

Peer Review

Review of: "Towards More Economical Context-Augmented LLM Generation by Reusing Stored KV Cache"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

This poster-length preliminary paper asks whether the well-known latency benefit of reusing precomputed KV caches for repeated long contexts also survives a purely economic test on public cloud infrastructure. The authors argue that, instead of keeping context KV caches resident in GPU memory (which does not scale to a large corpus of contexts) or recomputing the prefill on every request, a developer can park the KV cache on cloud storage and stream it into GPU memory on demand. To support this, **Section 2** introduces two analytical cost expressions: an unnumbered equation for text recomputation, $C_{text} = C_{GPU} \times N[T_{prefill}(L_{context} + L_{prompt}) + T_{decode}(L_{output})]$, and a second unnumbered equation for the reuse pipeline, C_{KV} , decomposed into a GPU-inference term, a storage term $C_{storage} \times S_{storage}(L_{context}) \times T$, and a transmission term $C_{transmission}(S_{storage}(L_{context}), SLO)$. Instantiated with AWS EBS io2 pricing and a V100 hourly rate (**Section 2**, “**Insights**”, and **Footnotes 1–2**), the model is claimed to show that reuse pays off “as long as the context is reused more than once per hour,” and that storage is a negligible fraction of total cost. **Section 3** reports a Llama-7B run on an EC2 p3.8xlarge instance over 200 TriviaQA contexts, each assumed reused five times, with savings quoted as 1.1–2.9× delay and 1.3–3.6× cost when sweeping input length (**Figure 2(a)**) and 1.6–3.5× delay and 1.7–4.5× cost when sweeping output length (**Figure 2(b)**).

Note on the dump reviewed. The version I was given is the Qeios page text, and it contains no figure images: **Figure 1** (“Illustration of text recomputation and KV cache reusing”) and **Figure 2** (“The cost and end-to-end delay for Llama-7B by varying input lengths”, with panels (a) and (b) referenced in **Section 3**, **Takeaways**) are present only as captions. The paper contains *no tables at all*, so every quantitative claim

in **Section 3** rests on figure panels I could not inspect; my comments on those numbers are therefore limited to what the prose asserts. The two display equations in **Section 2** and the simplified ratio at the end of that section are unnumbered, which makes precise reference awkward; I refer to them below as Eq. (A) (C_{text}), Eq. (B) (C_{KV}) and Eq. (C) (the C_{text}/C_{KV} ratio).

2. Whether the conclusion holds

The framing is genuinely useful — the systems literature on KV reuse (cited here as [2] PagedAttention and [3] ChunkAttention) has been almost entirely latency-driven, and asking the cost question is the right move. But as presented, the central claim is not established, for four reasons that compound.

(a) The simplified ratio structurally cannot produce a break-even. Eq. (C), $C_{text}/C_{KV} \approx 1 + [(N-1)/N] \times T_{prefill}(L_{context}) / [T_{decode}(L_{output}) + T_{prefill}(L_{prompt})]$, is the only closed form offered, and it drops both the storage term and the transmission term that Eq. (B) had just introduced. With those terms gone, the ratio is ≥ 1 for every $N \geq 1$ and every parameter setting: the approximation makes reuse weakly favorable by *construction*. The paper’s headline conclusion — that reuse is economical — is thus baked into the algebra rather than derived from it. Correspondingly, the break-even statement in **Section 2**, “**Insights**” (“more economical as long as the context is reused more than once per hour”) appears nowhere as a derivation: no step sets Eq. (A) equal to Eq. (B) and solves for N or for a reuse rate, and the serving period T — the only place where “per hour” could enter — is one of the symbols Eq. (C) discards. A reader cannot check the threshold and cannot see under which ($L_{context}$, L_{output} , $C_{storage}/C_{GPU}$) combinations it would move.

(b) The price inputs are hard-coded, undated, unregioned, and partly unsourced. Two prices carry the entire numerical argument. The first is in **Section 2**, “**Insights**”: io2 “costs 0.125 per GB $\times$ month [4]” — given without a currency symbol, without an AWS region, and without a retrieval date; reference [4] is a bare URL to the EBS pricing page with no access date, and that page is mutable by definition. The second is in **Footnote 2**: “Single V100 GPU cost per hour ...”, which carries *no citation at all* — no instance family, no region, no purchase model (on-demand vs. reserved vs. spot), no date. For a paper whose thesis is a price comparison, this is the reproducibility fault that matters most: a cost model whose inputs have no snapshot date is a model that cannot be re-evaluated, re-checked, or refuted a year later, and the ratio $C_{storage}/C_{GPU}$ is exactly the quantity that GPU price movements alter fastest. The article carries a publication date of 9 April 2025, but an implicit date is not a snapshot.

(c) The one worked numerical comparison does not reconcile. **Footnote 1** computes storage as $0.00017 \times 5.2 = 0.00088$ — i.e., the io2 monthly rate prorated to an hour, times a 5.2 GB cache. **Footnote 2** computes

prefill as $\times (1 \text{ h} / 3600 \text{ s}) \times T_{\text{prefill}}(10\text{K}) = 3 \times (1/3600) \times 7 = 0.0058$. The body text then states these are “.8e-4” and “around e-3” and concludes “This is already more than 7 times larger.” The actual ratio is $0.0058 / 0.00088 = 6.59\times$; even using the rounded e-3, it is $6.8\times$. The claim “more than 7 times” is therefore not supported by the authors’ own footnotes. Separately, the body text calls .8e-4 “the total storage and transmission cost,” but **Footnote 1** computes storage only — the transmission term of Eq. (B) is silently set to zero in the one place the model is instantiated, and the SLO argument it takes is never assigned a value anywhere in the paper. The third term of the cost model is thus never actually exercised. Finally, $T_{\text{prefill}}(10\text{K}) = 7 \text{ s}$ in **Footnote 2** is asserted with no measurement, batch size, precision, or attention-implementation context.

(d) **“Validated” is used in three incompatible ways, and no validation is shown.** The **Abstract** promises “an validated analytical model”; **Section 1** says “we validate this result by simulation under various workloads”; **Section 3, Evaluation setup** describes a real hardware run (“we run Llama-7B model on an Amazon’s EC2 p3.8xlarge instance (4 V100 GPUs)”). Simulation and measurement are different evidentiary claims, and the paper never states which one **Figure 2** reports. More importantly, validation of a cost model means plotting model-predicted cost against realized cost and reporting the residual. Nothing in **Section 3** does this: the takeaways report reuse-vs-recompute ratios, not model-vs-reality error. On the evidence presented, Eq. (A)–(C) are self-consistent, not validated.

There is also an internal inconsistency in the cost basis that may be worth a factor of four. **Section 2** defines C_{GPU} as “the cost of the GPU *instance* per hour,” but **Footnote 2** substitutes a “single V100 GPU cost per hour,” while **Section 3** runs HuggingFace model parallelism across all four V100s of a p3.8xlarge. If the workload occupies the whole instance, the billed rate is the instance rate, not one GPU’s share, and the prefill side of the comparison is undercounted relative to the storage side — which would make the “7×” (really 6.59×) gap larger, but only by an amount the paper has not audited.

Two modeling choices in **Section 2, “Insights”** deserve explicit defense rather than a passing sentence. First, EBS io2 is a block device attached to a single instance; it is not obviously the right primitive for the paper’s own motivating scenario in **Section 1**, a “dedicated storage server” serving KV caches to a fleet of GPU workers, where object storage or a shared filesystem plus cross-AZ network egress would apply — and egress is precisely the transmission cost Eq. (B) leaves unpriced. Second, the sentence “Since our IO operations are not frequent, we do not need to pay much extra provisioned IOPS fee” appears to treat provisioned IOPS as usage-metered; on io2 the IOPS charge is for *provisioned* capacity per month, independent of how many operations are issued, and the volume must be provisioned for the peak

throughput the paper wants (it claims “the highest tier storage with 4GB/S throughput”). If so, the storage term of Eq. (B) is understated, and by more than a rounding margin. The attainability of 4 GB/s from a p3.8xlarge specifically — instance family, volume type, and Block Express support — also needs to be stated rather than assumed.

Verdict on the conclusion. The qualitative direction is plausible and is consistent with the structure of Eq. (A) versus Eq. (B): when $L_{\text{context}} \gg L_{\text{prompt}} + L_{\text{output}}$, amortizing one prefill over N requests should win. What does *not* hold as stated is the specific, quotable claim — the “once per hour” break-even — because it is asserted rather than derived, rests on two undated hard-coded prices, uses a simplified ratio from which the adverse cost terms have been removed, and is accompanied by an arithmetic slip in the single worked example.

3. Reproducibility

Assessed as a preliminary poster, not a full paper — the text itself says “We hope this poster sparks more discussions” (Section 1), even though Qeios lists it as a Research Article. Even at poster length, several items are cheap to supply and currently missing.

Prices. No snapshot date for either price. The io2 figure in Section 2, “Insights” cites [4], a dateless URL; the /h GPU figure in Footnote 2 has no source. No region, no currency symbol on the io2 number, no purchase model. An independent reader in 2026 cannot reconstruct either input.

Model and cache size. “Llama-7B” (Section 2, Insights; Section 3) is given without a version (Llama-1 vs. Llama-2 7B), and the 5.2 GB figure for a 10K-token cache is stated without the formula that produces it — layer count, head count, head dimension, and above all the KV dtype. 5.2 GB is what one gets for a 32-layer, 4096-hidden model at 16-bit KV ($2 \times 32 \times 4096 \times 2 \text{ B} \times 10,000 \approx 5.24 \text{ GB}$), but the reader should not have to reverse-engineer this, and the number moves by $2\times$ or $4\times$ under fp8/int4 KV quantisation — a compression axis the Abstract itself names as part of the design space.

Hardware and serving stack. Section 3 gives instance type (p3.8xlarge, 4×V100) and “HuggingFace’s model parallelism,” which is helpful, but omits region, batch size, sequence-parallel vs. tensor-parallel details, attention kernel, precision, framework versions, and whether the KV cache was served from EBS io2 during the runs or if the transfer time was modelled.

Workload parameters. Section 3 fixes $N = 5$ (“each context is used five times”) over 200 TriviaQA contexts, and sweeps input length 1–10K and output length 1–100 tokens. But N is the parameter the

break-even claim is about, and it is never swept; nor is the serving period T , nor the arrival rate that converts N into “reuses per hour.”

Code and data. No repository, no artifact link, no release of the cost-model script or the measurement harness. The dataset is public ([1], TriviaQA), but the selection of 200 contexts and their length distribution is not specified.

Measured validation. None reported, in the sense defined above (predicted vs. observed cost). **Figure 2**, in any case, is absent from the dump I reviewed.

4. What specifically to fix

1. **Number the equations.** Section 2 contains three unnumbered expressions (my Eq. (A) C_{text} , Eq. (B) C_{KV} , Eq. (C) the C_{text}/C_{KV} ratio). Number them (1)–(3); the footnote markers (1) and (2) currently collide visually with equation numbering.
2. **Fix the symbol collision in Eq. (B).** The variable list after Eq. (B) defines $C_{storage}$ twice: first as “the storage cost per hour $\times$ GB,” then as “the cost per hour to maintain the transmission link that satisfies SLO.” The second is presumably the rate inside $C_{transmission}$; give it its own symbol (e.g. C_{link}).
3. **Define the bare $T_{prefill}$ term in Eq. (B).** The GPU-inference brace reads $N[T_{decode}(L_{output}) + T_{prefill}(L_{prompt})] + T_{prefill}$, where the trailing $T_{prefill}$ has no argument. State explicitly that this is the one-off cost of producing the stored cache, i.e. $T_{prefill}(L_{context})$ incurred once per context, and say whether it is amortized over one serving period T or over the cache’s whole lifetime.
4. **Derive the break-even instead of asserting it.** Section 2, “Insights” claims reuse wins “as long as the context is reused more than once per hour.” Set Eq. (A) = Eq. (B) and solve for the reuse rate N/T ; report the closed form, the parameter values assumed, and how the threshold shifts with $L_{context}$, L_{output} , and $C_{storage}/C_{GPU}$. As it stands, the threshold cannot be checked.
5. **State that Eq. (C) is an upper-bound-favorable approximation, or repair it.** Eq. (C) omits the storage and transmission terms of Eq. (B) and is therefore ≥ 1 unconditionally, which makes the paper’s conclusion true by construction. Either retain both terms in the ratio or add an explicit sentence stating what was dropped and bounding the error.
6. **Correct the ratio in Section 2, “Insights”.** “This is already more than 7 times larger than the fixed cost of storage” does not follow from **Footnotes 1 and 2**: $0.0058 / 0.00088 = 6.59\times$ ($6.8\times$ with the

rounded $e-3$ quoted in the body). Use one consistent value for the prefill cost in the body and footnote, and restate the factor.

7. **Reconcile the “storage and transmission” label with Footnote 1.** The body calls $.8e-4$ “the total storage and transmission cost,” but **Footnote 1** computes $0.00017 \times 5.2 = 0.00088$, which is storage alone. Either price the transmission term of Eq. (B) explicitly (with the link bandwidth and the SLO value used) or relabel the quantity as storage-only and say the transmission term is deferred.
8. **Give every price a snapshot date, region, and source.** For the io2 rate in **Section 2, “Insights”** (“0.125 per GB $\times$ month [4]”), add currency, AWS region, and an access date on reference [4]. For the “V100 hourly rate in **Footnote 2**, add a citation, region, purchase model, and date — it currently has no source of any kind. A reproducibility table listing every price input, its value, its region, its source, and its retrieval date would resolve items 8 and 9 at once.
9. **Resolve the C_{GPU} basis mismatch.** **Section 2** defines C_{GPU} as the per-hour cost of the GPU *instance*; **Footnote 2** uses a *single* V100 rate; **Section 3** runs across all 4 GPUs of a p3.8xlarge. State which basis the numbers use, and if the instance is occupied wholly, bill the instance rate.
10. **Justify the provisioned-IOPS assumption in Section 2, “Insights”.** “Since our IO operations are not frequent, we do not need to pay much extra provisioned IOPS fee” reads as though IOPS were usage-metered on io2. If the volume must be provisioned to sustain the claimed 4 GB/s, that provisioned capacity is billed whether or not it is exercised; include it in the storage term of Eq. (B) or explain why it does not apply.
11. **Justify EBS io2 as the storage primitive.** **Section 1** motivates a “dedicated storage server” serving many GPU workers, but **Section 2** prices single-attach block storage. Either restrict the claim to single-instance caching or add an object-storage / shared-filesystem variant with cross-AZ egress priced into the transmission term, which is where a fleet deployment would pay.
12. **Show the cache-size derivation.** “A context of length 10K-token requires a storage space of 5.2 GB” (**Section 2, “Insights”**) needs the formula, the Llama-7B configuration used, and above all the KV dtype. Then report how the break-even of item 4 moves under fp8 or int4 KV quantization — the **Abstract** names compression as part of the design space but no result addresses it.
13. **Add an actual validation of the model.** The **Abstract** (“an validated analytical model”), **Section 1** (“we validate this result by simulation”), and **Section 3** (a hardware run on p3.8xlarge) describe three different things. Pick one, and add a panel or table plotting Eq. (A)/(B) predictions against measured dollar cost with the residual reported. Without it, “validated” should be removed from the Abstract.

14. **Sweep N. Section 3, Evaluation setup** fixes “each context is used five times.” Since the paper’s claim is a threshold in reuse frequency, add a sweep over N (and over the arrival rate that converts N into reuses per hour), and mark the break-even point on it.
15. **Supply the missing figures and their axes.** In the version I reviewed, **Figure 1** and **Figure 2(a)/2(b)** are captions only, with no images, and the paper contains no tables. Please confirm the published record renders them, label the axes with units and currency, and state whether **Figure 2** reports simulated or measured cost. The caption of **Figure 2** mentions only “varying input lengths” while **Section 3, Takeaways** bullet 2 attributes output-length sweeps to **Figure 2(b)** — the caption should cover both panels.
16. **Report absolute costs alongside ratios.** **Section 3, Takeaways** gives savings as 1.1–2.9× and 1.3–3.6× (**Figure 2(a)**) and 1.6–3.5× and 1.7–4.5× (**Figure 2(b)**). A table of dollars per 1,000 requests at each sweep point would let readers re-derive the ratios under their own prices, which is the whole point of an analytical cost model.
17. **Fix the citation of [1].** **Section 1** states “Prior work found that different user requests may need to use the same context[1],” but [1] is the TriviaQA dataset paper (Joshi et al., 2017), which does not make a context-reuse observation. Cite a workload-characterization source for that claim and keep [1] for the dataset used in **Section 3**.
18. **Update reference [3].** It is listed as “Anonymous. ChunkAttention ... openreview.net”; this work has since appeared non-anonymously and should be cited in its published form.
19. **Settle the paper’s genre and tidy the prose.** **Section 1** ends “We hope this poster sparks more discussions,” while the venue lists a Research Article; choose one. Also: the **Abstract** has “an validated analytical model”; **Section 3, Takeaways** bullet 1 says “shorter inputs (like 1,000 units)” where the unit is tokens, and the same sentence says the saved prefill time “does not compromise the delay it takes to load the KV cache,” which appears to mean “does not compensate for.”

5. Recommendation

Major revisions. The question the paper poses — whether latency-motivated KV cache reuse is also cost-motivated on public cloud — is well chosen and under-studied, and the two-equation decomposition in **Section 2** is a reasonable skeleton for answering it. But the headline claim in **Section 2, “Insights”** (break-even at “more than once per hour”) is asserted rather than derived; the simplified ratio Eq. (C) drops exactly the storage and transmission terms that could falsify it; the single worked example in **Footnotes 1–2** yields 6.59× where the text claims “more than 7 times” and quietly prices the transmission

term at zero; and both price inputs are hard-coded without date, region, or (for the /h GPU rate) any source. “Validated” in the **Abstract** is not supported by anything in **Section 3**, which reports reuse-vs-recompute ratios rather than model-vs-measurement error. I judge all of this fixable: items 4, 5, 6, 7, 8, 13, and 14 above are the load-bearing ones, and a revision that derives the threshold, dates its prices, and shows one predicted-vs-measured comparison would make this a useful reference point for practitioners. I would be glad to look at a revised version.

Declarations

Potential competing interests: No potential competing interests to declare.