PARTREP: Learning What to Repeat for Decoder-only LLMs

A learned selector for selective prompt repetition

Research group reading

Paper authors: Andikawati P Widjaja, Yongjun Kim, Hyounghun Kim, and Jaeho Lee

arXiv:2607.01792v1 | 2 July 2026
Reading prepared: 29 September 2026

Keep the original prompt; learn which tokens deserve a second appearance.

The main result is a quality–cost trade-off

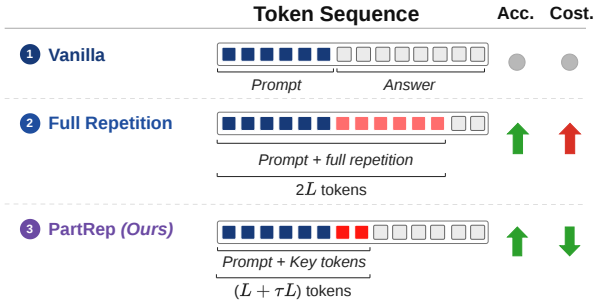
Qwen2.5-3B: seven-task average	
No repetition	66.9%
Full repetition	68.5%
PartRep	**68.3%**

Relative to full repetition	
59.5%	KV tokens
79.0%	prefill FLOPs

- Select only a fraction of the prompt, using a learned gate.
- The original prompt is retained in full.
- Savings are measured against full repetition; cost still rises relative to no repetition.

Paper, Table 1, PDF p.6; $\tau = 0.15$. Ratios recomputed from the displayed table. The abstract reports 59.4% KV, a small unexplained discrepancy.

A late copy can read the entire original prompt



- Early tokens cannot see later context.
- A repeated token has a new, later representation.
- That copy can attend to all of the first prompt.

Mechanistic intuition

Repetition creates additional information paths under the same causal mask. It does not retroactively update the original prefix.

Paper, Sections 1 and 3, PDF pp.1–3; original Figure 1. Figure arrows are conceptual, not per-task guarantees.

Choose a subsequence under an explicit budget

Let $\mathbf{t} = (t_1, \ldots, t_L)$ be the original prompt.

$$\mathbf{t}_{\mathrm{FR}} = \mathbf{t} \oplus \mathbf{t}, \qquad \mathbf{t}_{\mathrm{PR}} = \mathbf{t} \oplus f(\mathbf{t}).$$

$$\max_{f} \ \mathrm{acc}(\mathbf{t} \oplus f(\mathbf{t})) \quad \text{subject to} \quad |f(\mathbf{t})| \le \tau L.$$

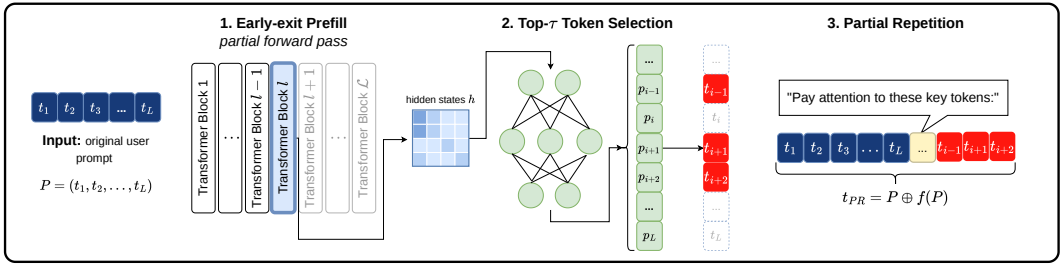
- $f(\mathbf{t})$ is an order-preserving subsequence.
- Default short-task budget: $\tau = 0.15$.
- The selector adds information cues; it does not remove original input.

A different objective from cache eviction

A token can be valuable to retain but provide little benefit when repeated. Evicting after a full repeat also retains the longer prefill cost.

Paper, Equations (2)–(5), Section 3, PDF p.3; Section 3.1, p.4. Connector and window expansion are omitted from this idealized length constraint.

Select during prefill, then append the chosen tokens



1. Extract original-prompt hidden states at layer $l^\star$.
2. Score positions and select the top-τ fraction.
3. Append selected tokens; reuse prefix computation and finish prefill.

Paper, original Figure 2, PDF p.4; Sections 4.2 and 4.4, pp.5–6. Qwen layer-choice ablation selects layer 18.

High NLL supplies labels for what to repeat

$$\mathrm{NLL}(t_i) = -\log P(t_i \mid t_{<i}).$$

- High NLL: the target model finds a token hard to predict from its prefix.
- Training positives: positions in the highest-NLL τ fraction of each sequence.
- At inference, a small gate predicts membership in that group.

What this signal does and does not establish

Surprisal is an empirical proxy for repetition value. It is not a direct estimate of a token's causal effect on answer accuracy.

Paper, Equations (6)–(8), Section 4.1, PDF pp.4–5. Top-τ means the upper tail; the paper's τ-quantile wording is ambiguous.

The gate reads learned prototypes, one token at a time

$$\mathbf{q} = W_2\ \mathrm{SiLU}(W_1\mathbf{h}), \qquad p = g(\mathbf{h}) = \sigma\left[\mathbf{v}^\top\ \mathrm{Softmax}\left(\frac{\mathbf{q}K^\top}{\sqrt{d_h}}\right)\right].$$

- $\mathbf{h} \in \mathbb{R}^d$: base-model hidden state.
- W_1: $d \to 2d_h$.
- W_2: $2d_h \to d_h$.
- $d_h = 64$.
- $K \in \mathbb{R}^{s \times d_h}$: learned keys.
- $\mathbf{v} \in \mathbb{R}^s$: learned values.
- $s = 64$ by default.
- p: one score per input position.

No additional $L \times L$ input self-attention in the gate.

Paper, Equations (9)–(11), PDF p.5; Table 7, p.13. Query orientation follows the paper's row-vector attention notation.

Train the gate; keep the base language model frozen

$$\ell(\mathbf{t}) = \frac{1}{L}\sum_{j=1}^{L} \lambda_j \, \ell_{\mathrm{CE}}(p_j, y_j), \qquad \lambda_j = \begin{cases} (1-\tau)/\tau, & y_j = 1, \\ 1, & y_j = 0. \end{cases}$$

- Store hidden-state/NLL pairs offline; derive binary labels for τ.
- 3M training tokens: 2M FineWeb-Edu + 1M Ultra-FineWeb-Edu.
- Sequence lengths: 10–1024 tokens.
- Adam, learning rate 10^{-3}, batch size 4096, 25 epochs; cosine schedule and early stopping with patience 5.

Cost is shifted offline

Each target LLM needs its own gate. No cross-model zero-shot transfer is shown.

Paper, Equations (12)–(15), PDF pp.5–6; Appendix A.1, p.11; Limitations, p.9. Validation split and batch unit are not specified.

The final repeated span depends on formatting and windows

Online input construction

Original prompt + connector + selected tokens in original order + any required answer formatting.

- Default connector: *Pay attention to these key tokens:*
- Optional expansion restores a whole word or neighboring words.
- Fragmented subwords and missing local relations motivate expansion.
- Causal prefix computation can be reused when prefix positions stay fixed.

$$N = L + m + w + c, \qquad m \leq \tau L,$$

where w counts extra window tokens and c counts connector/formatting tokens.

Paper, Section 4.4, PDF p.6; Appendix C.3, p.13. Length decomposition is a reader explanation. Detailed cache scheduling and post-expansion budget rules are not reported.

The evidence spans different experimental settings

Experiment	Model(s)	Setting
Seven-task main result	Qwen2.5-3B	$\tau = 0.15$
Cross-model MMLU	Qwen, Llama, Gemma	Three model families
RULER	Qwen2.5-3B	$\tau = 0.3$ + windows
Window ablation	Llama3.2-3B	ARC-Challenge
Other ablations	Qwen2.5-3B	ARC-Challenge

- Accuracy ↑; stored prompt KV tokens ↓; reported prefill FLOPs ↓.
- Greedy decoding, one run, bfloat16; no error bars.
- RULER: 100 examples per subtask, 1300 total.
- Eight benchmarks across the study; not every model on every benchmark.

Paper, Section 5.1, PDF pp.6–7; Sections 5.3–6.1, pp.7–8; Appendix A.2, p.11. Model names: Qwen2.5-3B Instruct, Llama3.2-3B Instruct, Gemma4-E4B it.

PartRep approaches full-repeat quality at lower cost

Method	Avg. accuracy	KV tokens	Prefill FLOPs
No repetition	66.9	235.5	1.549 T
Append model summary	67.9	277.4	3.676 T
Append LLMLingua summary	67.4	277.4	2.105 T
Full repetition	68.5	471.1	3.140 T
PartRep	**68.3**	**280.4**	**2.481 T**

- $+1.4$ percentage points over no repetition.
- -0.2 percentage points versus full repetition.
- More accurate than appended LLMLingua, but also more compute.
- About 87.5% of the full-repeat gain, using rounded reported means.

Reorganized from paper, Table 1, PDF p.6. Qwen2.5-3B, seven benchmarks, $\tau = 0.15$. T $= 10^{12}$ FLOPs; derived gain fraction is descriptive, not a confidence estimate.

The average hides meaningful task-level differences

Benchmark	No repetition	Full repetition	PartRep
ARC-Challenge	79.8	82.8	81.4
OpenBookQA	76.6	78.0	77.8
MMLU	62.5	60.3	63.2
MedQA	47.5	51.0	50.3
SciQ	90.3	92.9	92.2
MMLU-Pro	27.8	27.3	27.8
GSM8K	84.2	87.3	85.5

- PartRep improves six tasks over no repetition; one is tied.
- It exceeds full repetition on two tasks and trails on five.
- Full repetition itself can hurt: MMLU and MMLU-Pro.

Reorganized from paper, Table 1, PDF p.6; accuracy in percent. Claims of universal task-wise superiority would overstate these data.

Separate ideal attention scaling from reported total FLOPs

Ignoring selection, formatting and expansion:

$$\frac{M_{\mathrm{PR}}}{M_{\mathrm{FR}}} \approx \frac{1+\tau}{2}, \qquad \frac{F_{\mathrm{attn,PR}}}{F_{\mathrm{attn,FR}}} \approx \frac{(1+\tau)^2}{4}.$$

At $\tau = 0.15$	KV ratio	Compute ratio
Ideal length-based calculation	57.5%	33.1% (attention only)
Table 1: PartRep / full repeat	59.5%	79.0% (total prefill)

- Total compute also includes linear layers and method overhead.
- Versus no repetition: KV +19.1%, prefill FLOPs +60.2%.
- FLOPs and KV token counts are not latency and peak GPU memory.

Paper, Section 3, PDF p.3; Table 1, p.6. Ratios derived from displayed values. Abstract KV claim: 59.4%; table recomputation: 59.52%.

MMLU improves over vanilla across three model families

Method	Qwen2.5-3B	Llama3.2-3B	Gemma4-E4B
No repetition	62.5	54.1	74.0
Full repetition	60.3	54.8	72.9
Append LLMLingua	61.0	53.5	76.1
PartRep	63.2	54.3	75.8

- Gains over vanilla: $+0.7$, $+0.2$, $+1.8$ percentage points.
- PartRep is not the best method in every model column.
- Gemma features must come from a global-attention layer.
- Each model has its own trained gate; this is not gate transfer.

Reorganized from paper, Table 2 and Section 5.3, PDF p.7. Gemma extraction restriction: Section 5.1, p.6; no exact Gemma layer number given. Single-run differences have no reported uncertainty.

RULER gains use a larger budget and local windows

Method	4k	8k	16k
No repetition	77.7	80.0	72.3
Full repetition	79.2	82.3	72.3
LLMLingua	69.2	73.1	63.5
PartRep	**80.0**	**83.8**	**73.2**

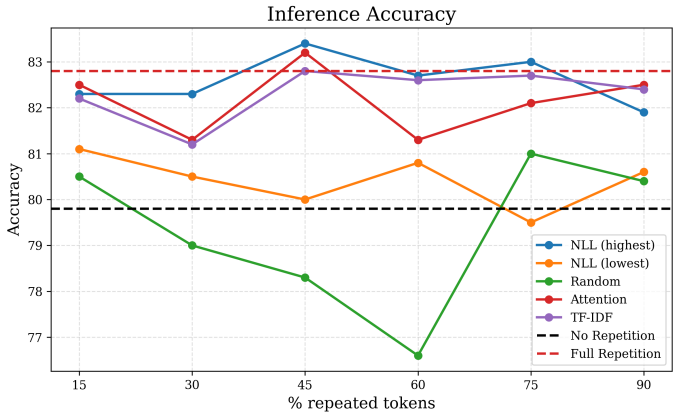
Qwen2.5-3B: $\tau = 0.3$ + local-context expansion

Gains over full repetition: $+0.8$, $+1.5$, $+0.9$ percentage points.

- Actual post-expansion cache size and FLOPs are not in this table.
- Do not transfer the short-task cost ratios to these results.
- The advantage over full repetition is not monotonic in length.

Reorganized from paper, Table 3 and Section 5.4, PDF pp.7–8. LLMLingua label follows Table 3; its precise variant is not clarified there.

High NLL is useful, but does not win at every budget



- Highest NLL generally beats low NLL and random.
- Attention is higher at 15% and 90%.
- This compares scoring signals, not just learned-gate quality.

Paper, original Figure 4, Appendix B.1, PDF p.11. Qwen2.5-3B on ARC-Challenge. Curves are single-run results without uncertainty.

Window expansion repairs cues at an additional token cost

Expansion	ARC accuracy (%)	KV tokens
No windowing	61.5	90.1
Complete the word	64.6	94.5
Add one neighboring word on each side	66.0	110.2

Llama3.2-3B on ARC-Challenge

The widest tested window adds 4.5 percentage points and about 22.3% KV tokens relative to no expansion.

- Lexical integrity and local relations matter.
- The selection ratio alone does not determine the final budget.

Reorganized from paper, Table 5, Section 6.1, PDF p.8. KV growth is derived from displayed values. This experiment uses Llama, not Qwen.

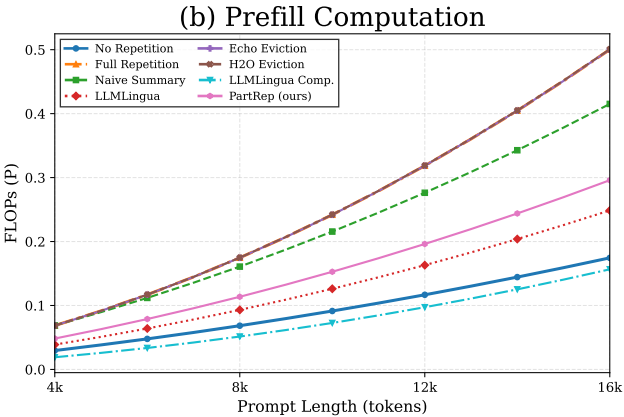
The connector is part of the intervention

Connector style	ARC accuracy (%)
Natural-language cue (default)	**81.4**
ChatML	70.6
Natural language + ChatML	79.5
Raw concatenation	79.4
XML	71.6
Technical natural-language cue	80.9

- The default cue exceeds raw concatenation by 2.0 percentage points.
- Benefits cannot be attributed to token selection alone.
- Wrapper effects may depend on model/template interpretation; the cause is not isolated.

Values from the labels in paper Figure 7, Appendix B.4, PDF p.12; templates in Table 6, p.11. Qwen2.5-3B on ARC-Challenge.

Estimated prefill cost stays below full repetition



- Dense attention remains $O(L^2)$.
- KV remains $O(L)$ at fixed τ (panel a).
- FLOP savings do not establish speedups.

Paper, Figure 3(b), Section 6.2, PDF p.8; author vector panel, all axes and legends retained. P = 10^{15} FLOPs. Panel (a), reproduced in the reading notes, gives the KV scaling comparison.

The deployment case needs several missing measurements

- **Per-model training:** a new target LLM requires a new gate.
- **Statistical scope:** single runs; small gains lack uncertainty.
- **System evidence:** no full-method TTFT, throughput, or peak-memory comparison with a common backend.
- **Reproducibility:** cache scheduling, expanded budgets, prompt protocols, and exact FLOP accounting are incompletely specified.
- **Generalization:** long real-world documents, larger models, and multimodal tokens remain open.

A focused follow-up

Fix the final appended-token budget and connector, then compare selectors on both accuracy and measured end-to-end cost.

Authors' limitations: PDF p.9. Evaluation protocol: Appendix A, p.11. Remaining points and proposed follow-up are reader analysis.

A compact second look is promising under the tested conditions

1. **Core idea:** learn which input tokens benefit from another late-context representation.
2. **Engineering choice:** distill high-NLL membership into a small gate using intermediate hidden states.
3. **Observed trade-off:** near full-repeat average accuracy with fewer cached tokens and lower reported prefill FLOPs.
4. **Qualification:** task, model, connector and window settings change the outcome; runtime benefits still need direct validation.

Synthesis of paper Sections 4–6, PDF pp.4–8, and Appendices A–B, pp.11–13. This presentation analyzes the paper; it does not report an independent reproduction.

Primary source and reading scope

Andikawati P Widjaja, Yongjun Kim, Hyounghun Kim, Jaeho Lee.

PARTREP: Learning What to Repeat for Decoder-only LLMs.

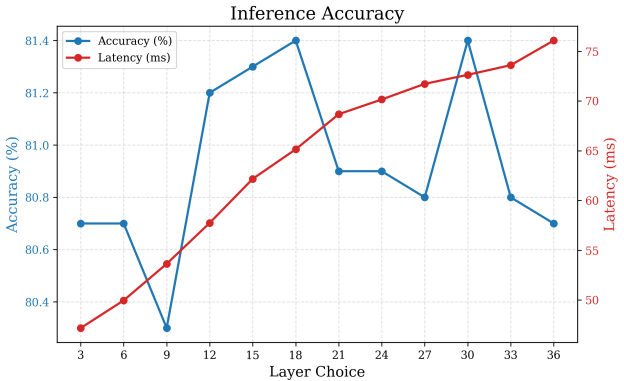
arXiv:2607.01792v1, submitted 2 July 2026.

arxiv.org/abs/2607.01792v1

- Read the author TeX, all substantive sections and Appendices A–E.
- Checked equations, seven figures and seven tables against the 15-page PDF.
- Original figures are reused; reorganized tables retain source locators.
- Ratios and implementation cautions are explicitly marked as reader analysis.

Materials retrieved 29 September 2026. Main talk: slides 1–22, approximately 20–30 minutes. Remaining slides provide optional discussion details.

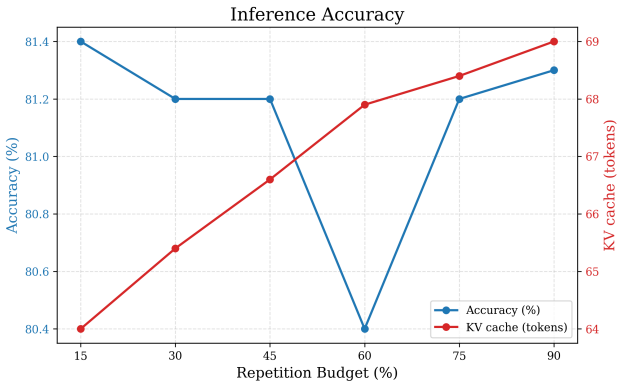
Backup: deeper features do not monotonically improve accuracy



- Layers 18 and 30 both reach about 81.4%.
- Layer 18 has lower latency.
- Timing scope is not fully specified.

Paper, original Figure 5, Appendix B.2, PDF p.12. Qwen2.5-3B on ARC-Challenge; this does not establish a universal layer choice.

Backup: increasing the repetition budget can reduce accuracy



- Default: $\tau = 0.15$.
- Accuracy is non-monotonic.
- The 64–69-token right-axis accounting is not explained.

Paper, original Figure 6, Appendix B.3, PDF p.12. Qwen2.5-3B on ARC-Challenge. Author explanation about cue dilution is a hypothesis.

Backup: a small gate is a reasonable tested operating point

Architecture

Gate	Acc. (%)	FLOPs
Linear	80.7	270.6 k
Two-layer MLP	81.4	549.0 k
Transformer	81.3	654.3 k

Learnable keys

s	Acc. (%)	FLOPs
32	80.1	544.8 k
64	81.4	549.0 k
128	80.6	557.3 k

- The reported two-layer configuration includes learned-key attention.
- Larger gates do not improve these single-run results.
- Gate FLOP normalization is unspecified; do not directly divide these values by the main-table request cost.

Reorganized from paper, Table 4, PDF p.8, and Table 7, p.13. Qwen2.5-3B on ARC-Challenge. No repetition: 79.8%.

Backup: eviction saves cache after paying for a full repeat

Method	Avg. acc. (%)	KV tokens	Prefill FLOPs
Full repetition	68.5	471.1	3.140 T
+ Echo eviction	64.4	235.5	3.152 T
+ H2O eviction	62.9	270.8	3.162 T
LLMLingua compression	14.9	78.6	1.130 T
PartRep	68.3	280.4	2.481 T

- Echo removes first-copy KV; H2O retains selected heavy hitters.
- LLMLingua compression feeds only the compressed repeated input.
- Its poor score is specific to this setup, not a general verdict.
- Echo/H2O use manual KV decoding; other methods use vLLM.

Reorganized from paper, Table 1, PDF p.6; Appendix A.2, p.11, and C.6–C.8, pp.13–14. Qwen2.5-3B, seven tasks.

Backup: notation does not supply an optimality theorem

The paper motivates selection using

$$\max_{f} H(f(\mathbf{t}) \mid \mathbf{t}).$$

- For ordinary random variables and deterministic f, $H(f(X) \mid X) = 0$.
- The paper acknowledges this and uses a model-based information notion.
- It gives no approximation guarantee linking NLL ranking to accuracy.
- The selector notation f alternates between scores and a subsequence.

Useful operational interpretation

Frozen hidden states $\rightarrow$ per-position scores $\rightarrow$ top-k positions $\rightarrow$ restore order $\rightarrow$ optionally expand $\rightarrow$ append.

Paper, Equations (4), (7)–(9), PDF pp.3–5, especially the footnote on p.5. Operational decomposition and caution are reader analysis.