You don't need long context

What compaction and selective reading actually establish

Peter Belcak et al. (12 authors)
Research-group presentation

arXiv:2611.05306v1 5 November 2026

Technical report, 3 November 2026

Long work does not always require a long live context

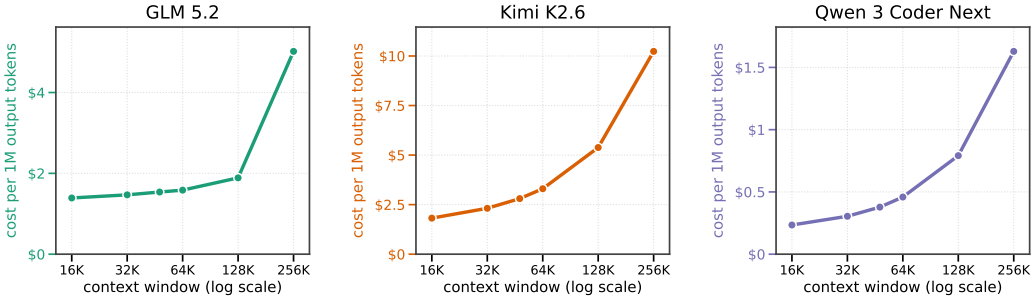
Paper's central test

Can an agent finish long-horizon or long-input tasks while each model call sees only compact state and selected evidence?

- Six long-horizon and seven long-input benchmark suites; primary models: Claude Opus 4.5 and GPT 5.6 Sol.
- Usually similar task scores with shorter live contexts; one-action handoffs work on two software tasks, with model-specific losses.
- Publication: technical report; no formal venue found as of 5 Nov 2026.
- Paper-specific public code not found as of 5 Nov 2026; the base mini-swe-agent is public.

Paper, Abstract and Secs. 2–5, PDF pp. 1, 4–15.

Longer windows make decoding more expensive



- At 256K versus 16K, cost per output token rises by 3.6× (GLM), 5.6× (Kimi), and 7× (Qwen).
- Fixed GPU allocation, cached prefixes, 1,024 generated tokens per request; this measures decode throughput.

Paper, Fig. 1 and Sec. 2.5, PDF pp. 1, 8. Original figure from author TeX.

The paper separates three research questions

1. **Long horizon:** Must all past actions and observations remain in the next prompt?
2. **Long input:** Must a large source be loaded all at once?
3. **One action:** Can a fresh call continue after every individual tool action?

Unifying idea

Preserve an action-ready state; keep original evidence available for selective retrieval.

Paper, Sec. 1, PDF pp. 2–3.

The test bed spans workflows and document tasks

Six long-horizon suites	Seven long-input suites
SWE-bench Verified and Multilingual	LongBench v2, LOONG, AA-LCR
Terminal-Bench 2.1	NIAH, RULER
FRAMES	Counting-Stars, BABILong
τ^2-bench	Source material is large; the task question may be short.
BFCLv3	

Interaction history grows as the agent acts.

Paper, Secs. 2.2–2.3, PDF pp. 4–6. Metrics differ by suite.

Three regimes vary how much history survives

FHA: full history	EoCC: threshold	SAC: every action
Task + every action + every observation stay in one growing conversation.	At a threshold, write a bounded handoff; restart with original task and compact state.	One tool action per conversation; update semantic state before the next fresh call.
Reference for long-horizon tasks	Example: 16K limit / 12K threshold	Tested on two coding suites

Paper, Sec. 2.4 and Fig. 3, PDF p. 7. Abbreviations: FHA, EoCC, SAC.

A useful handoff is an execution brief

What the compact state carries

Original task; verified progress; current work and intent; completion criteria; next action; pointers to relevant artifacts.

- It prioritizes facts needed for future work over a balanced narrative of the past.
- The environment and source documents persist across resets.
- A lost fact can be recovered if its original record remains searchable.

The paper's coin-toss example shows why a fixed-size handoff alone cannot retain an arbitrarily long sequence of exact observations.

Paper, Sec. 3.1, PDF p. 10; Apps. A and G, PDF pp. 20, 30–31.

Large inputs become searchable external sources

Question + file manifest $\longrightarrow$ search entities and claims $\longrightarrow$ bounded passage reads $\longrightarrow$ answer

- Long inputs are stored as files; Bash searches scan them, while only returned excerpts enter the live conversation.
- The provided bounded reader defaults to 6,000 characters and caps a read at 8,000.
- This changes the *input presentation* as well as context policy. On many long-input runs, median compactions are zero.

Paper, Sec. 2.4, PDF p. 7; App. E, PDF pp. 25–26; App. F, PDF pp. 27–29.

Measure peak context, total work, and task score separately

$$L_{\text{peak}} = \max_j L_j,$$

$$L_{\text{total}} = \sum_j L_j,$$

where L_j is the final length of conversation stretch j.

- Peak is the largest live conversation during one task.
- Total sums final lengths across restarted conversations; it may exceed the window limit.
- Token spend separately counts uncached input, cached input, and generated output.

Paper, Sec. 2.4, PDF p. 7; App. C table captions, PDF pp. 22–23. Equations restate the paper's definitions.

RQ1: tight limits mostly preserve long-horizon scores

Suite	Limit / trigger	Opus baseline → EoCC	GPT baseline → EoCC
SWE-bench Verified	16K / 12K	70.0 → 68.8	84.4 → 84.6
SWE-bench Multilingual	16K / 12K	66.3 → 64.3	79.3 → 77.7
Terminal-Bench 2.1	16K / 12K	47.2 → 48.3	87.6 → 85.4
FRAMES	16K / 12K	81.0 → 79.6	90.2 → 87.1
BFCLv3	12K / 8K	62.0 → 62.5	39.5 → 39.5
τ^2-bench	8K / 6K	79.3 → 78.8	73.9 → 73.9

Across 22 settings per model, at most five lose more than 2 percentage points.

Paper, Fig. 5, PDF p. 9; App. C, Tables 3–8, PDF pp. 22–23. Scores are suite-specific percentages.

RQ1: accuracy and priced cost are distinct claims

- EoCC accuracy changes range from -3.7 to $+5.6$ points for Opus and -3.3 to $+4.4$ for GPT across six suites and settings.
- Modeled optimized-retail savings range from 2.2–8.8$\times$ and 2.4–5.0$\times$, respectively.
- But on SWE-bench Verified, GPT at 16K / 12K has *higher nominal retail cost*: \$203.12 versus \$167.68 for full history.

Interpretation

Shorter windows preserve many task scores. The savings multiples assume context-sensitive future prices.

Paper, Sec. 3 and Fig. 5, PDF pp. 9–10; App. C, Table 3, PDF p. 22.

RQ2: real document tasks benefit from selective reading

Suite	Reported metric	Opus	GPT
LongBench v2	accuracy, 16K / 12K	68.0 → 67.6	70.8 → 72.6
LOONG	mean score, 32K / 24K	70.1 → 80.2	79.3 → 84.0
AA-LCR	accuracy, 16K / 12K	65.0 → 84.0	82.0 → 83.0

- LOONG perfect-answer rate also rises: Opus 38.9% → 53.7%; GPT 44.4% → 52.1%.
- Files remain accessible; only selected passages enter the active context.

Paper, App. F, Tables 11–13, PDF pp. 27–28. Each arrow compares with the full-input baseline.

RQ2: retrieval-heavy suites show larger gains

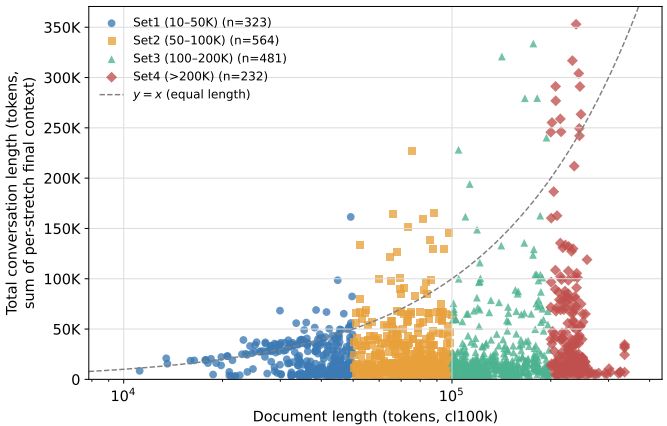
Suite	Metric	Opus baseline → selective	GPT baseline → selective
NIAH	accuracy	60.0 → 100.0	80.0 → 100.0
RULER	mean recall	62.5 → 95.7	65.5 → 96.4
Counting-Stars	exact match	65.1 → 81.2	76.6 → 81.2
BABILong	accuracy	67.1 → 92.0	70.7 → 94.0

NIAH's lexical overlap makes its inserted facts especially searchable. The other suites include aggregation or reasoning beyond locating one sentence.

Modeled optimized-retail savings across seven suites: 6.5–54.6× (Opus), 4.4–233.3× (GPT); these are not observed bills.

Paper, Sec. 4, Fig. 6, PDF pp. 11–13; App. F, Tables 14–17, PDF pp. 28–29. Selective settings: NIAH 48K/36K; others 16K/12K.

Document size does not dictate conversation size



- LOONG: 1,600 multi-document questions.
- Horizontal: source length, log scale.
- Vertical: total conversation length across stretches.
- Many runs fall below $y = x$; some remain long.

Paper, Fig. 8, PDF p. 13. Opus at 48K / 40K. Original figure from author TeX.

RQ3: one action per context remains viable on coding

Model	Suite	FHA score	SAC score	FHA peak	SAC peak
Opus	SWE-bench	70.0	70.2	32 / 200	8 / 16
Opus	Terminal-Bench	47.2	51.8	27 / 130	9 / 19
GPT	SWE-bench	84.4	82.4	18 / 173	7 / 15
GPT	Terminal-Bench	87.6	84.3	17 / 133	8 / 37

- Scores are percentages; peak context is median / maximum in K tokens per task.
- GPT loses 2.0 and 3.3 percentage points; Opus matches or improves.

Paper, Sec. 5, Table 2, PDF p. 14. SAC was tested on these two software suites.

SSSM is a proposal motivated by SAC, not a trained model

Natural-language state s_t $\rightarrow$ choose action a_t $\rightarrow$ observe o_t $\rightarrow$ write updated state s_{t+1}

- A Semantic State Space Model would be post-trained to make each state sufficient for a later independent inference call.
- Unlike a neural SSM's hidden state, this state is interpretable language.
- The paper evaluates SAC with existing models; it does not train or benchmark an SSSM.

Paper, Sec. 5.1, PDF pp. 14–15. Diagram is a teaching restatement of the proposal.

Measured throughput supplies one cost ratio

$$r_M = \frac{C_{\mathrm{out},M}(16\,\mathrm{K})}{C_{\mathrm{out},M}(256\,\mathrm{K})} = \frac{T_M(256\,\mathrm{K})}{T_M(16\,\mathrm{K})}.$$

Measured model	r_M	256K / 16K output-cost ratio
GLM 5.2	0.2772	3.6×
Kimi K2.6	0.1777	5.6×
Qwen 3 Coder Next	0.1439	7.0×

The paper rounds their mean to $r = 0.20$ for its later price model.

Paper, App. B, Eq. (1), PDF p. 20; Fig. 4, PDF p. 8. T_M is sustained decode throughput.

Reported task-cost savings use a pricing model

$$\mathrm{EORC}_M(W) = I\, C_{\mathrm{input},M}(W) + H\, C_{\mathrm{cached},M}(W) + O\, C_{\mathrm{out},M}(W).$$

- I, H, O: aggregate millions of uncached input, cached input, and output tokens.
- Output price is interpolated between 20% of standard price at 16K and standard price at the reference window.
- Input/output and cached/uncached ratios are held fixed.

Practical reading

These are estimated optimized-retail costs, not observed benchmark invoices or end-to-end latency measurements.

Paper, App. B, Eqs. (2)–(4), PDF pp. 20–21.

The boundary is model and workflow dependent

Open-weight stress test

Qwen 3 Coder Next drops from 53.0% to 3.6% on SWE-bench Verified at 16K / 12K; all five tested open-weight models lose some SWE-bench accuracy at this tightest setting.

- The paper attributes Qwen's collapse largely to verbose trajectories leaving too little useful work between compactions.
- External source access is part of the successful short-context design.
- The evidence does not exclude tasks that benefit from a longer live context.

Paper, App. D, Tables 9–10, PDF p. 24; Secs. 3.1 and 6, PDF pp. 10, 15.

Takeaways for a research group

1. Distinguish *task duration* and *source size* from the information needed in the next model call.

2. A compact, action-ready state plus searchable records performs well on many tested workloads.

3. Evaluate accuracy, peak context, total token spend, actual price, and end-to-end latency separately.

4. SAC supports the SSSM idea, but the model itself remains a future research direction.

Paper, Secs. 3–6 and Apps. B, D, PDF pp. 9–15, 20–21, 24.

Paper and source links

Paper: Peter Belcak, Saurav Muralidharan, Jindong Jiang, Yonggan Fu, Wonmin Byeon, Hanrong Ye, Ryan Angilly, Ali Taghibakhshi, Shih-Yang Liu, Siru Ouyang, Greg Heinrich, and Pavlo Molchanov. *You don't need long context.* Technical report, 2026.

Versioned arXiv record PDF Author TeX source

Related base harness: SWE-agent/mini-swe-agent. This is not the paper's released experiment code.

Status checked 5 November 2026: no formal publication record or paper-specific public code found.