, c) Whiskers: 95% paper-cluster bootstrap intervals; dashed and dotted segments: full context and recurrent state only. (d) Per-paper gain of NCC over AM against paper length, with least-squares fits and 95% bootstrap bands; there is no significant trend (greedy $r = -0.12$, $p = 0.35$; constrained $r = -0.04$, $p = 0.73$).

trained KV cache better than LoRA. Query-time selection and retrieval (Tang et al., 2024; Chen et al., 2025; Lewis et al., 2020) are complementary: they keep the full cache or the text and pay a prefill or a selection per question, whereas a compiled state is fixed before any question and read without prefill. Other designs give heads different spans (Xiao et al., 2025) or preload whole caches (Chan et al., 2025).

Method	Artifact	Per-doc. optim.	Frozen reader	Supervision	Output objective	Per-doc. generation	Query-agnostic
Selection of original cache rows (KV cache eviction)							
StreamingLLM Xiao et al., 2024	KV subset	✗	✓	position	✗	✗	✓
H2O, SnapKV Zhang et al., 2023; Li et al., 2024	KV subset	✗	✓	attn. scores	✗	✗	✗[a]
KVzip Kim et al., 2025	KV subset	✗	✓	attn. scores	✗	✗	✓
Per-document fitting at the attention level							
Attention Matching (AM) Zweiger et al., 2026	new KV	✓	✓	attn. outputs	✗	optional[b]	✓
KVSculpt Jiang & Jin, 2026	new KV	✓	✓	attn. outputs	✗	✗	✓
Amortized compressors (one shared model for all documents)							
Gist Mu et al., 2023	new KV	✗	✗	instructions	CE	✗	✓
AutoCompressors Chevalier et al., 2023	soft tokens	✗	✗	text	CE	✗	✓
ICAE Ge et al., 2024b	soft tokens	✗	✓	text, instructions	CE	✗	✓
KV-Distill, Still Chari et al., 2025; O'Neill et al., 2026	new KV	✗	✓	instructions, gen. QA	KL	✗	✓
Doc-to-LoRA Charakorn et al., 2026	LoRA	✗	✗	gen. QA	KL	✗	✓
Per-document distillation at the output level							
Cartridges (self-study) Eyuboglu et al., 2026	new KV	✓	✓	gen. QA	KL	✓	✓
Cartridges at Scale Hardalov et al., 2026	new KV	✓	✓	gen. QA	KL	✓	✓
DCD Caccia et al., 2025	LoRA	✓	✗	doc. text, gen. QA	KL + hidden	optional[c]	✓
LCC Li et al., 2026	new KV	✓	✓	doc. text, instructions	KL	✗[d]	✓
NCC (ours)	new KV	✓	✓	doc. text	KL	AM only[b]	✓

Table 41: **How NCC relates to prior compression and document-memory methods.** *Artifact*: what is stored per document; *KV subset* keeps original cache rows, *new KV* holds rows in the model's native cache format that are not copies of original rows (fitted, trained or produced by an encoder), soft tokens are input embeddings, and LoRA is a weight update. *Per-doc. optim.*: the artifact is fitted to each document, in closed form or by gradients, rather than selected or produced by a shared pretrained compressor. *Frozen reader*: the model that reads the artifact keeps its original weights. *Supervision*: the signal that the artifact, or the shared compressor, is fitted to. *Output objective*: the loss is on next-token outputs rather than on attention (CE: hard-label cross-entropy; KL: divergence from a teacher that reads the uncompressed context; hidden: matching the teacher's hidden states). *Per-doc. generation*: building the artifact for a new document requires sampling text from a model; amortized compressors may still be trained on generated data (e.g., Still, Doc-to-LoRA). *Query-agnostic* (as in Kim et al., 2025): the artifact is built before any question and reused for all of them. DCD: Deep Context Distillation; LCC: Latent Context Compilation. Also placed in Figure 4: PyramidKV (Cai et al., 2025) and Expected Attention (Devoto et al., 2025) (attention scores), Activation Beacon (Zhang et al., 2025) and RestoreKV (Baek et al., 2026) (amortized compressors), and prompt distillation (Kujanpää et al., 2025) (context distillation on generated data). [a]Both score tokens with attention from a prompt that contains the question: H2O accumulates it over the prompt and the generated tokens, and SnapKV reads it from an observation window at the end of the prompt. Question-free variants are possible, as in our SnapKV baseline. [b]AM can fit on reference queries from a prefill of the context, or of its repetition, alone; its self-study query source (their term; the generated text only supplies reference queries and is not distilled) samples text, as does the recipe we use (seven model-generated sequences per paper, 9.7K tokens on average). NCC inherits these sequences from its AM initialization; its refinement stage samples nothing, and with repeat-prefill AM no stage does (Appendix D). [c]Document DCD distills on the document's own chunks; Synthetic DCD uses generated QA pairs or summaries. [d]LCC generates no document-specific questions; its regularizer pairs the context with generic Alpaca instructions (Taori et al., 2023).

N BUDGET SWEEP

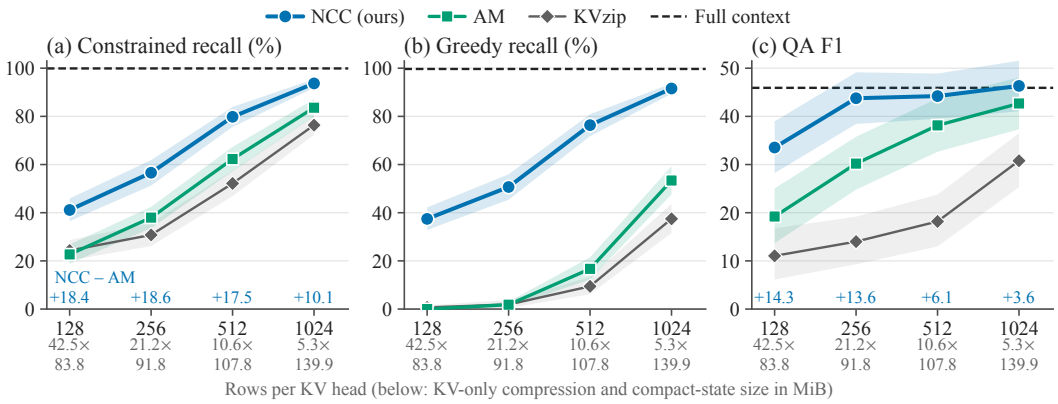


Figure 17: **NCC improves on AM at every row budget.** Constrained recall (a), greedy recall (b) and QA F1 (c) of NCC, AM and KVzip at 128–1024 rows per KV head on the 64 held-out papers. Shading: 95% bootstrap intervals; dashed: full context. Blue numbers: paired NCC−AM differences, all with intervals above zero (Table 43). Below each axis: mean KV-only compression and compact-state size.

Setup. We repeat AM, NCC and KVzip at $R \in \{128, 256, 1024\}$ rows per KV head in each of the 16 attention layers. As in the main comparison, every budget includes the 61 template rows, so 67, 195 and 963 rows are left for the document. The 512-row states are those of Table 1 and Appendix H. The evaluation covers all 64 held-out papers, all 916 number-recall and 234 QA questions, and first reads only, with the reader and decoders of the main comparison.

- *AM* (Appendix O.2) is fitted separately at each budget. It uses the same fit sequences, the same selected reference-query positions (identical to the main run for all 64 papers) and 50,000 reference vectors per KV head.
- *NCC* refines the AM state of the same budget with the recipe of Appendix O.3, unchanged: 128 spans, cues of at most 32 tokens and 256 updates with the same optimizer and learning rate. The teacher targets are computed once per paper and shared by all budgets.
- *KVzip* (Appendix H) scores each paper once and keeps the $R - 61$ highest-scoring document rows of every KV head together with the template rows. It uses this uniform per-head budget, not KVzip's non-uniform allocation across heads.

Before the sweep, one paper (1606.00189) was rerun at 512 rows. The AM refit matched the main run on 64 of 64 heads, and the reads of AM, NCC and KVzip matched the earlier reads on 35 of 35 answers each. No new read raised an error. At most 7 of the 234 QA reads per state and one number-recall read (AM at 128 rows, constrained decoding) reached the generation limit (Appendix O.5).

Results. Table 42 lists every state and Table 43 the paired differences. NCC scores above AM at every budget on all three metrics, and every 95% interval lies above zero. For QA at 1024 rows the lower bound is 0.1. The difference in constrained recall is 18.4, 18.6, 17.5 and 10.1 points from 128 to 1024 rows, and the QA difference is 14.3, 13.6, 6.1 and 3.6 F1, so both are smaller at 1024 rows than at the smaller budgets. Under greedy decoding, AM answers "Unanswerable" to all 916 number-recall questions at 128 rows and to 897 at 256 rows (NCC: 150 and 167), so its greedy recall is 0.0% and 1.8% and the greedy differences are larger (37.4 to 59.7 points). Across budgets, NCC with R rows has a higher QA F1 than AM with $2R$ rows at each of the three pairs, but the paired interval excludes zero only at 256 against 512 rows. In constrained recall, NCC with R rows is above AM with $2R$ rows at 128 against 256 rows and below it at the two larger pairs. KVzip is never above AM with an interval that excludes zero. From 256 rows on it is below AM in constrained recall, and it is below AM in QA at every budget.

State size. The compact state always contains the complete 75.75 MiB recurrent and convolutional state, which dominates at small budgets. On these papers (2.4K–9.2K tokens), KV-only compression

			Accuracy			State size		
Rows	Doc.	State	Greedy	Constr.	QA	MiB	Total	KV
all	all	Full context	99.7	99.9	45.9	415.6	1	1
128	67	KVzip	0.7	24.3	11.0	83.8	4.96	42.5
		AM	0.0	22.7	19.2			
		NCC	**37.4**	**41.1**	**33.5**			
256	195	KVzip	1.9	30.8	14.0	91.8	4.53	21.2
		AM	1.8	38.0	30.2			
		NCC	**50.7**	**56.6**	**43.8**			
512	451	KVzip	9.4	52.2	18.2	107.8	3.85	10.6
		AM	16.7	62.3	38.1			
		NCC	**76.4**	**79.8**	**44.2**			
1024	963	KVzip	37.5	76.4	30.8	139.9	2.97	5.3
		AM	53.4	83.6	42.7			
		NCC	**91.6**	**93.7**	**46.3**			

Table 42: **Accuracy and state size across row budgets.** Rows: rows per KV head in each attention layer, including the 61 template rows; Doc.: rows left for the document. Greedy and Constr.: paper-averaged number recall (%) under greedy and number-constrained decoding; QA: Qasper F1. MiB: compact state, i.e. keys, values and β plus the 75.75 MiB recurrent and convolutional state (for full context, the mean over papers). Total: full state divided by compact state; KV: full KV rows divided by compact rows; both are means over papers. **Bold**: best compact state per column at each budget.

	Greedy recall			Constrained recall			QA F1		
Rows	Diff.	95% CI	+/−	Diff.	95% CI	+/−	Diff.	95% CI	+/−
NCC − AM, same budget									
128	+37.4	[32.8, 42.2]	63/0	+18.4	[14.5, 22.8]	54/3	+14.3	[10.8, 18.0]	51/5
256	+48.9	[44.0, 53.7]	64/0	+18.6	[15.1, 22.3]	56/2	+13.6	[10.3, 17.0]	47/7
512	+59.7	[55.3, 64.1]	64/0	+17.5	[14.7, 20.5]	57/1	+6.1	[2.4, 9.7]	35/14
1024	+38.2	[33.0, 43.4]	60/1	+10.1	[7.7, 12.6]	41/1	+3.6	[0.1, 7.3]	31/17
KVzip − AM, same budget									
128	+0.7	[0.0, 1.6]	3/0	+1.6	[−0.5, 3.8]	26/15	−8.2	[−12.5, −3.9]	3/33
256	+0.1	[−1.8, 2.1]	6/10	−7.2	[−10.8, −3.7]	13/38	−16.2	[−20.1, −12.4]	2/52
512	−7.2	[−11.2, −3.3]	11/35	−10.1	[−14.6, −5.7]	18/42	−19.9	[−24.4, −15.7]	2/55
1024	−15.9	[−21.5, −10.1]	11/46	−7.2	[−10.0, −4.5]	8/36	−11.9	[−15.8, −7.8]	10/44
NCC at R rows − AM at $2R$ rows									
128 vs. 256	+35.6	[31.1, 40.3]	63/0	+3.2	[0.4, 6.0]	24/14	+3.3	[−0.4, 6.9]	32/22
256 vs. 512	+34.0	[29.6, 38.3]	61/0	−5.7	[−8.7, −2.8]	16/35	+5.6	[2.7, 8.7]	34/19
512 vs. 1024	+23.0	[18.5, 27.6]	56/3	−3.8	[−6.1, −1.4]	9/28	+1.5	[−1.3, 4.3]	26/24

Table 43: **Paired differences across row budgets** in percentage points (F1 points for QA), with 95% paper-level percentile bootstrap intervals (20,000 resamples of the 64 papers) and the number of papers with a positive/negative difference. The last block compares NCC with AM at twice its budget.

averages 42.5× at 128 rows (18.7–71.6× across papers) and 5.3× at 1024 rows (2.3–8.9×). At the same two budgets, total compression averages 4.96× (2.69–7.74×) and 2.97× (1.61–4.64×).

Cost. Times are per paper on one GH200 GPU with the model resident, as means over the 64 papers unless stated otherwise. The AM pipeline takes 39 s at 128 rows, 62 s at 256 rows and 603 s at 1024 rows, most of it in the fitting step. The teacher targets for NCC take 15 s and serve all budgets. The 256 updates take a median of 76–79 s at each of the three budgets; the mean at 128 rows, 102 s, includes a one-time kernel warm-up of about 100 s in each GPU process. KVzip scoring takes 1.3 s, full-context prefill 14.6 s and one read of a state (35 answers) 11–15 s. Peak allocated GPU memory is 67.4 GiB, during AM fitting at 1024 rows. Costs at 512 rows are reported in Appendix P.

O Implementation details

O.1 Model and state

We use Qwen3.6-27B (Qwen Team, 2026b) in BF16 and evaluation mode (Transformers 5.14.0.dev0, flash-linear-attention 0.5.1, tilelang 0.1.12) with thinking disabled in the chat template. The model has 16 full-attention layers with four KV heads of dimension 256 and 48 Gated DeltaNet layers (Yang et al., 2025b). Reading n tokens (the document and 61 template tokens) produces n KV rows per head in every attention layer, together with the fixed-size recurrent and convolutional states M_D, C_D.

The compact state of Equation (1) keeps 512 rows per KV head in every attention layer. The tokenizer's character offsets identify the rows that lie wholly inside the document: 451 rows per KV head in every paper. Only the keys and values of these document rows are trained; the other 61 rows hold template tokens (system prompt, document delimiters and question marker) and stay fixed. Compact rows keep their original positions, so a question is read at the position that follows the full document rather than at position 512. Every read starts from a fresh copy of the artifact, and the state then evolves with the question and the generated tokens. Training uses PyTorch's reference (math) implementation of scaled dot-product attention (SDPA); reading uses the model's default SDPA kernel.

O.2 Attention Matching initialization

We follow the query-agnostic recipe of Zweiger et al. (2026). After reading a document, the model generates seven fit sequences (the model-generated sequences of Section 2) that cover verbatim repetition, summarization, fact aggregation, structured JSON output and answers to self-generated questions. Teacher-forced passes over these sequences supply reference queries: at most 8,334 positions per query head and 50,000 reference query vectors per KV head, drawn from its group of six query heads, with the same indices reused when the queries of later layers are re-extracted on-policy, that is, after the earlier layers are compacted, as in AM. AM then fits compact keys, values and per-row mass corrections β under a uniform budget of 512 rows per KV head, with the OMP selection and fitting code released with AM, unmodified. AM also allows non-uniform head budgets. We give AM, NCC, self-study and the main training-free baselines the same uniform width, so that they store equal resident bytes in the native cache, and NCC refines whichever AM state it starts from. On two development papers, a non-uniform allocation fitted for Qwen3.6-27B lowered AM's loss on reference continuations, but padded to a common width it stored 3.9–4.1× more K/V bytes (69.75 against 17.81 MiB and 89.75 against 22.06 MiB at a nominal 5% budget). The pipeline also generates a few diagnostic sequences that the fit does not use; their time is included in the AM initialization (Table 44).

O.3 Source-continuation distillation

Spans and cues. For a document of L_D tokens, the nominal span boundaries are $\lfloor bL_D/128 \rfloor$ for $b = 1, \ldots, 127$. Each boundary moves to the nearest position at which the token sequence can be split reversibly (decoding and re-encoding each side gives the same token IDs; the lower one on a tie), keeping at least one token per span. The 128 target spans are the original token IDs and concatenate to the full document; they contain 18 to 72 tokens across all papers. Every span after the first carries a cue of at most 32 preceding source tokens, quoted as exact text and shortened only to a reversible boundary. Target IDs are never re-tokenized from text. For two papers that together

contain three canonically composable accented characters, boundaries are computed on the Unicode NFC-normalized text after checking that it yields identical token IDs.

Requests and targets. The first request asks the model to repeat the document verbatim from its beginning. Every later request supplies the cue and asks the model to continue the document immediately after it, without repeating the cue. For span b, teacher and student both read the request followed by $t_{b,<t}$ and are compared at every target position. One teacher-forced pass per span, from a copy of the full-context state, gives full-vocabulary targets stored in FP32. No end-of-sequence token is appended at span boundaries.

Optimization. Each update uses one span and averages Equation (2) over its target positions, so positions in shorter spans receive more weight. Two epochs over the 128 spans give 256 updates. We use AdamW with learning rate 0.01, moment coefficients $(0.9, 0.999)$, $\epsilon = 10^{-8}$, no weight decay and global gradient-norm clipping at 1, with FP32 master copies of the trainable rows. The backbone, β, the template rows, all positions and M_D, C_D are frozen, and the refined cache is exported in BF16.

O.4 Self-study control

Self-study generates 256 factual questions from eight document windows. Window $i \in \{0, \dots, 7\}$ starts at token $\lfloor i \max(0, L_D - 2048)/7 \rfloor$ and contains up to 2,048 tokens. Questions cycle through the windows and are sampled with temperature 0.7 and top-p (nucleus) sampling at 0.95 (Holtzman et al., 2020), up to 96 tokens. The full-context teacher answers each question greedily, up to 96 tokens and with an instruction to use at most 60 words. A teacher-forced pass over each answer, including its end-of-sequence token when one was generated, gives full-vocabulary targets. Duplicate questions and truncated answers stay in the bank, and no answer is filtered. Initialization, trainable rows, objective, optimizer and reader are those of NCC.

The 256-update checkpoint visits each of the 256 conversations once. The exposure-matched checkpoint continues cyclically until it matches NCC in three measures of student exposure. For an update sequence $\mathcal{S}$ in which conversation b has n_b student input tokens (question and answer history) and m_b target tokens, we track

$$B(\mathcal{S}) = \left(\sum_{b \in \mathcal{S}} n_b, \quad \sum_{b \in \mathcal{S}} m_b, \quad \sum_{b \in \mathcal{S}} \left[512 n_b + \frac{n_b(n_b+1)}{2} \right] \right), \tag{5}$$

where the last term counts query–key pairs against the 512 compact rows and the causal history. Training stops at the first update at which every component is at least that of the 256 NCC updates, with a cap of 1,024 updates. The resulting checkpoints use 256 to 466 updates (mean 329.8); for two papers they coincide with the 256-update checkpoint, and both checkpoints are still read separately. The rule matches exposure, not FLOPs or wall-clock time.

O.5 Decoding

Greedy decoding selects the most probable token over the full vocabulary. Number-constrained decoding uses the same logits but permits only signed integers and decimals, including correctly grouped thousands separators and a leading decimal point, and allows the end-of-sequence token only after a complete number; it excludes alphabetic answers such as `Unanswerable`, fractions and exponent notation. Both decoders stop at 32 tokens for number recall; QA uses greedy decoding with a 128-token limit. Every question is read twice with each state and decoder, and all scores use the first read.

O.6 Statistics

All scores are paper-averaged: question scores are averaged within each paper, and the 64 paper means are averaged with equal weight. Paired 95% intervals are percentile bootstrap intervals (Efron, 1979) from 20,000 resamples of the 64 papers (seed 2026092183). Each resample keeps all states and questions of a paper together, so the intervals account for correlation within papers; they are pointwise, without adjustment for multiple comparisons. For question-weighted quantities this resampling is a cluster bootstrap with papers as clusters, which we call paper-cluster intervals. Win, tie and loss counts compare per-paper scores. Cost ratios divide the sum of per-paper preparation times of NCC by that of the comparator and use the same bootstrap.

P PREPARATION COST AND MEMORY ACCOUNTING

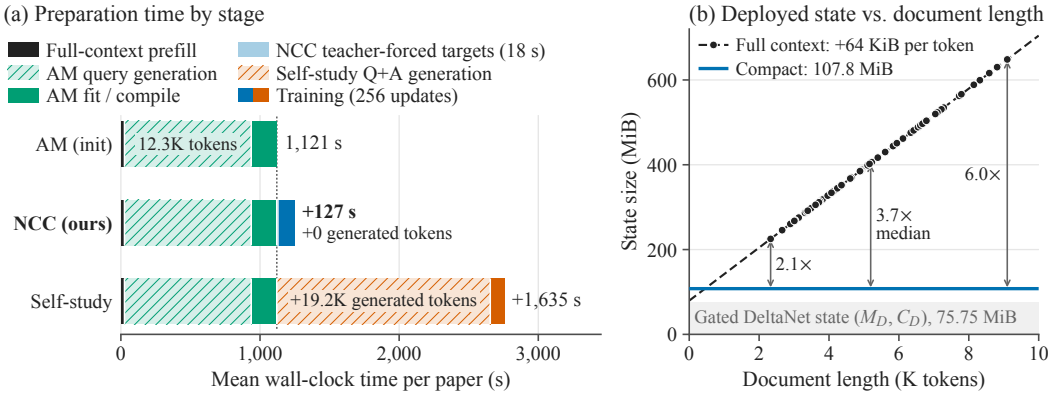


Figure 18: **Preparation cost and memory.** (a) Mean wall-clock time per paper by stage. All methods share the AM initialization, which itself generates 12.3K tokens (9.7K in fit sequences, the rest in unused diagnostic sequences); self-study then spends most of its refinement generating 19.2K tokens of questions and answers, whereas NCC replaces generation with teacher-forced passes over the source. (b) Full-context state of each paper against the fixed 107.8 MiB compact state (2.1–6.0× smaller, median 3.7×).

P.1 PREPARATION TIME AND TOKEN COUNTS

Each paper is prepared on one GH200 GPU. The NCC and self-study refinements of a paper run back to back on the same GPU, in alternating order across papers (32 papers each way). The shared initialization consists of the full-context prefill and the AM pipeline. The refinement stage of each method covers loading its inputs, building its supervision, optimization and exporting the checkpoint; computing span boundaries is charged to NCC and to the exposure-matched checkpoint, whose stopping rule uses the NCC exposure totals $B(\mathcal{S})$. Table 44 lists the stages together with token counts that do not depend on the implementation.

All generated supervision, namely the AM fit and diagnostic sequences and the self-study questions and answers, is produced by the same sequential token-by-token decoding loop in Transformers at batch size one, so wall-clock ratios reflect this implementation while the token counts do not depend on it. Decoding dominates both pipelines that generate text: the AM fit sequences take 729.8 of the 1,120.6 initialization seconds, and self-study questions and answers take 1,528.7 of its 1,635.0 refinement seconds. Summed over papers, the refinement stage of NCC takes 7.7% [7.1, 8.5] of the self-study time at 256 updates and 7.6% [7.0, 8.4] at the exposure-matched checkpoint; with the shared initialization included, the ratios are 45.3% [44.4, 46.2] and 44.9% [44.0, 45.8].

P.2 STATE SIZE

Per cached token, the full-context KV cache stores keys and values for 16 layers, four KV heads and 256 dimensions in BF16, that is $2 \times 16 \times 4 \times 256 \times 2$ bytes $= 64$ KiB. The 48 Gated DeltaNet layers hold 75.75 MiB of recurrent and convolutional state regardless of length, and both the full-context and the compact state keep it. The compact state therefore takes $32 + 0.0625 + 75.75 = 107.8125$ MiB for every paper, while the full-context state takes $64\,\text{KiB} \cdot n + 75.75$ MiB (Table 45). All compression comes from the KV cache, and it grows linearly with document length.

Q TASK CONSTRUCTION AND CONTENT ANALYSIS

Q.1 DOCUMENTS

We use the public Qasper v0.3 training release (Dasigi et al., 2021). A paper is eligible if it has title, abstract and section/paragraph fields, 12,000–40,000 characters of constructed text and at least two official questions. The document concatenates the title, the abstract, every section heading and paragraph, and all textual figure and table captions in their original order, separated by blank lines;

			Self-study	
	AM init.	NCC	256 updates	Exp.-matched
Wall-clock time (s)				
Full-context prefill and export[a]	25.9			
AM fit-sequence generation	729.8			
AM diagnostic-sequence generation[b]	180.9			
AM reference-query extraction and budget allocation	10.4			
AM fitting with on-policy queries	168.6			
Other AM steps	4.9			
Setup and span boundaries		0.6	0.1	0.6
Question generation		–	750.6	750.6
Teacher targets[c]		17.8	778.0	778.0
Other data preparation		–	5.3	5.3
Optimization and export		108.2	100.9	124.0
Total for the stage	1,120.6	126.7	1,635.0	1,658.6
Total including AM init.	1,120.6	1,247.3	2,755.6	2,779.2
Token and update counts				
Generated tokens	12,271	0	19,176	19,176
Prompt tokens for generation[d]	1,031	0	580,352	580,352
Teacher-forced passes	7	128	256	256
Teacher-forced tokens	13,332	20,294	34,024	34,024
Optimizer updates	–	256	256	329.8
Student input tokens	–	40,589	34,024	43,681
Student target tokens	–	10,752	9,709	12,346

[a]Includes first-call kernel compilation and warm-up on 58 papers; on the other six, the prefill takes 0.3–1.0 s. [b]Generated by the AM pipeline but not used by the fit. [c]NCC: one teacher-forced pass per source span. Self-study: greedy teacher answers followed by one teacher-forced pass per answer. [d]Tokens prefilled before decoding, excluding the cached document; each self-study question prompt contains a 2,048-token document window.

Table 44: **Preparation cost per paper**, as means over the 64 papers on one GH200 GPU. The AM initialization is shared by every compact state; the other columns list the refinement stage that each method adds to it. Stage times may not add up to the totals because of rounding. Counts exclude document prefills (about 5.4K tokens each).

Component	Full context	Compact (NCC)
KV rows per head and layer	n	512
KV cache (MiB)	149.4–572.6 (328.3)	32
Mass corrections β (MiB)	–	0.0625
Recurrent and convolutional state M_D, C_D (MiB)	75.75	75.75
Total (MiB)	225.1–648.4 (404.1)	107.8125
Compression of the full state		2.09–6.01× (3.75×)
Compression of the KV cache		4.67–17.89× (10.26×)

Table 45: **State size per paper.** n is the number of cached tokens (2,390–9,162 on our papers). Ranges are minimum–maximum over the 64 papers, with the median in parentheses.

figure and table contents that are not available as text are omitted. We sort the eligible identifiers, shuffle them with Python's random seed 20260919123, and take the first 12 papers for development and the next 64 for evaluation, with no shared identifier or identical text between the two sets. Evaluation documents have 2,329–9,101 tokens: 18 have fewer than 4,096, 42 have 4,096–8,191 and four have 8,317–9,101. Compilation reads only the document text. Qasper is public, so the model may have seen some of these papers during pretraining; the question-only reader (Appendix W) measures how much of the task the model can answer without the document.

						Self-study	
Content role	n_q	n_D	Full	AM	NCC	256 upd.	Exp.-matched
Quantity	718	64	100.0	59.2	78.7	77.5	78.3
Event time	15	11	100.0	68.2	86.4	72.7	72.7
Mathematical constant	21	14	100.0	96.4	92.9	90.5	92.9
Identifier	79	35	100.0	56.2	66.5	60.6	64.0
Reference or structure	78	24	99.7	72.4	85.4	80.4	76.9
Author markup	3	1	100.0	0.0	0.0	0.0	0.0
Uncertain	2	2	100.0	50.0	100.0	50.0	50.0

Table 46: **Number recall by content role**, with number-constrained decoding (%). Scores are averaged within each paper and role, then over the n_D papers that contain the role; n_q counts questions. Rows cover different sets of papers.

Q.2 Number recall

Candidate sentences are split at newlines, or at sentence-final punctuation followed by whitespace and an uppercase character. The rule keeps sentences of 60–500 characters and excludes bibliography sections, rendered equations, Markdown headings, mathematical LaTeX, and scientific-notation, range, ratio, exponent and uncertainty patterns. It also rejects bracketed citations, figure/table/section reference numbers, some identifier fragments, repeated parenthesized enumerations, space-grouped number fragments, and values that would remain visible elsewhere in the question after masking. For $E \geq 2$ eligible numbers in source order, we choose $K = \min(E, 16)$ of them at indices

$$i_k = \left\lfloor \frac{(k + 1/2)E}{K} \right\rfloor, \qquad k = 0, \ldots, K - 1. \tag{6}$$

A sentence can contribute several items. The question gives the paper title and the exact sentence with one number replaced by `[NUMBER]`, asks for Arabic digits in the unit shown, forbids converting percentages to fractions and allows the answer `Unanswerable`. All 64 papers have eligible items, which gives 916 questions. The rule depends only on the source text and does not interact with the spans or cues of NCC.

A number parser reads each answer, and an answer is correct when its canonical numerical value equals the masked value. Abstentions, unparseable answers, answers with several numbers and truncated answers count as incorrect.

Q.3 Document QA

Document QA uses all 234 official Qasper questions for the 64 papers, including unanswerable, ambiguous and figure-dependent ones. Answers are scored with Qasper's lexical answer F1 against the original annotations.

Q.4 Content of the number-recall items

The mechanical rule selects any literal number in a sentence. To see which kinds of numbers the compact states recover, every one of the 916 items was labeled by the role of its masked number, based on the masked sentence and its source context, using an AI-assisted, source-grounded review. The labels change no question, reference or score. *Quantities* are empirical counts, reported results, dimensions, hyperparameters and substantive thresholds, including those in quoted examples or related work. *Event times* date an actual event or data collection. *Mathematical constants* are definitional or conventional values. *Identifiers* include dataset editions, software versions, named configurations and grant numbers. *References or structure* covers bibliographic years, table and figure labels and enumeration markers. *Author markup* covers leftover annotation or formatting parameters, and two items have an *uncertain* role.

Quantities make up 718 of the 916 items and occur in all 64 papers (Table 46). On them, NCC recalls 78.7% against 59.2% for AM, and self-study reaches 77.5% and 78.3%, so the gain over AM is not driven by identifiers or formatting. NCC also leads on event times, identifiers and references, while AM is slightly ahead on the 21 mathematical constants. No compact state recovers the three

	Greedy ↑	Constrained ↑
Full context	99.8	100.0
Attention Matching	12.7	51.4
Self-study, 256 updates	61.7	71.8
Self-study, exposure-matched	62.4	72.4
NCC (source continuations)	**71.1**	**74.4**
NCC minus Attention Matching	+58.4 [53.2, 63.6]	+23.0 [18.6, 27.3]
NCC minus self-study, 256 updates	+9.4 [4.9, 14.2]	+2.6 [−0.9, 6.0]
NCC minus self-study, exposure-matched	+8.8 [3.8, 14.0]	+2.0 [−1.9, 5.8]

Table 47: **Number recall on the 636 items that the question-only reader answers incorrectly** (paper-averaged over 64 papers, %). Paired differences with 95% bootstrap intervals over papers.

author-markup items, which come from one paper. Because the rule is mechanical, a few items are degenerate: twelve items in one paper mask the same table cutoff (20), and one masks only the `000` of a number printed as `20, 000`. All items are kept, and paper-level averaging limits the weight of any single paper.

Q.5 Items the question alone does not answer

The question shows the paper title and the full sentence around the masked number, so some numbers can be guessed from the sentence or from prior knowledge. The question-only reader (Appendix W) answers 280 of the 916 items correctly under number-constrained decoding. Table 47 restricts the comparison to the other 636 items, which span all 64 papers. On these items, NCC recalls 74.4% against 51.4% for AM, a paired gain of 23.0 points [18.6, 27.3] that holds on 53 papers, with 10 ties and one loss. NCC remains level with self-study under constrained decoding and ahead of it under greedy decoding, as on the full set. On the 280 items that the question-only reader answers (62 papers), all compact states score between 89.2% (AM) and 94.8% (NCC).

Q.6 Document QA by answer type

Table 48 splits the 234 Qasper questions by the answer type of their references, as annotated in Qasper (Dasigi et al., 2021). Scores are question-weighted answer F1 of first reads; the table gives point estimates, and Figure 16(c) shows 95% paper-cluster bootstrap intervals for AM, self-study (256 updates) and NCC. On extractive questions, the largest group, NCC is closest to the full-context reader among compact states; self-study is stronger on abstractive and yes/no questions, and AM is highest on unanswerable questions because it often abstains.

Answer type	n	Full context	AM	Self-study 256	Self-study exp.	NCC
Extractive	126	51.5	38.7	43.3	42.7	**45.8**
Abstractive	57	19.9	19.5	25.7	**26.8**	23.7
Yes/no	33	54.5	39.4	**66.7**	63.6	63.6
Unanswerable	18	66.7	**77.8**	55.6	55.6	61.1

Table 48: **Qasper answer F1 by answer type** (question-weighted, first reads).

R Development experiments

The recipe was selected on the 12 development papers (Appendix Q.1); the 64 evaluation papers were never used to select it. The comparisons below use ten of the development papers: nine of them have eligible number-recall items (135 questions) and all ten have Qasper questions (38). All scores are paper-averaged. The two comparisons were read in separate rounds, so states that appear in both, such as AM, score slightly differently.

Learning rate. In an earlier development round on the same 12 papers, refining a compact cache by continued training on factual self-study data scored higher with learning rate 0.01 than with 0.001 on both QA (31.7 against 30.3 F1) and number recall (46.7% against 35.8%), and we used 0.01 for every refinement thereafter.

State	Number recall ↑ Greedy	Constr.	QA F1 ↑
Full context	100.0	100.0	37.0
Attention Matching	19.9	60.2	36.4
Self-study, 256 updates	77.2	81.3	40.2
Self-study, exposure-matched	76.3	81.1	41.7
AM-sequence reuse, 7 updates	29.0	65.1	34.7
AM-sequence reuse, 256 updates	52.7	76.1	36.2
NCC (source continuations), 256 updates	**79.3**	**82.2**	**46.0**

Table 49: **Development comparison of supervision sources** on ten development papers (nine for number recall, 135 questions; ten for QA, 38 questions). Paper-averaged scores (%). All refined states start from the same AM cache and refine keys and values.

Which coordinates to refine. Starting from the same AM cache and the same source-continuation supervision, we compared refining only a shared offset of the document-row mass corrections β (one scalar per layer and KV head), only the values, and the keys and values jointly. Refining the mass offset left constrained recall at the AM level (60.8% against 60.9%). Refining the values raised it to 75.0% and QA F1 from 36.4 to 41.9, while joint K/V refinement reached 81.6% and 46.0. The same ordering held for self-study supervision (79.2% with values only, 81.3% with keys and values). Under value-only refinement, source continuations trailed self-study on number recall; with joint refinement the two were level, and we adopted joint K/V refinement.

Reusing the AM fit sequences. AM already generates seven sequences per paper (Appendix O.2). We tested whether distilling on them could replace new supervision: we extract full-vocabulary teacher targets on all seven sequences and apply the objective of Equation (2) for one pass (7 updates) and for 256 updates. Table 49 shows that reuse improves constrained recall over AM (65.1% and 76.1% against 60.2%) but stays below source continuations (82.2%), and that it does not improve QA F1. Because the fit sequences are long, 256 reuse updates also process 4.8M student input tokens over the ten papers, against 0.41M for 256 source-continuation updates. We therefore kept source continuations as the supervision.

S Decoder outcomes

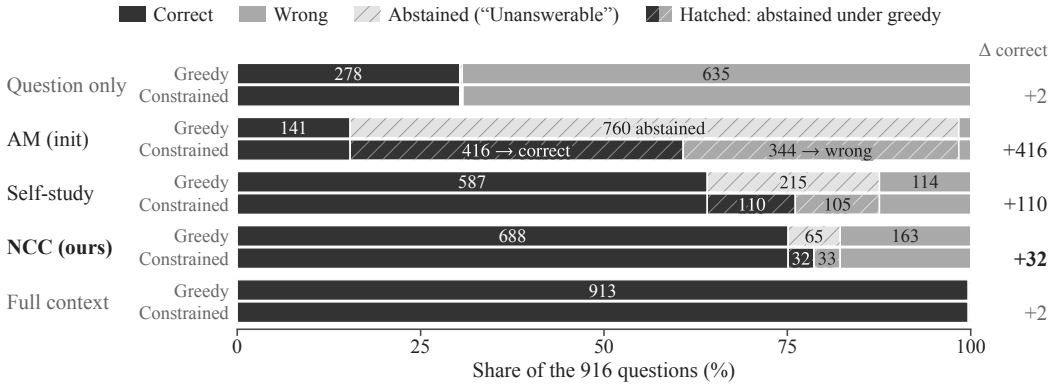


Figure 19: **Outcomes of the 916 number-recall questions** under greedy and number-constrained decoding (self-study: 256 updates; *wrong* includes unparseable answers). For every state that reads the document, each answer repaired by constrained decoding was a greedy abstention (hatched), and no greedy answer is lost. NCC rarely abstains, so it gains least from constrained decoding.

T Theory: proofs and supporting analyses

All logarithms are natural. $\Delta^\circ(\mathcal{V})$ denotes the strictly positive distributions on the finite vocabulary $\mathcal{V}$, with $|\mathcal{V}| \geq 2$. For a nonempty proper subset $G \subset \mathcal{V}$, $P(G) = \sum_{i \in G} P_i$, $P_i^G = P_i / P(G)$ for

Result	Claim	Assumptions	Status
Proposition 1	mass agreement set is null in $\mathbb{R}^d$ and on ellipsoids; selection never agrees	n distinct full keys, $m < n$ rows; $d \geq 2$ for ellipsoids	written
Proposition 2	importance-weighting and Donsker–Varadhan transfer; coverage is necessary	finite $\mathcal{H}$, fixed artifact	written; (i) for point masses in Lean
Lemma 1	chain rule under a permitted set	$P, Q \in \Delta^\circ(\mathcal{V})$	written (standard)
Theorem 1	KL below τ keeps the winner; τ is sharp	strict teacher winner	(i) and the minimum in (ii) Lean; runner-up reduction, strict-overtake infimum and (iii) written
Corollary 1	identical greedy or constrained answers	bound at every teacher prefix	greedy induction in Lean; constrained case written
Proposition 3	soft versus hard targets	P strictly positive, $\lvert\mathcal{V}\rvert \geq 2$	written
Lemma 2	exponential sums have few zeros	distinct rates	written (classical)

Table 50: **Theoretical results at a glance.** "Lean" marks statements machine-checked in Lean 4 (Appendix T.10); "written" marks pen-and-paper proofs in this appendix; every result was also checked numerically (Appendix T.9).

$i \in G$, and $d(p\|q) = p\log(p/q) + (1-p)\log[(1-p)/(1-q)]$ is the binary KL divergence. P_h and Q_h abbreviate $P_D(\cdot \mid h)$ and $Q_A(\cdot \mid h)$ from Section 3. Table 50 lists the results, their assumptions and how each was checked.

T.1 Compaction is lossy (Proposition 1)

Lemma 2 (Zeros of exponential sums)**.** *Let $\lambda_1 < \cdots < \lambda_R$ be real and let $c_1, \ldots, c_R$ be real and not all zero. Then $g(t) = \sum_{r=1}^{R} c_r e^{\lambda_r t}$ has at most $R-1$ real zeros.*

Proof. Induction on R. Dropping zero coefficients lowers R, so assume every $c_r \neq 0$. For $R = 1$, g has no zero. Let $R \geq 2$ and suppose g had R distinct zeros. Then so does $\tilde{g}(t) = e^{-\lambda_1 t} g(t) = c_1 + \sum_{r \geq 2} c_r e^{(\lambda_r - \lambda_1)t}$, and by Rolle's theorem $\tilde{g}'(t) = \sum_{r \geq 2} c_r(\lambda_r - \lambda_1) e^{(\lambda_r - \lambda_1)t}$ has at least $R-1$ distinct zeros. But $\tilde{g}'$ is an exponential sum with $R-1$ distinct rates and nonzero coefficients, which by induction has at most $R-2$ zeros, a contradiction. □

The lemma is a weak form of Descartes' rule of signs for exponential sums (Pólya & Szegő, 1998). Restricted to lines, it also shows that exponentials $e^{\langle q, x_r \rangle}$ with distinct $x_r \in \mathbb{R}^d$ are linearly independent on every nonempty open set.

Proof of Proposition 1. Write the full rows as $(k_i, b_i, v_i)_{i \leq n}$ and the compact rows as $(c_j, \beta_j, \tilde{v}_j)_{j \leq m}$, with masses $Z(q) = \sum_i e^{\langle q, k_i \rangle + b_i}$ and $Z'(q) = \sum_j e^{\langle q, c_j \rangle + \beta_j}$ and weighted value sums $N(q) = \sum_i e^{\langle q, k_i \rangle + b_i} v_i$ and $N'(q) = \sum_j e^{\langle q, c_j \rangle + \beta_j} \tilde{v}_j$.

Merging. Let $x_1, \ldots, x_R$ ($R \leq n + m$) be the distinct vectors among $k_1, \ldots, k_n, c_1, \ldots, c_m$. Then

$$g(q) = Z(q) - Z'(q) = \sum_{r=1}^{R} a_r e^{\langle q, x_r \rangle}, \qquad a_r = \sum_{i:\, k_i = x_r} e^{b_i} - \sum_{j:\, c_j = x_r} e^{\beta_j}. \tag{7}$$

Coinciding compact keys are merged here. The compact keys take at most m distinct values, and each equals at most one k_i because the full keys are distinct. At least $n - m \geq 1$ full keys therefore equal no compact key, and for such a key $x_r = k_i$ we have $a_r = e^{b_i} > 0$. Hence $\mathcal{R} = \{r : a_r \neq 0\}$ is nonempty.

Lines. For distinct $r, s \in \mathcal{R}$, $x_r - x_s \neq 0$, so $H_{rs} = \{u : \langle u, x_r - x_s \rangle = 0\}$ is a hyperplane. For $u \notin \bigcup_{r \neq s} H_{rs}$ and any q_0, the function $t \mapsto g(q_0 + tu) = \sum_{r \in \mathcal{R}} \left(a_r e^{\langle q_0, x_r \rangle}\right) e^{t\langle u, x_r \rangle}$ is an

exponential sum with $|\mathcal{R}| \le n + m$ distinct rates and nonzero coefficients. By Lemma 2 it vanishes at most $n + m - 1$ times.

Null set. $E = g^{-1}(0)$ is closed because g is continuous. Fix a unit vector $u \notin \bigcup H_{rs}$ and write $q = q_0 + tu$ with $q_0 \in u^\perp$. By Tonelli's theorem, the Lebesgue measure of E is $\int_{u^\perp} \big|\{t : q_0 + tu \in E\}\big|\, dq_0 = 0$, since every section is finite. A nonempty open set has positive measure, so E has empty interior.

Consequences at $q \notin E$. (i) Let $Z_F(q) > 0$ be the mass of the tokens read after the cache (question and generated tokens), taken to be the same for both caches as in the statement. The share $Z/(Z + Z_F)$ is strictly increasing in Z, so $Z(q) \neq Z'(q)$ gives different shares. (ii) Append rows with total mass $Z_F > 0$ and weighted value sum N_F. The head outputs $(N + N_F)/(Z + Z_F)$ and $(N' + N_F)/(Z' + Z_F)$ agree if and only if

$$(Z' - Z)\, N_F = N'Z - NZ' + (N' - N)\, Z_F. \tag{8}$$

With the appended keys and biases fixed, Z_F is fixed, and since $Z' \neq Z$ exactly one N_F solves Equation (8). For a single appended row (k', b', w) we have $N_F = Z_F w$, so at most one w gives agreement. If instead $Z' = Z$, Equation (8) reads $(N' - N)(Z + Z_F) = 0$. Hence the outputs agree for every appended row if and only if both $Z' = Z$ and $N' = N$ at q, the two quantities that AM fits.

Ellipsoids. Let $d \ge 2$ and $\mathcal{S} = \{Mv : \|v\| = 1\}$. The map $v \mapsto Mv$ is a diffeomorphism from the unit sphere $\mathbb{S}^{d-1}$ onto $\mathcal{S}$ and preserves null sets, and $g(Mv) = \sum_{r \in \mathcal{R}} a_r e^{\langle v, y_r \rangle} =: \tilde{g}(v)$ with distinct $y_r = M^\top x_r$. It therefore suffices to show that $\tilde{g}$ vanishes on a null subset of $\mathbb{S}^{d-1}$. Fix a unit vector e and write $v = \cos\theta\, e + \sin\theta\, w$ with $\theta \in (0, \pi)$ and w a unit vector in $e^\perp$. The poles $\pm e$ are null, and the surface measure is $\sin^{d-2}\theta\, d\theta\, d\sigma(w)$ with σ the surface measure of the unit sphere of $e^\perp$ (counting measure on two points if $d = 2$). For fixed w, put $\pi_r = (\langle e, y_r \rangle, \langle w, y_r \rangle) \in \mathbb{R}^2$. Then $\gamma_w(\theta) = \tilde{g}(\cos\theta\, e + \sin\theta\, w) = \sum_r a_r \exp\langle(\cos\theta, \sin\theta), \pi_r\rangle$ extends to an entire function of θ. Suppose the π_r are distinct. Then some ϕ gives $\langle(\cos\phi, \sin\phi), \pi_r\rangle$ a unique maximizer r^*. Because $\mathrm{Re}\langle(\cos(\phi + iT), \sin(\phi + iT)), \pi\rangle = \cosh T\, \langle(\cos\phi, \sin\phi), \pi\rangle$, the term r^* dominates $\gamma_w(\phi + iT)$ as $T \to \infty$. Hence $\gamma_w \not\equiv 0$, and it has finitely many zeros in $[0, \pi]$. Two points π_r, π_s coincide only if $\langle e, y_r - y_s \rangle = 0$ and $\langle w, y_r - y_s \rangle = 0$; then $0 \neq y_r - y_s \in e^\perp$. This excludes a σ-null set of w for $d \ge 3$ and no w for $d = 2$. Tonelli's theorem completes the proof.

Selection. If the compact rows are the rows indexed by $S \subsetneq \{1, \dots, n\}$, with their biases, then $g(q) = \sum_{i \notin S} e^{\langle q, k_i \rangle + b_i} > 0$ for every q. □

Scope. (a) *Sharpness.* With $m \ge n$, copying the cache, and splitting one row's weight e^{b_i} over the extra rows, gives $E = \mathbb{R}^d$ and identical outputs. With $m < n$, E can be nonempty and even infinite: a single row whose bias matches the mass at one query q_0 agrees on the whole zero set of $Z - Z'$ through q_0, generically a hypersurface. The line bound $n + m - 1$ is attained: full keys ± 1 against one compact key 0 of weight 3 agree where $2\cosh t = 3$, twice. In the numerical check of Appendix T.9, a 6-row cache whose keys and biases are fitted to the mass of a 12-row cache at six reference queries agrees there to machine precision and at none of 200,000 fresh queries. Attention Matching fits the mass at a finite set of reference queries by nonnegative least squares (NNLS) (Zweiger et al., 2026). Its authors note that a subset of rows without biases never receives more mass than the full cache, which the selection clause makes strict, and that biases are needed even at $q = 0$. They also show that matching both the mass and the attention output of the cached rows (AM's block output) at q preserves the head output under any appended rows. The proof of (ii) gives the converse. Proposition 1 shows that biases can buy agreement on a null set of queries but not on any open set. (b) *Repeated keys, positions and normalized queries.* If full keys coincide, merging them (adding their weights e^{b_i}) reduces n to the number of distinct keys. A compact row with bias $-\infty$, which a zero NNLS weight produces, is simply absent. Keys are stored after rotary position embedding (RoPE; Su et al., 2024), and the query of a token at position p is a rotated, scaled projection. Our reader also RMS-normalizes each query head (QK-norm, as in Yang et al., 2025a) before a partial rotary embedding, so the queries at a fixed position lie on ellipsoids $\{sR_p\Gamma v : \|v\| = c\}$, with s the attention scale, R_p the rotation, Γ the diagonal norm gain and c just below $\sqrt{d}$. These ellipsoids are Lebesgue-null in $\mathbb{R}^d$, so the null-set clause alone would not exclude E containing them; the ellipsoid clause does, provided no gain is zero. Agreement has probability zero under any query distribution with a density on $\mathbb{R}^d$ or on such an ellipsoid. A reader realizes only finitely many queries, and a null

set can contain all of them: the proposition excludes agreement on every open set of queries and on every such ellipsoid, not at a given finite set, where exact fits are possible (see (a)). The clause needs $d \geq 2$: for $d = 1$ the "sphere" $\{\pm 1\}$ is two points, and a 2-row cache with keys $\pm\frac{1}{2}$ matches a 3-row cache with keys $-1, 0, 1$ at both. (c) *Head versus model.* The statement concerns each head separately. Later layers can compensate, so it does not imply that the model's next-token distribution changes; that is one reason to measure fidelity at the output, as Proposition 2 and Theorem 1 do. (d) *No rate.* The proposition bounds no error size. In a small numerical example, compact caches optimized on a ball of queries reach a mean squared relative mass error of 4.8×10^{-5} with 2 of 12 rows: clustered keys can be summarized well, and the objective decides where.

T.2 Transfer by change of measure (Proposition 2)

Proof of Proposition 2. (i) Since $\ell \geq 0$ and $\operatorname{supp}\mu \subseteq \operatorname{supp}\nu$, $\mathbb{E}_\mu \ell = \sum_{h:\,\nu(h)>0} \nu(h)\frac{\mu(h)}{\nu(h)}\ell(h) \leq \rho\,\mathbb{E}_\nu \ell$. For μ a point mass at h, $\rho = 1/\nu(h)$.

(ii) If $\operatorname{supp}\mu \not\subseteq \operatorname{supp}\nu$, then $D_{\mathrm{KL}}(\mu\|\nu) = \infty$ and there is nothing to prove. Otherwise, let $Z_\lambda = \mathbb{E}_\nu e^{\lambda\ell} \in [1, \infty)$, finite because $\mathcal{H}$ is finite, and define the tilted distribution $\nu_\lambda(h) = \nu(h)e^{\lambda\ell(h)}/Z_\lambda$, which has the support of ν. For $h \in \operatorname{supp}\mu$, $\log[\mu(h)/\nu_\lambda(h)] = \log[\mu(h)/\nu(h)] - \lambda\ell(h) + \log Z_\lambda$. Multiplying by $\mu(h)$ and summing over the finite support,

$$0 \leq D_{\mathrm{KL}}(\mu\|\nu_\lambda) = D_{\mathrm{KL}}(\mu\|\nu) - \lambda\,\mathbb{E}_\mu \ell + \log Z_\lambda, \tag{9}$$

which rearranges to the bound. Gibbs' inequality is an equality exactly when $\mu = \nu_\lambda$. This is the Donsker–Varadhan variational formula in finite form (Donsker & Varadhan, 1975; Boucheron et al., 2013). If $0 \leq \ell \leq \bar{\ell}$ on $\operatorname{supp}\nu$, Hoeffding's lemma gives $\log Z_\lambda \leq \lambda\mathbb{E}_\nu \ell + \lambda^2\bar{\ell}^2/8$ (Boucheron et al., 2013), so $\mathbb{E}_\mu \ell \leq \mathbb{E}_\nu \ell + D_{\mathrm{KL}}(\mu\|\nu)/\lambda + \lambda\bar{\ell}^2/8$. Choosing $\lambda = \sqrt{8D_{\mathrm{KL}}(\mu\|\nu)}/\bar{\ell}$ gives the bounded form $\mathbb{E}_\mu \ell \leq \mathbb{E}_\nu \ell + \bar{\ell}\sqrt{D_{\mathrm{KL}}(\mu\|\nu)/2}$; if $D_{\mathrm{KL}}(\mu\|\nu) = 0$, then $\mu = \nu$, and if $\bar{\ell} = 0$, both sides vanish.

(iii) Predictors are unrestricted maps $h \mapsto Q_h \in \Delta^\circ(\mathcal{V})$. Set $Q_h = P_h$ for $h \notin S$, so $L_\nu = 0$ because $\nu(S) = 0$. For $h \in S$, let y_h be the teacher's greedy token (a maximizer of P_h), pick $j \neq y_h$ and $\epsilon \in (0, 1)$, and set $Q_h = (1 - \epsilon)e_j + \epsilon\,\mathbf{1}/|\mathcal{V}|$. Applying Equation (3) with $G = \{y_h\}$ and dropping the nonnegative within-set terms, $\ell(h) \geq d\big(P_h(y_h)\|\epsilon/|\mathcal{V}|\big) \geq P_h(y_h)\log\big(P_h(y_h)|\mathcal{V}|/\epsilon\big) - e^{-1}$, which diverges as $\epsilon \downarrow 0$. Hence $\mathbb{E}_\mu \ell \geq \mu(S)\min_{h\in S}\ell(h)$ exceeds any B for small ϵ. The student's greedy token at $h \in S$ is $j \neq y_h$, its unique maximizer. If h is the first prefix of the teacher's greedy answer that lies in S, the student agrees with the teacher before h, because $Q = P$ there and both apply the same fixed tie-breaking rule (in our decoder, the lowest token id), and it deviates at h. □

Remarks. (a) The point-mass case of (i) is the covered-prefix bound $\ell(h) \leq L_\nu/\nu(h)$, which is machine-checked, as is a missing-prefix counterexample in the spirit of (iii) (Appendix T.10). (b) Combining (i) with Theorem 1: if μ is supported on teacher prefixes with strict winners and $\tau_{\min}$ is the smallest margin on $\operatorname{supp}\mu$, Markov's inequality gives $\mu\{h : \ell(h) \geq \tau(P_h, y_h)\} \leq \rho\,L_\nu(A)/\tau_{\min}$ for the fraction of deployment decisions not certified by Theorem 1. We call a decision *certified* when $\ell_A(h) < \tau(P_h, y_h)$, so that Theorem 1(i) guarantees it. (c) The proposition concerns one fixed artifact. Losses observed at different optimizer iterates are not L_ν of the final artifact, and no statement is made about optimization. (d) For NCC, $\nu(h_{b,t}) = 1/(128\,L_b)$, where L_b is the number of target tokens of span b, with repeated identical histories receiving their combined weight. For self-study, ν weights the answer positions of the 256 conversations, uniformly within each conversation. Evaluation histories begin with prompts that neither method trains on, so for both methods parts (i) and (ii) give no bound for the evaluation itself ($\rho = \infty$ and $D_{\mathrm{KL}}(\mu\|\nu) = \infty$). The proposition explains why the choice of ν matters, not the test scores.

T.3 Coverage of the self-study supervision

This post hoc analysis asks whether the facts probed by number recall are touched by the self-study supervision. For each paper we collect the 256 teacher answers that both self-study checkpoints train on. A number-recall item is *covered* if its masked value occurs verbatim as an unsigned number (commas removed) in at least one of these answers; the two negative values therefore count as uncovered. Comparing numbers by value instead of as strings (so that 68.0 matches 68) covers 634 items instead of 624 and changes no entry of Table 51 by more than 1.9 points. A second, contextual variant counts an item as covered only if a single self-study conversation (generated request and

	Value in an answer				Value in context			
	Greedy ↑		Constrained ↑		Greedy ↑		Constrained ↑	
State	cov.	unc.	cov.	unc.	cov.	unc.	cov.	unc.
Full context	99.8	99.3	99.8	100.0	100.0	99.3	100.0	99.8
AM	19.2	7.2	65.1	51.7	19.9	10.9	61.4	60.3
Self-study (256)	76.9	36.6	85.7	55.5	79.9	48.3	85.4	66.8
Self-study (exposure)	78.0	36.3	**86.4**	56.2	**80.6**	48.9	**85.6**	67.9
NCC	**81.4**	**61.6**	83.3	**68.5**	80.1	**70.1**	81.7	**75.5**

Table 51: **Number recall by self-study coverage** (first reads, % of questions). "Covered": the masked value occurs in one of the paper's 256 self-study teacher answers (624 of 916 items). "Value in context": one self-study conversation (request and answer) states the value and shares a word 4-gram with the masked sentence (458 of 916).

		Greedy			Constrained		
Coverage	Self-study	cov.	unc.	unc. − cov.	cov.	unc.	unc. − cov.
Value in an answer	256 updates	+4.5	+25.0	+20.5	−2.4	+13.0	+15.4
		[0.8, 8.3]	[17.0, 33.5]	[12.5, 28.7]	[−5.4, 0.6]	[7.4, 18.6]	[9.6, 21.0]
	exposure	+3.4	+25.3	+22.0	−3.0	+12.3	+15.4
		[−1.0, 7.7]	[16.6, 34.8]	[13.2, 31.3]	[−6.6, 0.3]	[7.4, 17.5]	[10.1, 20.5]
Value in context	256 updates	+0.2	+21.8	+21.6	−3.7	+8.7	+12.4
		[−4.1, 4.6]	[15.9, 28.3]	[15.0, 28.4]	[−7.1, −0.4]	[4.2, 13.2]	[7.1, 17.9]
	exposure	−0.4	+21.2	+21.6	−3.9	+7.6	+11.6
		[−5.3, 4.4]	[14.6, 28.4]	[14.3, 29.3]	[−7.9, 0.0]	[3.5, 11.7]	[6.5, 16.7]

Table 52: **NCC minus self-study within each coverage stratum** (percentage points, first reads). The last column of each group is the uncovered-minus-covered difference of the two contrasts. Brackets: 95% percentile intervals from resampling the 64 papers (20,000 resamples), with question-level means within each stratum.

teacher answer) both states the value and shares a word 4-gram with the item's masked sentence; it splits the items evenly (458 of 916). Coverage is lexical: an answer can mention the number in another role, and a fact can be supervised without its number. Table 51 reports first-read accuracy by stratum, pooled over questions, and Table 52 the paired contrasts. For contrasts we resample the 64 papers with replacement (20,000 resamples), recompute the question-level stratum means and report percentile intervals. Many papers have few uncovered items (median 4 per paper), which is why we pool questions instead of averaging per paper as in the main results.

The gap between NCC and self-study lies almost entirely on uncovered items: the difference of the two within-stratum contrasts is 20.5 points [12.5, 28.7] under greedy decoding and 15.4 [9.6, 21.0] under constrained decoding, and 11.6–22.0 points for every variant and checkpoint, with all intervals above zero. Under constrained decoding NCC trails self-study slightly on covered items (2.4–3.9 points; of the four intervals, only that for 256 updates under the contextual variant lies entirely below zero). Self-study's own accuracy falls by about 40 greedy points from covered to uncovered items, against 20 for NCC. Coverage was not randomized, however: covered items are easier for every compact state (AM, which never sees self-study data, gains 12.0 greedy points from uncovered to covered), so the within-state gaps mix coverage with difficulty. The between-method contrasts within a stratum are the informative quantity. They are consistent with Proposition 2, but not implied by it, since neither method trains on the evaluation questions.

T.4 Chain rule for KL divergence (Lemma 1)

Proof of Lemma 1. Strict positivity gives $p, q \in (0, 1)$, so all conditionals are defined. On G,

$$\sum_{i \in G} P_i \log \frac{P_i}{Q_i} = \sum_{i \in G} p P_i^G \left(\log \frac{p}{q} + \log \frac{P_i^G}{Q_i^G} \right) = p \log \frac{p}{q} + p \, D_{\mathrm{KL}}(P^G \| Q^G). \tag{10}$$

The same computation on G^c gives $(1-p)\log[(1-p)/(1-q)] + (1-p)D_{\mathrm{KL}}(P^{G^c}\|Q^{G^c})$, and adding the two gives Equation (3). This is the chain rule for relative entropy with the binary partition as the first variable (Cover & Thomas, 2006). All three terms are nonnegative, which gives $p\, D_{\mathrm{KL}}(P^G\|Q^G) \le D_{\mathrm{KL}}(P\|Q)$. For Q with $Q^G = P^G$ and $Q^{G^c} = P^{G^c}$, Equation (3) reduces to $d(p\|q)$, which diverges as $q \downarrow 0$. □

With G the tokens that continue a well-formed number, the constrained decoder reads only Q^G and ignores $d(p\|q)$ and the term on G^c; a singleton G fixes the next token with no margin required. Teacher and student use the same G at a shared prefix, and renormalization does not change the order of probabilities within G. Theorem 1 applied to P^G and Q^G therefore certifies the teacher's *constrained* decision, not its unconstrained one. A constrained decoder can remove the abstention option without repairing a wrong ranking among the permitted tokens.

T.5 Sharp decision margin (Theorem 1)

Proof of Theorem 1. Pair average. Fix $j \neq y$ with $a = P_y \ge b = P_j$, and define R by

$$R_y = R_j = \tfrac{a+b}{2}, \qquad R_i = P_i \quad (i \notin \{y, j\}). \tag{11}$$

R is strictly positive and normalized, has $R_j = R_y$, and satisfies $D_{\mathrm{KL}}(P\|R) = \psi(a,b)$ because only two coordinates differ, where $\psi(a,b)$ denotes the right side of Equation (4) for the pair (a,b). Equivalently, $\psi(a,b) = (a+b)\, d\big(\frac{a}{a+b} \,\|\, \frac{1}{2}\big)$, which is Equation (3) with $G = \{y, j\}$.

Minimality on the boundary. For any $Q \in \Delta^\circ(\mathcal{V})$ with $Q_j \ge Q_y$, the inequality $-\log u \ge 1 - u$ gives

$$D_{\mathrm{KL}}(P\|Q) - D_{\mathrm{KL}}(P\|R) = -\sum_i P_i \log \frac{Q_i}{R_i} \ge \sum_i P_i\Big(1 - \frac{Q_i}{R_i}\Big) = \frac{a-b}{a+b}\,(Q_j - Q_y) \ge 0. \tag{12}$$

For the equality, the coordinates outside the pair contribute $(1-a-b) - (1-Q_y-Q_j) = Q_y + Q_j - a - b$, and the two pair terms contribute $a + b - 2(aQ_y + bQ_j)/(a+b)$; the sum is $[(a+b)(Q_y+Q_j) - 2aQ_y - 2bQ_j]/(a+b) = (a-b)(Q_j - Q_y)/(a+b)$. Hence R minimizes $D_{\mathrm{KL}}(P\|\cdot)$ on $\{Q_j \ge Q_y\}$.

Union over competitors and part (i). The set on which y loses its strict lead is the finite union $\bigcup_{j\neq y}\{Q : Q_j \ge Q_y\}$. By the previous step every member has divergence at least $\min_j \psi(P_y, P_j) = \tau(P, y)$, and the minimizing pair average attains it. This proves the "minimum" claim of (ii) and, by contraposition, part (i). The margin τ is positive: each pair average differs from P when $P_y > P_j$, and Gibbs' inequality is strict for distinct distributions.

Runner-up. For fixed a and $0 < b < a$, $\partial\psi(a,b)/\partial b = \log[2b/(a+b)] < 0$, so $\psi(a,\cdot)$ is decreasing and the largest competitor probability, the runner-up, attains the minimum in τ.

Strict overtake. Let $j^\star$ be the runner-up and R its pair average. Set $Q_y = R_y - \epsilon$ and $Q_{j^\star} = R_{j^\star} + \epsilon$ with $0 < \epsilon < (a+b)/2$, leaving the other coordinates of R unchanged. Then Q is strictly positive and normalized with $Q_{j^\star} > Q_y$, and $D_{\mathrm{KL}}(P\|Q) \to \psi(a,b)$ as $\epsilon \downarrow 0$ by continuity. For a strict teacher winner and a strict overtake, the last term in Equation (12) is positive, so τ is an infimum over strict overtakes and not a minimum. Attainment concerns the weak boundary $\{Q_j \ge Q_y\}$, where a decoder's tie-breaking rule may still select y.

Part (iii). Write $\psi(a,b) = (a+b)\, d(\pi\|\frac{1}{2})$ with $\pi = a/(a+b)$, so $\pi - \frac{1}{2} = (a-b)/[2(a+b)]$. The binary Pinsker inequality (Cover & Thomas, 2006) $d(\pi\|\frac{1}{2}) \ge 2(\pi - \frac{1}{2})^2$ gives the lower bound. For the upper bound, $\log x \le x - 1$ gives $d(\pi\|\sigma) \le \chi^2(\pi\|\sigma) = (\pi-\sigma)^2/[\sigma(1-\sigma)]$, which is $4(\pi - \frac{1}{2})^2$ at $\sigma = \frac{1}{2}$. Finally, $d(\pi\|\frac{1}{2}) = \log 2 - H(\pi)$ with H the binary entropy, which is positive for $\pi \in (0,1)$, and $a + b \le 1$; hence $\tau < \log 2$. □

For comparison, a probability gap $\gamma = a - b$ and Pinsker's inequality $\mathrm{TV}(P,Q) \le \sqrt{D_{\mathrm{KL}}(P\|Q)/2}$ give the sufficient condition $D_{\mathrm{KL}}(P\|Q) < \gamma^2/2$, because each coordinate moves by at most the total variation (Cover & Thomas, 2006). Part (iii) shows that $\tau \ge \gamma^2/(2(a+b)) \ge \gamma^2/2$, so the sharp margin τ is never smaller and is larger by at least the factor $1/(a+b)$, which is large when the two leading tokens carry little joint mass. A logit gap γ_z instead requires a logit perturbation bound, $\|z_Q - z_P\|_\infty < \gamma_z/2$.

T.6 WHOLE ANSWERS (COROLLARY 1)

Proof of Corollary 1. At $t = 0$ the student and teacher share the prefix h_0 (the question), and Theorem 1(i) gives the student's first token y_0. If the tokens before t agree, the student's next prefix is exactly h_t, and the same implication gives y_t. By induction the answers agree through the end-of-sequence token, and a common stopping rule stops both at step T. Under constrained decoding the same induction runs along the teacher's constrained path. The permitted set is a function of the prefix, so teacher and student use the same G_t at each step, and a decoder that renormalizes on G_t keeps the order of the probabilities. A singleton G_t fixes the token directly, and $G_t = \mathcal{V}$ is the greedy case. Otherwise $D_{\mathrm{KL}}(P^{G_t}_{h_t} \| Q^{G_t}_{h_t}) < \tau(P^{G_t}_{h_t}, y_t)$, where y_t is the teacher's strict winner within G_t, and Theorem 1(i) applies to the strictly positive conditionals on G_t. Since Lemma 1 gives $D_{\mathrm{KL}}(P^{G_t}_{h_t} \| Q^{G_t}_{h_t}) \le \ell_A(h_t)/P_{h_t}(G_t)$, the condition $\ell_A(h_t) < P_{h_t}(G_t)\,\tau(P^{G_t}_{h_t}, y_t)$ also suffices. At a supervised prefix, Proposition 2(i) with point masses gives $\ell_A(h_t) \le L_\nu(A)/\nu(h_t)$, so $L_\nu(A) < \nu(h_t)\,\tau(P_{h_t}, y_t)$ suffices for a greedy step there; at evaluation prefixes $\nu(h_t) = 0$. □

Agreement at an isolated teacher-forced prefix does not imply that the student reaches that prefix on its own, which is the exposure-bias problem of sequence prediction (Bengio et al., 2015; Ross et al., 2011); the corollary requires the bound along the whole teacher path. The coverage (support) condition cannot be dropped, by Proposition 2(iii). For example, take the tokens `6`, `0` and end-of-sequence. Suppose both readers put 0.8 on `6` at the supervised first prefix. At the unsupervised prefix `6`, the teacher puts 0.8 on `0` and 0.1 on end-of-sequence, and the student swaps these two values. The teacher then answers `60` and the student `6`. All preservation results concern the teacher's behavior; they preserve a teacher's wrong answer as faithfully as a right one.

T.7 SOFT VERSUS HARD TARGETS

Proposition 3 (Soft versus hard targets)**.** *Fix a history, a teacher distribution $P \in \Delta^\circ(\mathcal{V})$ and student logits z with $Q = \mathrm{softmax}(z)$.* (i) *$D_{\mathrm{KL}}(P\|Q) = H(P,Q) - H(P)$ and $\nabla_z D_{\mathrm{KL}}(P\|Q) = Q - P$.* (ii) *For a label $y \sim P$, the hard-label gradient $\nabla_z[-\log Q_y] = Q - e_y$ is unbiased for $Q - P$ and has total variance $\mathrm{tr}\,\mathrm{Cov}(e_y) = 1 - \sum_i P_i^2$.* (iii) *For a fixed label t (for example, the next source token), $-\log Q_t$ can be made arbitrarily small while $D_{\mathrm{KL}}(P\|Q)$ grows without bound, since $P_t < 1$.*

Proof. (i) $D_{\mathrm{KL}}(P\|Q) = -\sum_i P_i \log Q_i + \sum_i P_i \log P_i$. With $\partial \log Q_i/\partial z_k = \delta_{ik} - Q_k$, $\partial_{z_k} D_{\mathrm{KL}} = -\sum_i P_i(\delta_{ik} - Q_k) = Q_k - P_k$. (ii) By (i) with $P = e_y$, the hard-label gradient is $Q - e_y$; $\mathbb{E}[e_y] = P$, and $\mathrm{Cov}(e_y) = \mathrm{diag}(P) - PP^\top$ has trace $1 - \sum_i P_i^2$. (iii) Let $z_t \to \infty$ with the other logits fixed. Then $Q_t \to 1$, so $-\log Q_t \to 0$, while $Q_i \to 0$ for every $i \ne t$. Because P is strictly positive and $|\mathcal{V}| \ge 2$, some $P_i > 0$ with $i \ne t$, and $P_i \log(P_i/Q_i) \to \infty$ while every other term is bounded below. □

Soft per-token targets on teacher-forced prefixes are word-level knowledge distillation (Kim & Rush, 2016). At a history that occurs once, as each continuation history does, a hard label is a single draw, and the source token is not even a draw from the full-context teacher. The soft target removes this noise, which is the variance-reduction view of distillation (Hinton et al., 2015; Menon et al., 2021), and it minimizes the quantity that Theorem 1 and Corollary 1 require. The two objectives coincide as P_h approaches a point mass on the source token; they differ exactly where the full-context teacher is uncertain. The proposition does not claim that hard-label refinement performs worse on our tasks. Our ablation tests it (Table 3): because the full-context teacher is nearly one-hot on source continuations, hard labels reach the same accuracy (-0.7 [-2.6, 1.2] constrained recall). Eyuboglu et al. (2026) report that next-token training on the raw corpus is not competitive with in-context learning; our initialization ablations locate the difference in the starting rows (Section 5.5).

T.8 FIRST-TOKEN MARGINS ON THE EVALUATION DATA

For each first-read number-recall answer we log the top-5 token ids at the first answer token, their BF16 logits and the log-normalizer over the full vocabulary, so the five probabilities are exact given the logged logits. The teacher's margin τ is exact, since its runner-up is among its top five. For a student we lower-bound ℓ_A at that token by the binary divergence $d(P_y\|Q_y)$ on the partition

State	First token differs (%)	Lower bound $\geq \tau$ (%)	Median lower bound (nats)	Disagreements with bound $\geq \tau$
AM	83.1	83.6	4.019	761 / 761
Self-study (256)	30.0	31.8	0.177	275 / 275
Self-study (exposure)	28.6	31.6	0.175	262 / 262
NCC	14.4	16.5	0.031	132 / 132

Table 53: **First answer token of the 916 number-recall questions** (greedy, first reads). By Theorem 1(i), every first-token disagreement requires $\ell_A \geq \tau$; the last column counts disagreements for which the lower bound $d(P_y \| Q_y)$ from the logged probabilities (Appendix T.8) already shows this, which it does for every disagreement of every state. The converse does not hold: a large divergence permits a changed decision without forcing one.

$\{y, \mathcal{V} \setminus \{y\}\}$, where y is the teacher's first token: Equation (3) with $G = \{y\}$ gives $\ell_A \geq d(P_y \| Q_y)$, since its other two terms are nonnegative. The teacher's P_y is always logged. If y is among the student's logged top five, Q_y is exact. Otherwise $Q_y \leq p_{(5)}$, the student's fifth logged probability, and since $d(P_y \| q)$ decreases in q on $(0, P_y]$, we use $d(P_y \| p_{(5)})$ when $p_{(5)} < P_y$ and 0 otherwise. The full-context teacher is nearly deterministic at this token: its top probability exceeds 0.99 on 93.1% of the 916 questions, $\tau \geq 0.69$ nats on 81.2% (the supremum is $\log 2 \approx 0.6931$), and the 5th percentile of τ is 0.602. A student's first token is the one it actually decoded; in seven student answers the logged BF16 logits tie at the top, and the decoder took the tied token with the lowest id.

T.9 Numerical checks

Each result was checked on random instances in double precision, and Theorem 1(iii) in 60-digit arithmetic because the two sides cancel for tiny margins. Table 54 lists the checks.

T.10 Machine-checked statements

A Lean 4 formalization (v4.33.1; de Moura & Ullrich, 2021), with Mathlib (The mathlib Community, 2020), machine-checks, for finite categorical distributions, the positivity, normalization and KL value of the pair average in Equation (11), its minimality on each competitor boundary and on their union (Theorem 1(ii), weak boundary), the positivity of the margin for a strict winner, winner preservation below the margin (Theorem 1(i)), the inequality Equation (12), the covered-prefix bound (the point-mass case of Proposition 2(i)), the induction from per-step bounds to an identical greedy sequence (Corollary 1), and a missing-prefix counterexample. The proofs use only Lean's standard axioms (`propext`, `Classical.choice`, `Quot.sound`). Proposition 1 (including its ellipsoid clause), Lemma 2, the general forms of Proposition 2, Lemma 1, the runner-up reduction, the strict-overtake infimum, Theorem 1(iii), the constrained case of Corollary 1 and Proposition 3 are written proofs, each checked numerically. None of these statements concerns model logits, floating-point arithmetic or how well a trained cache generalizes to unseen histories.

Result	Check	Outcome
Proposition 1	12-row head ($d = 4$); 6-row cache with keys and biases fitted to the mass at 6 reference queries	rel. error $\leq 8.0 \times 10^{-15}$ there; $\geq 3.0 \times 10^{-7}$ on 2×10^5 fresh queries
	sign changes of $Z - Z'$ on 500 random lines	at most 4 (bound 17)
	optimized m-row caches on a ball, $m = 2, 4, 6, 8, 11, 12$	mean sq. rel. error 4.8×10^{-5}, 1.5×10^{-5}, 1.7×10^{-6}, 3.1×10^{-7}, 2.2×10^{-7}, 1.4×10^{-31}
	selection; one appended row	$Z - Z' \geq 2.27$; outputs agree only at the predicted w
	$m = 1$, $d = 2$; random lines with coinciding, cancelling or absent compact rows (3,000); circle sections of random ellipsoids, $d = 2$–5 (300)	agreement along a whole curve; zeros $\leq n + m - 1$, bound attained; $g \not\equiv 0$ on every section
Proposition 2	20,000 random (μ, ν, ℓ); Donsker–Varadhan bound with $\lambda \in \{0.1, 1, 3\}$; bounded form; tilted distribution	0 violations; tilt attains equality to 2×10^{-14}
Lemma 1	2,000 random (P, Q, G), $\lvert\mathcal{V}\rvert \leq 29$	max error 3.6×10^{-15}
Theorem 1	Numerical minimum (SciPy SLSQP; Virtanen et al., 2020) of $D_{\mathrm{KL}}(P\lVert Q)$ over the tie set, 300 random P	max rel. deviation from τ: 6.1×10^{-8}
	2×10^5 random Q (88,043 with $D_{\mathrm{KL}} < \tau$)	0 winner changes
	10^5 perturbations of the pair average (39,220 with $D_{\mathrm{KL}} < \tau$)	0 winner changes
	runner-up attains τ (20,000 cases); strict overtake $\epsilon \to 0$	always; $D_{\mathrm{KL}} \to \tau$ from above
	two-sided bounds of (iii), 50,000 random P	0 violations; $\tau/(\gamma^2/2)$ median 3.10, min 1.00
Corollary 1	4,000 random prefix trees ($\lvert\mathcal{V}\rvert = 5$, depth ≤ 6)	1,210 certified paths, 0 mismatches
	constrained certificate, 10^5 random (P, Q, G)	44,481 certified, 0 violations
	4,000 random prefix trees with permitted sets $\mathcal{V}$, singletons or triples	2,170 certified paths, 0 mismatches
Proposition 3	identity, finite-difference gradient, 4×10^5 label draws	errors 2.3×10^{-15}, 1.8×10^{-9}; variance 0.7992 vs. 0.7992

Table 54: **Numerical checks of the theory.** Random instances; "violations" counts cases that contradict the stated result.

U Discussion of the theoretical results

Supervision distributions. Averaged over spans, Equation (2) is L_ν with $\nu(h_{b,t}) = 1/(128\, L_b)$, positive at the history preceding every source token; self-study spreads ν over the answer positions of its 256 conversations. Evaluation prompts are supervised histories for neither method.

Compaction. In one attention head, a cache row holds a key $k \in \mathbb{R}^d$, a log-mass bias b (zero in an ordinary cache, β in AM) and a value. A query q gives the rows the attention mass $Z(q) = \sum_i e^{\langle q, k_i \rangle + b_i}$, with rotary position and scale absorbed into q and k; Z is the normalizer of attention viewed as a kernel smoother (Tsai et al., 2019), and AM adds β to match it (Zweiger et al., 2026). The proof of Proposition 1 also bounds the zeros on a generic line and describes head outputs after appended rows (Appendix T.1). On a generic line the mass difference is a nonzero sum of at most $n+m$ exponentials with distinct rates, which by Rolle's theorem has at most $n+m-1$ zeros (Pólya & Szegő, 1998). The ellipsoid clause matters for our reader, whose per-head query RMS normalization (QK-norm) places the queries at each position on such ellipsoids, themselves Lebesgue-null. The result is exact but qualitative: it rules out exact reproduction, not small error, and complements worst-case lower bounds on the memory and time that attention requires (Haris & Onak, 2025; Alman & Song, 2023). AM decides where to be accurate with a finite set of reference queries, and selection methods with a positional or attention-based rule (Xiao et al., 2024; Li et al., 2024; Kim et al., 2025); neither consults the reader's next-token distributions.

Transfer. Parts (i) and (ii) of Proposition 2 are the importance-weighting and Donsker–Varadhan forms of a change of measure (Shimodaira, 2000; Cortes et al., 2010; Donsker & Varadhan, 1975; Boucheron et al., 2013), the same source-loss-plus-divergence form as domain-adaptation bounds (Ben-David et al., 2010); part (iii) shows that coverage cannot be dropped. A gap of the same kind, between teacher-forced and self-generated prefixes, causes exposure bias in sequence prediction and motivates on-policy distillation (Bengio et al., 2015; Ross et al., 2011; Agarwal et al., 2024). Since no evaluation prompt is a training history for either method, Proposition 2 does not guarantee test answers: transfer to questions runs through the KV rows that all histories share, which it does not model. It locates where each objective lacks direct control.

Margins. Lemma 1 is the chain rule for KL divergence (Cover & Thomas, 2006). By Theorem 1, the cheapest way to create a tie is to average the two leading probabilities, so the margin τ is set by the teacher's top two tokens; by part (iii) it is at least $1/(a+b)$ times the threshold $(a-b)^2/2$ that Pinsker's inequality gives. Geometrically, τ is the radius of the KL ball around P within which the winner is kept, as in certified robustness; unlike certificates for Rényi divergences in the reverse argument order (Li et al., 2019), it uses the forward KL that Equation (2) minimizes. The optimum ranges over all categorical distributions, of which a frozen reader with a fixed-size cache realizes only some (cf. Yang et al., 2018); for our model, part (i) is a sufficient condition and part (ii) a lower bound on the divergence needed to change a decision. At the first answer token of the number-recall questions, the teacher's top probability exceeds 0.99 on 93.1% of questions and the median τ is 0.693 nats, essentially the supremum $\log 2$; a binary lower bound on ℓ_A from the logged top-5 probabilities already reaches τ on 83.6% of questions for AM, 31.8% for self-study and 16.5% for NCC (Appendix T.8). Forward KL to soft targets is exactly the quantity these results require: at a single history, the hard-label loss on the source token can approach zero while ℓ_A grows without bound (Proposition 3), although on our source continuations the full-context teacher is nearly one-hot and the two losses reach the same accuracy (Section 5.5).

V PER-PAPER RESULTS

Table 55 lists number-constrained recall and QA F1 for every evaluation paper, identified by its arXiv ID. Each entry averages the paper's questions, and each column mean is the paper-averaged score in Table 1. SS_{256} and SS_{exp} are the self-study checkpoints after 256 updates and at matched exposure. Figure 14 compares them pairwise.

	Constrained number recall ↑					QA F1 ↑				
Paper	Full	AM	NCC	SS_{256}	SS_{exp}	Full	AM	NCC	SS_{256}	SS_{exp}
1606.00189	100.00	56.25	87.50	75.00	68.75	36.96	16.67	48.15	50.00	58.18
1606.02006	100.00	56.25	93.75	93.75	93.75	58.33	83.33	83.33	83.33	83.33
1610.08597	100.00	50.00	68.75	43.75	43.75	19.44	21.05	21.05	40.54	36.31
1701.00185	100.00	37.50	50.00	68.75	75.00	42.39	38.86	40.26	36.10	37.94
1702.06777	100.00	70.00	90.00	80.00	80.00	15.35	7.48	10.73	13.08	13.13
1703.04617	100.00	25.00	62.50	56.25	62.50	70.59	21.54	53.12	49.70	49.70
1704.00253	100.00	66.67	75.00	75.00	75.00	0.00	0.00	0.00	0.00	0.00
1705.07368	100.00	50.00	75.00	62.50	68.75	38.58	30.92	28.00	42.60	40.66
1707.06939	100.00	53.85	92.31	76.92	76.92	61.25	12.50	44.15	43.92	43.92
1708.01776	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00	0.00	0.00
1708.09157	100.00	61.54	76.92	69.23	76.92	24.33	27.11	24.70	12.50	29.04
1710.07395	100.00	75.00	81.25	87.50	87.50	70.43	60.99	79.51	41.42	44.28
1711.04457	100.00	50.00	62.50	68.75	68.75	65.33	41.67	50.61	62.22	62.22
1804.08050	100.00	75.00	100.00	100.00	100.00	33.33	33.33	33.33	33.33	33.33
1806.04524	100.00	93.33	100.00	100.00	100.00	76.30	76.30	79.63	76.30	76.30
1806.04535	100.00	93.75	100.00	100.00	100.00	81.86	52.94	57.92	53.33	50.28
1807.00868	100.00	75.00	81.25	68.75	68.75	60.71	63.74	46.84	39.35	39.35
1808.05077	100.00	56.25	87.50	75.00	75.00	81.25	90.00	84.38	62.50	62.50
1808.09920	100.00	31.25	43.75	62.50	75.00	62.46	57.15	67.14	54.84	64.52
1809.08731	100.00	50.00	50.00	62.50	68.75	75.00	80.00	75.00	41.67	41.67

Continued on the next page

Table 55 continued

	Constrained number recall ↑					QA F1 ↑				
Paper	Full	AM	NCC	SS_{256}	SS_{exp}	Full	AM	NCC	SS_{256}	SS_{exp}
1810.00663	100.00	43.75	50.00	68.75	75.00	31.84	33.73	36.28	40.02	26.52
1810.08732	100.00	18.75	62.50	75.00	68.75	42.22	48.41	27.69	39.29	40.81
1811.00942	100.00	68.75	93.75	93.75	93.75	49.21	53.33	55.87	61.82	61.82
1811.09353	100.00	77.78	100.00	100.00	100.00	82.86	87.37	80.91	82.86	82.86
1812.08879	100.00	80.00	100.00	80.00	80.00	50.00	50.00	50.00	50.00	50.00
1901.09755	100.00	68.75	81.25	81.25	87.50	56.64	42.34	47.32	42.46	44.36
1904.00648	100.00	75.00	100.00	93.75	87.50	12.42	16.09	12.81	29.65	22.72
1904.08386	100.00	83.33	100.00	100.00	83.33	28.09	17.86	33.74	9.14	20.35
1905.01976	100.00	75.00	93.75	93.75	93.75	50.00	100.00	50.00	50.00	50.00
1905.06906	100.00	93.75	100.00	100.00	100.00	33.33	33.33	33.33	33.33	33.33
1905.11901	100.00	56.25	81.25	75.00	68.75	39.23	14.54	38.33	11.51	20.65
1906.02715	100.00	18.75	50.00	56.25	62.50	27.33	0.00	12.90	23.61	31.45
1906.11180	100.00	25.00	50.00	75.00	75.00	33.33	33.33	33.33	34.92	35.00
1907.08501	100.00	81.25	93.75	87.50	87.50	58.29	58.29	56.51	58.29	58.29
1909.00088	100.00	81.25	93.75	75.00	81.25	65.85	8.33	41.67	65.85	65.85
1909.00694	100.00	75.00	87.50	87.50	87.50	36.04	34.76	35.77	33.22	32.93
1909.01515	100.00	50.00	62.50	62.50	56.25	48.33	48.33	56.67	48.33	61.43
1909.02265	100.00	81.25	87.50	93.75	93.75	69.96	67.28	67.28	67.28	67.28
1909.05358	100.00	62.50	81.25	81.25	81.25	59.95	19.93	48.77	35.00	35.00
1909.05438	100.00	43.75	68.75	62.50	68.75	31.32	26.73	25.40	23.67	24.11
1909.08167	100.00	100.00	100.00	100.00	100.00	11.00	35.56	27.68	25.54	18.84
1909.08402	100.00	56.25	62.50	81.25	81.25	73.63	33.06	47.53	59.89	32.08
1910.00458	100.00	68.75	75.00	87.50	75.00	46.38	38.27	38.36	48.30	48.30
1910.02789	100.00	50.00	93.75	75.00	75.00	55.35	49.91	30.15	30.20	33.75
1910.03891	93.75	68.75	62.50	81.25	81.25	18.88	17.42	28.69	31.90	30.58
1910.04601	100.00	43.75	68.75	56.25	56.25	45.22	28.00	54.55	49.05	51.90
1910.08418	100.00	56.25	81.25	87.50	87.50	70.04	29.57	43.05	31.67	25.24
1910.11768	100.00	72.73	100.00	81.82	90.91	51.67	50.38	50.38	54.23	50.38
1910.11949	100.00	62.50	75.00	75.00	68.75	58.93	38.63	63.01	59.33	68.37
1910.13215	100.00	43.75	56.25	68.75	68.75	33.33	33.33	33.33	0.00	0.00
1911.00069	100.00	68.75	81.25	81.25	75.00	58.21	69.72	69.72	78.29	77.72
1911.01188	100.00	37.50	50.00	43.75	50.00	17.04	17.44	20.76	25.53	24.92
1911.06171	100.00	100.00	100.00	100.00	100.00	76.99	40.57	68.88	45.36	51.42
1911.08370	100.00	68.75	93.75	87.50	87.50	50.00	50.00	50.00	50.00	50.00
1911.11744	100.00	100.00	100.00	100.00	100.00	38.10	17.28	18.84	25.96	25.96
1912.00871	100.00	68.75	87.50	81.25	81.25	63.22	40.44	58.02	63.22	43.22
1912.02761	100.00	100.00	100.00	100.00	100.00	50.00	50.00	50.00	50.00	50.00
1912.13072	100.00	37.50	75.00	50.00	43.75	51.52	33.33	47.01	50.88	50.88
2002.05058	100.00	50.00	81.25	56.25	62.50	4.35	5.26	37.70	41.40	42.81
2002.06424	100.00	46.15	61.54	46.15	46.15	12.86	15.62	40.79	69.51	57.41
2003.04967	100.00	43.75	56.25	62.50	62.50	39.70	16.82	50.00	40.35	37.85
2003.06651	100.00	68.75	81.25	62.50	68.75	40.23	39.39	38.89	71.11	72.73
2003.08370	100.00	50.00	87.50	68.75	75.00	42.26	30.66	39.71	46.44	48.11
2004.04228	100.00	37.50	62.50	50.00	50.00	50.00	50.00	50.00	50.00	50.00

Table 55: **Per-paper results on the 64 evaluation papers** (%). Number recall uses number-constrained decoding; QA is Qasper answer F1. SS_{256}: self-study, 256 updates; SS_{exp}: self-study, exposure-matched. The two self-study checkpoints coincide for two papers.

Examples. A source-grounded review of the answers, AI-assisted and used descriptively, illustrates typical differences. In paper 1812.08879, NCC recalls the 300 target-domain training examples of the K-shot setting, whereas both self-study checkpoints answer 10, a different count (generated sentences per semantic representation). In paper 1707.06939, the direction reverses: NCC answers 119, the number of rating workers, where the question asks for the 4,300 ratings, which both self-

study checkpoints recall. On document QA, lexical F1 can disagree with correctness: in paper `1808.05077`, the answer `CNN` receives F1 of 1, while the correct expansion "Convolutional Neural Networks (CNNs)" receives 0. All scores use the original annotations.

W QUESTION-ONLY READER

The question-only reader, the question-only baseline of Kaushik & Lipton (2018) applied to number recall, measures how much of the number-recall task can be solved without the document. It answers the same 916 questions from an empty state: no document, no compact KV rows and no recurrent state, and every call starts at position zero. The user message is the unchanged question, which already contains the paper title and the masked sentence (Figure 1). The system instruction below replaces that of the document readers and allows prior knowledge and inference from the visible sentence. Model, precision, decoders, stopping rule and the 32-token limit are those of the main study, and scores again use the first of two reads. QA is not evaluated for this reader.

Question-only system instruction

```
Complete the missing scalar using the supplied cue and your existing
knowledge or deductions.  No full document or compiled memory is
provided.  Return only the missing numerical value in Arabic digits;
preserve the unit already shown in the cue.  If you cannot determine it,
answer Unanswerable.  Do not give an explanation.
```

In this prompt, which we use verbatim, "cue" refers to the question (paper title and masked sentence), not the source cue of Section 2, and "scalar" means number.

The question-only reader recalls 31.2% of the numbers with greedy decoding and 31.4% with number-constrained decoding (278 and 280 of the 916 questions), so, unlike AM, it gains almost nothing from the constraint (Table 56). Every state in Table 56 is far above it, and NCC adds 48.4 points [43.7, 52.9] under constrained decoding. At the item level, NCC answers 455 questions that the question-only reader misses, misses 15 that it answers, and has the same outcome on the remaining 446. The question-only reader also differs from the compact states in its system instruction, starting position and recurrent state. It therefore bounds what the question alone reveals rather than isolating the contribution of the KV rows, which the recurrent-state-only control addresses (Section 5.1 and Appendix H). Appendix Q.5 compares the compact states on the 636 items that the question-only reader answers incorrectly.

State	Greedy	Constr.	Gain over question only
Question only	31.2	31.4	–
Full context	99.7	99.9	+68.5 [63.8, 72.9]
Attention Matching	16.7	62.3	+30.9 [26.3, 35.6]
Self-study, 256 updates	65.1	77.4	+45.9 [41.5, 50.5]
Self-study, exposure-matched	65.8	77.9	+46.4 [41.8, 51.0]
NCC (source continuations)	**76.4**	**79.8**	**+48.4** [43.7, 52.9]

Table 56: **Number recall with and without the document** (paper-averaged over 64 papers, %). The last column is the gain over the question-only reader under number-constrained decoding, with 95% paired bootstrap intervals over papers.