Context Language Models

Learning to manage the live context

Paper by Rulin Shao et al.

Research group reading presentation

arXiv:2609.37725v1 | 10 October 2026

Make context an editable file, then learn how to maintain it.

arxiv.org/abs/2609.37725v1 | Main discussion: 20–30 minutes

Different information needs different context operations

Information	Required operation	Fixed-summary difficulty
Exact identifiers	Preserve verbatim	Paraphrase loses identity
Current board or plan	Update one field	Rewriting adds work/errors
Large search results	Offload, evict, retrieve	Storage alone leaves history
Multi-agent progress	Maintain a live ledger	Old status accumulates

Research question

Can a model choose and implement its own context transformations?

Paper, Secs. 1–3, PDF pp. 1–4. Examples summarized for this presentation.

CLM controls the next context, including edits to history

Append-only LM

$$c_{t+1} = c_t \oplus f_\theta^{\mathrm{LM}}(c_t)$$

Output is appended to existing history.

Context Language Model

$$c_{t+1} = f_\theta^{\mathrm{CLM}}(c_t)$$

The model controls the transformation.

1. Mirror live context to a file; give its path to the model.
2. Allow general Bash/programmatic edits.
3. Synchronize the file with the next model invocation.

c_t: context at turn t; θ: model parameters; $\oplus$: concatenation.
Existing LMs can implement this interface without additional training.

Paper, Sec. 4.1, Eqs. (1)–(2), PDF p. 4.

Context editing can express reusable procedures

- Update task status in place.
- Keep conclusions and evidence pointers.
- Remove unhelpful search observations.
- Define a reusable compaction function.

Observed examples

163 ledger edits, with context held at 6–8K tokens.

A helper function reused 37 times in a research trajectory.

Multiple context files naturally represent multiple agents.

An external memory file stores material; a context file determines the next input.

Paper, Sec. 4.1 and Fig. 3, PDF pp. 4–5; App. A, PDF p. 14.

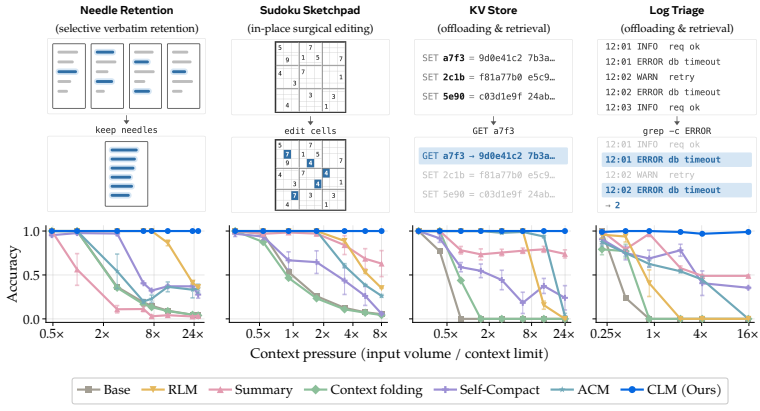
ContextBench isolates retention, updates, and retrieval

Task	Required behavior	Metric ↑
Needle Retention	Keep specified lines	Verbatim retention
Sudoku Sketchpad	Edit a 16×16 board	Exact board versions
KV Store	Store batches; answer 24 queries	Exact values
Log Triage	Offload logs; answer 24 queries	Exact answers

- GPT-5.4; 32,768 tokens, including a 2,048-token response reserve.
- Input arrives in live context before the agent can act.
- Grade from context; files alone receive no credit.
- Necessary retained information fits within the budget.

Paper, Sec. 3 and App. D–E, PDF pp. 3–4, 18–22; Table 3.

Precise edits remain effective as input pressure rises



Pressure = cumulative input / context limit. Higher accuracy is better.
All methods receive instructions adapted to their own mechanisms.

Paper, Fig. 2, PDF p. 4; GPT-5.4, 4 seeds per level, 8 at selected levels.

An edit can shorten context and still incur recomputation

$$F_t = C_{\text{token}}(U_t + G_t) + C_{\text{attn}}\left[\frac{P_t^2 - R_t^2}{2} + G_t P_t + \frac{G_t^2}{2}\right]$$

- P_t: prompt length; G_t: generated length; R_t: reusable prefix.
- $U_t = P_t - R_t$; trajectory cost is $\sum_t F_t$.
- C_{token}: MLP/projection cost per token; C_{attn}: cost per attention pair.

Reusable prefix R_t	18K	10K	0
Cost (10^{14} FLOPs)	1.41	5.74	10.81

Illustrative Qwen3.6-27B turn: $P_t = 20\text{K}$, $G_t = 500$.

Optimize the whole trajectory, including cache invalidation.

Paper, Eq. (3), PDF p. 5; App. C, Eq. (9) and Fig. 14, PDF pp. 17–18.

A skill document becomes an optimizable context policy

$$c_{t+1} = f_\theta^{\mathrm{CLM}}(c_t; s), \qquad s^* = \arg\max_s \mathbb{E}_{x\sim\mathcal{D}}[R(\tau(x; s))]$$

Textual evolution loop

Training rollouts $\longrightarrow$ propose skill rewrites $\longrightarrow$ development evaluation $\longrightarrow$ select a skill.

- s: natural-language guidance; τ: resulting task trajectory.
- Parameters and execution interface stay fixed.
- Assisted evolution: stronger external proposer.
- Self-evolution: the agent also proposes the next skill.

Freeze selection before the final held-out test evaluation.

Paper, Sec. 4.2, Eqs. (4)–(5), PDF pp. 5–6; App. E, PDF p. 23.

The efficiency advantage rewards cheaper successes

$$A_i^{\mathrm{eff}} = \begin{cases} \mathrm{clip}\left(\dfrac{\bar{c}_g - c_i}{\bar{c}_g}, -1, 1\right), & i \in \mathcal{G}_g^+, \\ 0, & i \notin \mathcal{G}_g^+, \end{cases} \qquad A_i = A_i^{\mathrm{out}} + w_{\mathrm{eff}} A_i^{\mathrm{eff}}.$$

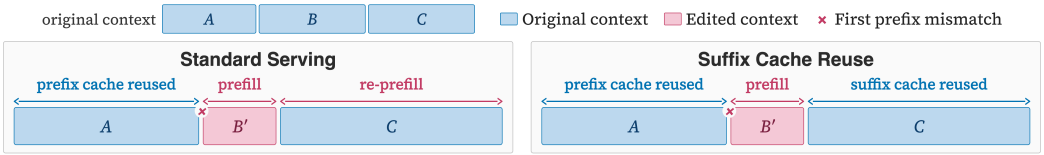
- c_i: trajectory FLOPs; $\bar{c}_g$: mean among successes in group g.
- Fewer than two successes: efficiency advantage is zero.
- Stepwise GRPO trains each segment under its original input.
- Main CLM recipe: $w_{\mathrm{eff}} = 0.25$ on context-management tokens.

Teaching example

Successful costs 1 and 3 PFLOPs give advantages $+0.5$ and -0.5.
A cheap failed trajectory receives no efficiency bonus.

Paper, Sec. 4.2, Eq. (6), PDF p. 6; App. A/E, PDF pp. 14, 24.

SCR reuses surviving suffix states after an edit



- Diff old and new prompts; relocate up to $K = 6$ surviving spans.
- Re-rotate cached keys to their new positions; prefill new text.
- Old states still encode the old prefix: reuse is approximate.
- Session-private cache slots; fall back if allocation fails.

Paper, Sec. 4.3, Fig. 4, PDF p. 6; App. B, PDF pp. 14–16.

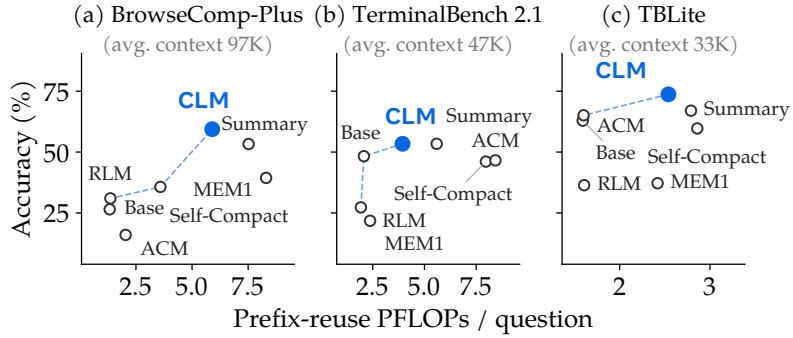
Evaluation spans minutes, hours, and multi-agent work

Setting	Main model	Outcome / resource measure
Coding + research	Qwen3.6-27B	Accuracy / PFLOPs per question
Mathematical search	Claude 4.6 Sonnet	Best score / 100 attempts or 5 h
EdgeBench-10	Qwen3.6-27B; Sonnet	Best-of-three score / 12 h
Software World	GPT-5.6-Sol, 6 agents	Downstream speedup / API spend

- Zero-shot baselines: Summary, MEM1, Self-Compact, ACM, RLM, Base.
- Fig. 5's 32K/100-turn caption differs from detailed budgets in App. E; backup slide lists both.

Paper, Sec. 5.1, PDF pp. 7–8; App. E, PDF pp. 19, 22–24.

CLM improves the accuracy–compute frontier



Task	Quality vs. Summary	Compute vs. Summary
BrowseComp-Plus	59.4%; +11.4% relative	−21.5%
TerminalBench 2.1	Matched accuracy	−29.5%
TBLite	73.7% vs. 67.0%	About −9%

Paper, Fig. 5 and Sec. 5.1.1, PDF p. 7; TB compute reduction also in introduction, p. 3.

General context control competes with specialized search

Method	Circle ↑	Heilbronn ↑	Ratio ↑	Overlap ↓
OpenEvolve	2.541	0.03127	0.07690	0.38123
OE-Agent	2.525	0.03053	0.07724	0.38167
CLM	2.618	**0.03653**	**0.07758**	**0.38094**
CLM + subagents	**2.636**	0.03617	**0.07758**	0.38109

- Same Claude 4.6 Sonnet model and evaluators; 32K context.
- One run per method; stop after 100 scored attempts or 5 hours.
- CLM vs. OpenEvolve: +16.8% Heilbronn; +3.0% circle packing.
- Best-of-run solutions; statistical repeatability is not measured here.

Paper, Table 1, PDF p. 7; Fig. 18, PDF p. 25; App. E, PDF p. 22.

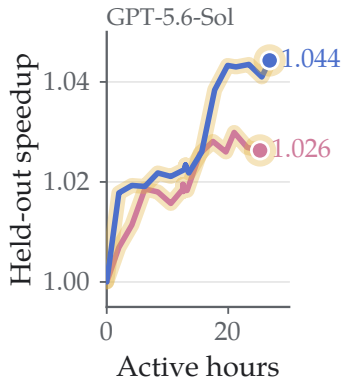
Long repository work benefits from compact working state

Model	Method (32K context)	Score ↑	PFLOPs/run ↓
Qwen3.6-27B	Summary	42.3	437
	CLM	**44.6**	**179**
	CLM + subagents	44.2	181
Claude 4.6 Sonnet	Summary	42.3	—
	CLM	**51.0**	—
	CLM + subagents	50.4	—

- EdgeBench-10: 12 hours; best score over 3 seeds per task.
- Qwen CLM: +2.3 score points with 59% less compute.
- Subagents add little at 32K; the 128K result differs (backup).

Paper, Sec. 5.1.2 and Fig. 6a, PDF p. 8; compute is mean per run, including subagents.

Software World tests transfer to unseen downstream work



- Six agents optimize interdependent repositories.
- GPT-5.6-Sol; 272K context; over 24 hours.
- Four unseen packages; 17 evaluation tasks.
- Geometric mean of speedup in executed instructions.

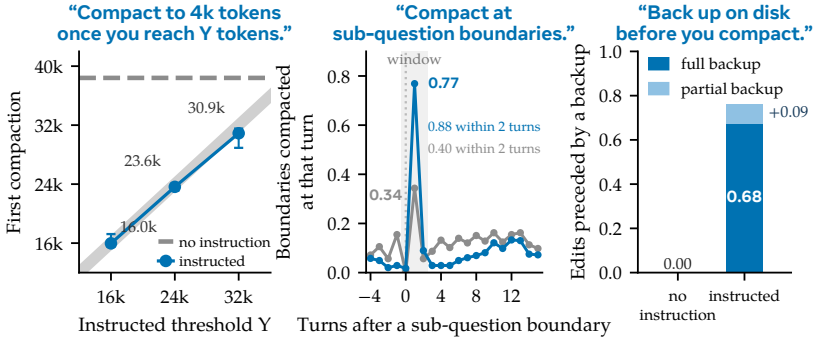
Interpret the 65% claim carefully

The paper reports 65% more *speedup gain* above 1.0 at matched spend.

Rounded final ratios: 1.044 vs. 1.026.

Paper, Fig. 6b and Sec. 5.1.2, PDF p. 8; App. E, PDF p. 23.

A sentence can change when and how context is edited



- Threshold requests 16K/24K/32K → first edits at 16.0K/23.6K/30.9K.
- Compaction within two turns of a boundary: 0.40 → 0.88.
- Full backup before an edit: 0.00 → 0.68; partial backup adds 0.09.

Paper, Fig. 7, PDF p. 9; Claude 4.6 Sonnet. These are behavior measures, not answer accuracy.

Evolved skills improve held-out context management

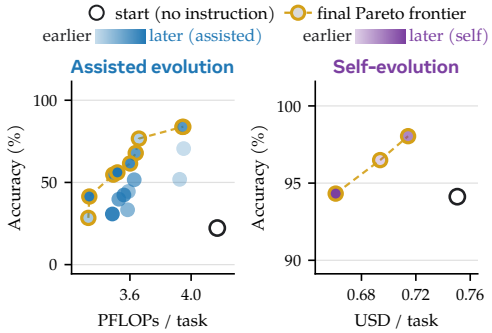


Fig. 8: KV Store evolution trajectories.

Assisted evolution: held-out test

KV Store: 38.3% → 74.2%

+35.9 percentage points.

- Qwen3.6-27B agent; Claude Fable 5.1 proposer.
- Development result is different: 22.3% → 83.8%.
- Opus 5 self-evolves from a much stronger initial policy.

Qwen cost excludes some unsuccessful/over-budget runs; it is not total search cost.

Paper, Figs. 8/20, PDF pp. 9/26; held-out result, App. F, p. 25; cost protocol, App. E, p. 23.

RL closes the small model's accuracy gap at lower cost

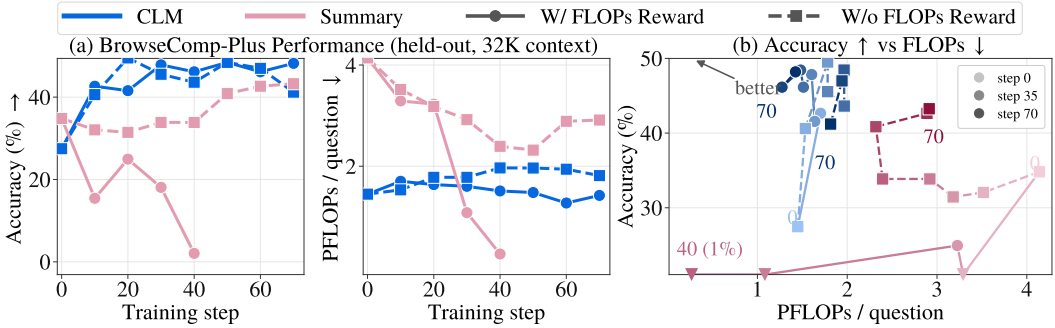
Qwen3.5-9B	Accuracy before → after	PFLOPs/Q before → after
Summary	34.7% → 42.1%	4.01 → 2.19
CLM	28.8% → **42.5%**	1.52 → **1.34**

- CLM gains 13.7 points (47.6% relative); own compute falls about 12%.
- After training: +0.4 points vs. Summary, with 38.8% less compute.
- Train on OpenResearcher; select on its validation set; test on BCP.

Appendix E describes the main Summary baseline as task-reward-only; CLM adds efficiency and tool/format terms.

Paper, Table 2, PDF p. 9; App. E, PDF pp. 23–24. Checkpoints selected by held-out validation.

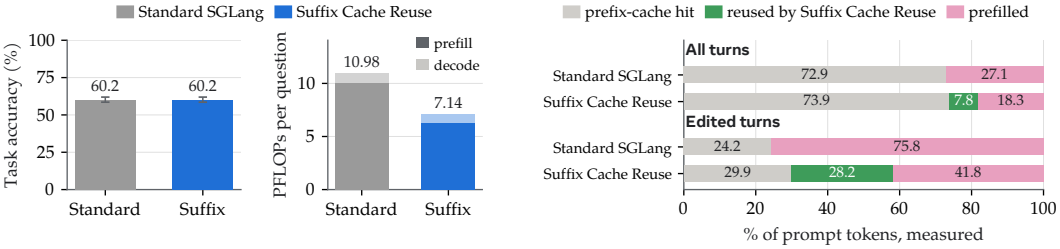
The reward ablation exposes instability in Summary



- Pink solid curve: Summary + FLOPs reward drops to about 1% at step 40.
- CLM's reward curves retain much higher accuracy at lower compute.
- This qualifies the p. 9 statement of no clear accuracy loss for either harness.

Paper, Fig. 21, PDF p. 26; discrepancy with Sec. 5.2, PDF p. 9. Interpretation from the plotted curves.

SCR cuts measured compute by 35% at matched accuracy



- Qwen3.6-27B, 830 BCP questions: 60.2% accuracy for both servers.
- 10.98 $\rightarrow$ 7.14 PFLOPs/question; these are separate SGLang runs.
- Of 7.8% extra token reuse, 5.3 points follow reasoning stripping; 2.5 follow other edits.
- Token fractions in the right panel are not FLOPs fractions or latency.

Paper, Fig. 9, PDF p. 9; stripping decomposition, Fig. 12, PDF p. 16.

Three conditions shape how these results should be used

Model capability and task organization

Smaller models edit less effectively. Larger context can change whether subagents help.

State semantics and context integrity

SCR retains old-prefix information. Writable context can also preserve injected or self-generated instructions.

Evaluation protocol

Best-of-run math, best-of-three EdgeBench, and matched-spend software gains measure different outcomes. Some budget descriptions differ across sections.

Paper, Sec. 6, PDF p. 10; App. B/E/F, PDF pp. 15, 22–25.

The main contribution is a learnable context interface

1. **Expressiveness:** live context becomes a model-editable artifact.
2. **Learning:** skills and RL optimize how that artifact evolves.
3. **Serving:** cache reuse can make the resulting edits cheaper.

Useful next questions

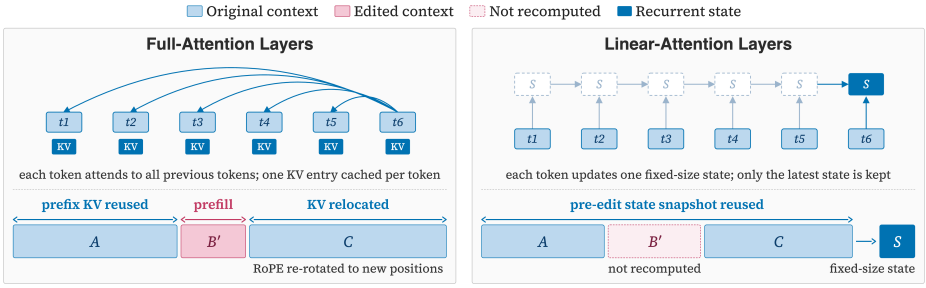
When should an agent pay a one-time cache penalty to save future work?

How quickly does reused state forget a corrected or deleted fact?

Can length awareness and reliable editing be learned together?

Synthesis of the paper. Future questions are presentation analysis. Paper: arXiv:2609.37725v1.

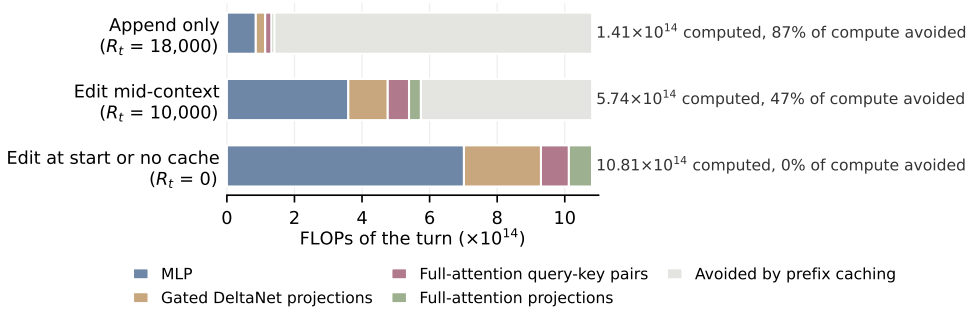
Backup: hybrid SCR carries a pre-edit recurrent state



- Qwen3.6-27B: 16 full-attention and 48 linear-attention layers.
- Full attention relocates KV; linear attention reuses a state snapshot.
- Edited history is not replayed through the linear recurrence.
- New text can still influence later layers through full-attention outputs.

Paper, App. B and Fig. 10, PDF p. 15.

Backup: prefix-reuse FLOPs are an explicit approximation



- Includes MLP/projections and full-attention pairwise operations.
- Omits embedding/output layers, Gated DeltaNet state updates, normalization, activations, and softmax.
- FLOPs, wall-clock time, and API spend answer different questions.

Paper, App. C, Eqs. (7)–(9), Fig. 14, PDF pp. 17–18. Lengths are illustrative.

Backup: 128K context changes the subagent tradeoff

Qwen3.6-27B, EdgeBench-10, 128K	Score ↑	PFLOPs/run ↓
Summary	47.8	222
CLM	47.3	**142**
CLM + subagents	**50.2**	219

- Single-agent CLM is cheaper but slightly lower scoring at 128K.
- Subagents improve score here, unlike the 32K comparison.
- Model scale also matters: on TB2.1, 9B edits 1.4 times/task vs. 2.6 for 27B.
- Median peak contexts: 30.2K vs. 17.6K under a nominal 32K limit.

Paper, App. F, Figs. 17/19, PDF pp. 24–25. EdgeBench scores use best-of-three.

Backup: detailed budgets qualify the shared-caption setup

Source / task	Context budget	Turn limit
Fig. 5 caption, all three tasks	32K	100
App. E, TB2.1 / TBLite	32,000	64
App. E, BrowseComp-Plus	23,560	100

- BCP excludes CLM editing turns; overflow can roll back up to 6 times.
- EdgeBench: text says up to 5 subagents; appendix says up to 6.
- Table 2 and zero-shot 9B BCP use different evaluation configurations.
- A reproduction should resolve these details before matching results.

Paper, Fig. 5, PDF p. 7; Sec. 5.1.2, p. 8; App. E, pp. 22–24.

Reference and discussion materials

Context Language Models

Rulin Shao, Shannon Zejiang Shen, Junjie Oscar Yin, Yuetai Li,
Minheng Wang, Hamish Ivison, Radha Poovendran, Nathan Lambert,
Teng Xiao, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, Pang Wei Koh.

arXiv:2609.37725v1, submitted 29 September 2026.

Paper: arxiv.org/abs/2609.37725v1

Author code link: facebookresearch/context-language-models

Figures are reproduced from the versioned paper and its author source.
Equations and selected tables are typeset for this reading presentation.
Companion file: `reading-notes.zh.md`.