EVALUATING THE ACCURACY OF KV CACHE REUSE TECHNIQUES

Anonymous authors
Paper under double-blind review

ABSTRACT

Position-independent KV cache reuse aims to reduce latency in retrieval-augmented generation by reusing chunk-level KV caches across prompts. We show that current evaluations of KV cache reuse techniques rely on measurements that fail to faithfully capture the loss of accuracy attributable to reuse, often artificially inflating the reported effectiveness. We also show that existing datasets do not exhibit the reuse dynamics needed to thoroughly evaluate such techniques. To address these issues, we propose an evaluation methodology that measures this accuracy loss without ambiguity and we introduce `Boxoffice`, a tool that programmatically generates evaluation datasets that exercise challenging KV cache reuse patterns.

1 INTRODUCTION

Retrieval-augmented generation (RAG) systems augment user queries with retrieved text chunks, supplying large language models (LLMs) with up-to-date and domain-specific context that improves answer accuracy and reduces hallucinations (Lewis et al., 2020; Ray et al., 2025; Hu et al., 2025). The additional text chunks, however, inflate the prompt. Because attention cost is quadratic in prompt length, the resulting prefill stage, where the Key-Value (KV) caches are computed, becomes the dominant component of latency (Vaswani et al., 2017). A promising direction for mitigating this overhead is position-independent KV cache reuse (Yao et al., 2025; Hu et al., 2024; Agarwal et al., 2025; Wang et al., 2026a): each chunk's KV cache is computed once, stored, and subsequently reused at any position within future prompts that contain it. Position-independent KV cache reuse is strictly more general than prefix caching, the current industry standard (Kwon et al., 2023; Zheng et al., 2024b), which amortizes prefill cost only when chunks recur as the same ordered prefix.

The main challenge in reusing KV caches at arbitrary positions is the absence of cross-chunk attention: tokens in one chunk carry no attention to tokens in other chunks. This is critical for accuracy: naive reuse of independently-prefilled chunks leads to severe accuracy degradation (Yao et al., 2025; Gim et al., 2024). The widespread solution to recover cross-chunk attention is to recompute the KV cache for few tokens at query time. To preserve the performance benefits of KV cache reuse, such recomputation has to involve as few tokens as possible. An emerging line of work, which we refer to as *warm* approaches (Agarwal et al., 2025; Wang et al., 2026a; Liu et al., 2025), aims to reduce the recomputation cost by embedding cross-chunk attention into the saved KV already at store time, so that less of it needs to be reconstructed at query time.

Pitfalls of KV cache reuse evaluations. We identify two issues in the evaluation methodology adopted in the literature on KV cache reuse approaches.

The first issue is methodological. The headline metric for evaluating KV cache reuse should measure its core objective: preserving full prefill accuracy on queries where the correct answer genuinely depends on the provided context. However, existing work evaluates performance by reporting an aggregate accuracy metric, such as F_1 score, computed over every query in a dataset. This includes queries the baseline cannot answer even with full prefill (so there is no accuracy to preserve), queries the model can answer from parametric memory alone (so the KV cache is not what produces the answer), and yes/no or disjunctive questions prone to lucky guesses. Including such queries conflates accidental regularization brought by approximated attention states, parametric leakage, and chance correctness into the accuracy metric; as we show in Section 4.1, this systematically inflates the reported accuracy of reuse techniques, masking the true loss of accuracy attributable to the approximation.

The second issue lies in the chunk and query distributions of the datasets themselves. We identify two structural properties of commonly used datasets that systematically mislead, or even prevent, the

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.

evaluation of emerging warm approaches. The first property is *clustered queries*. The chunks in the context of a query tend to be semantically very similar, because they concentrate on the same topic, and chunks are hardly reused in queries with a substantially different context. In a more general case the same chunk is included in different queries with different contexts: a chunk on Richard Feynman could appear in prompts together with chunks on other physicists, but also Nobel prize winners, or bongo players. We show in Section 4.2 that this clustering effect makes it overly easy to identify offline what cross-attention to bake into KV caches at construction time, just by relying on semantic proximity of chunks and without requiring any knowledge about the query and context distributions. The second property is the limited presence of cross-query chunk reuse. A direct approach to warm KV cache reuse is to run a full prefill the first time a chunk is encountered, so that the saved KV carries the cross-attention from the preceding chunks present in that first query. We show in Section 4.3 that in the most widely used datasets, queries are mostly composed of chunks that appear only once across the workload.

Contributions. We make two contributions. First, in Section 4, we present an evaluation methodology that isolates the accuracy loss attributable to KV cache reuse by removing its principal confounds. We define the *meaningful subset* of a benchmark as the set of queries on which the baseline answers correctly with full prefill, the model cannot answer without context, and the answer space is non-trivial. We quantify the gap from the aggregate scores reported by prior work: on the Q&A subset of the widely used LongBench benchmark (Bai et al., 2024), up to 42% of a reuse method's F_1 comes from queries that the full-prefill baseline cannot answer at all. Second, in Section 5, we propose a tool that generates a new dataset, which we call `Boxoffice`, specifically tailored to evaluate KV cache reuse strategies. `Boxoffice` is *meaningful by construction*: chunks are drawn from a fictional corpus, so the LLM parametric memory cannot help, answers have a low chance of being guessed by chance, and the baseline with full prefill answers the query correctly. `Boxoffice` also guarantees cross-query chunk reuse, and exhibit controlled, nuanced reuse patterns, including a new dynamic we call *stale cache*, where the same chunk is reused in conflicting contexts. We show that the same reuse method can score an F_1 of 0.98 or 0.00 depending only on the context in which a chunk was cached in the first place. These dynamics reproduce across three query templates and on models of different sizes (8B and 32B), and the reuse regime `Boxoffice` generates matches what production RAG traces report. Code and datasets are available at (Anonymous Authors, 2026a;b).

2 Background on Position-independent KV cache reuse

A Transformer (Vaswani et al., 2017) LLM consists of stacked layers; within each layer, input activations are linearly projected to query, key, and value matrices Q, K, V, and self-attention is computed as $A(Q, K, V) = \mathrm{Softmax}(QK^\top/\sqrt{d_k})\,V$. Inference proceeds in two phases: *prefill*, which processes the input prompt and computes K and V, and *decode*, which generates one output token per step. Prefill cost is quadratic in prompt length; the *KV cache* stores K and V so that decode reuses them. RAG systems prepend retrieved text chunks to the user query, at the cost of inflating the prompt and the prefill stage. Position-independent KV cache reuse, hereafter simply KV cache reuse, addresses this cost: each chunk's KV cache is computed once and can be reused at any position in other queries.

Challenges of KV cache reuse. A reused KV cache differs from the one a normal prefill would compute over the current prompt. First, the cached K entries encode the chunk's original positional embedding rather than its position in the new prompt; for RoPE-based models (Su et al., 2024) this is corrected by rotating K to the new offset (Yao et al., 2025; Lu et al., 2024). Second, the cached attention states encode no *cross-chunk* attention: tokens in one chunk carry no attention to tokens in any other chunk, whereas under full prefill every token attends to the preceding tokens. The widespread solution to this issue is *selective recomputation*, depicted in Figure 1: at query time, a fraction of tokens is selected and their attention values are recomputed against the reused K and V, recovering part of the missing cross-attention (Yao et al., 2025).

Cold KV cache reuse. The first breed of systems, which we refer to as *cold*, recover cross-attention only by means of selective recomputation, and differ in how they select the tokens to recompute. CacheBlend (Yao et al., 2025) selects the tokens whose freshly computed KV at layer 1 diverges the most from the cached values, i.e., the ones affected the most by the missing cross-chunk attention. Following systems, such as Cacheclip (Yang et al., 2025b) and ProphetKV (Wang et al., 2026b), identify, via cheap surrogates of the query-to-cache attention, the tokens the query will attend to

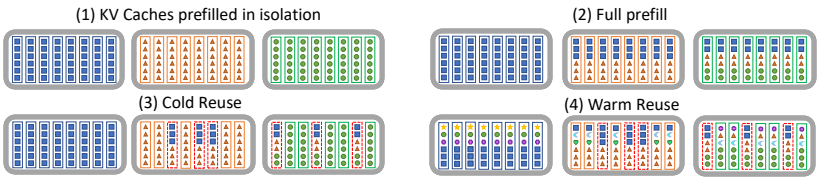


Figure 1: KV cache reuse visualization for a query over 3 chunks. Each gray box is the KV cache of one chunk, the rectangles inside it are its tokens; each chunk is associated with a shape; the shapes inside a token denote the chunks towards which it stores attention; dashed outlines mark tokens whose attention is recomputed. **(1)** *KV caches prefilled in isolation*: each cache stores intra-chunk attention only. **(2)** *Full prefill*: each cache additionally stores cross-chunk attention to other previous chunks in the prompt. **(3)** *Cold reuse*: start from (1) and recover cross-chunk attention through selective recomputation. **(4)** *Warm reuse*: start from caches that already store cross-chunk attention to other chunks, some of which coincide with the prompt's chunks and some of which do not (e.g. moon, stars); additionally recover the cross-chunk attention to the prompt's chunks through selective recomputation.

most, and recompute that subset. Across all cold variants, however, approximating full prefill accuracy requires recomputing a substantial fraction of tokens at query time.

Warm KV cache reuse. A second line of work, which we refer to as *warm*, embeds cross-chunk attention into the saved KV *at store time*, so that less has to be recovered at query time. FusionRAG (Wang et al., 2026a) generates the KV cache of a chunk C by prepending the chunk with its top-k nearest chunks T retrieved from the corpus by embedding similarity, so that the KV cache of C encodes the cross-chunk attention to T. Such *offline* approach assumes that if a query requires C, it also needs chunks similar to C.

Online warm methods build the warm cache opportunistically from the observed workload, so that the chunks that appear together in the prompts are determined by actual queries. LMCache (Liu et al., 2025) runs a full prefill the first time a chunk C appears in a query, and saves the corresponding KV cache, which hence carries the cross-attention from the chunks preceding C in that first query. CacheCraft (Agarwal et al., 2025) extends this idea with multi-versioning: it keeps several saved KVs per chunk, one for each distinct prefix the chunk has been seen with, and at query time selects the version whose prefix shares the most chunks with the current prompt.

3 Testing environment

Tasks. KV cache reuse has been investigated across three main tasks: question answering (QA), summarization, and agentic workloads. QA is the most amenable to systematic study: *multi-hop* datasets come with the guarantee that providing the correct answer requires cross-attention among chunks, with a clear correct answer, and with an explicit set of *golden chunks* (the ones that actually contribute to the answer). Summarization tasks exhibit no reuse because chunks of a text to summarize appear only once in the dataset. In agentic workloads, prompts differ only in specific portions (Ye et al., 2025; Liu et al., 2026a), or are the same but processed by different models (Ye et al., 2025; Liu et al., 2026b); solutions tailored to them do not apply to the RAG case we study, where queries and contexts differ at arbitrary points.

Datasets. Q&A is by far the most studied task in the KV cache reuse literature. We find that the vast majority of papers evaluate on a common set of three multi-hop subsets of LongBench (Bai et al., 2024): MuSiQue, HotpotQA, and 2WikiMQA. We therefore focus our study on these three datasets, which we refer to as *LongBench-QA*, that comprise 200 points each. Appendix A breaks down the prevalence of each dataset across the surveyed papers.

Models and metrics. For the core of our experiments we use LLama3.1-8B (Llama Team, 2024), Qwen3-8B (Yang et al., 2025a), and Mistral-7B (Jiang et al., 2023), which are used the most in existing experimental studies. We run a subset of the experiments of Sections 4 and 5 on Qwen3-32B to show that the dynamics that we present generalize to larger-size models. We use `e5-base-v2` (Wang et al., 2022) to compute the embeddings of the chunks, and cosine similarity to measure similarity of embeddings. We use F_1 score as metric to evaluate accuracy (van Rijsbergen, 1979), and we report the norm-F_1, i.e., the F_1 score obtained by a KV cache reuse method normalized w.r.t. that obtained with full prefill, so a value of < 1 indicates loss of accuracy.

KV cache reuse techniques. We consider the KV cache reuse techniques implemented by the following state-of-the-art systems, described in Section 2. CacheBlend (CB, offline, cold),

Table 1: **Left:** ρ under the progressive filters of Section 4.1; $\rho < 1$ means measuring $F1$ on the unfiltered dataset inflates the reported accuracy. **Right:** $B0$ ablation. B_0M_0 counts queries on which baseline and method both fail; B_0M_+ counts queries on which the baseline fails but the method succeeds; $\%F_1$ is the share of the unfiltered aggregate F_1 contributed by B_0M_+.

		$\rho_{F_1} = \text{normF1(filter)}/\text{normF1(ALL)}$									B0 ablation (200 points per dataset)								
		qwen			llama			mistral			qwen			llama			mistral		
	method / filter	B_0	+NC	+LI	B_0	+NC	+LI	B_0	+NC	+LI	B_0M_0	B_0M_+	$\%F_1$	B_0M_0	B_0M_+	$\%F_1$	B_0M_0	B_0M_+	$\%F_1$
musique	CB	0.76	0.69	0.69	0.91	0.87	0.87	0.70	0.80	0.80	95	26	24.1	108	13	9.5	96	36	29.8
	CB+Q	0.72	0.58	0.58	0.82	0.83	0.83	0.68	0.71	0.71	93	28	27.6	104	17	18.1	91	41	32.5
	FR	0.74	0.64	0.64	0.83	0.70	0.70	0.58	0.67	0.67	98	23	26.4	103	18	16.9	121	11	42.4
	FR+Q	0.73	0.65	0.65	0.75	0.70	0.70	0.63	0.79	0.79	95	26	26.6	97	24	25.4	123	9	36.8
2wiki	CB	0.87	0.58	0.52	0.88	0.84	0.86	0.96	0.81	0.79	89	15	13.2	73	13	12.5	61	12	4.5
	CB+Q	0.75	0.49	0.45	0.88	0.73	0.73	0.94	0.75	0.62	83	21	24.9	72	14	12.3	61	12	5.8
	FR	0.76	0.70	0.66	0.84	0.73	0.68	0.95	0.82	0.70	81	23	24.0	71	15	15.8	64	9	4.8
	FR+Q	0.81	0.70	0.69	0.82	0.66	0.71	0.90	0.81	0.75	87	17	18.5	68	18	18.4	61	12	9.5
hotpot	CB	0.93	0.91	0.92	0.95	0.90	0.90	0.92	0.94	0.92	50	14	7.3	56	10	5.2	43	30	8.1
	CB+Q	0.94	0.77	0.79	0.97	0.94	0.96	0.92	0.94	0.93	51	13	5.9	62	4	3.3	46	27	7.5
	FR	0.94	0.82	0.80	0.93	0.80	0.81	0.93	1.01	0.98	54	10	5.9	55	11	6.5	58	15	6.6
	FR+Q	0.97	0.90	0.89	0.94	0.85	0.84	0.93	0.97	0.94	56	8	3.0	55	11	5.6	59	14	7.1

FusionRag (FR, offline, warm), LMCache (LM, online, warm), Cachecraft (CC, online, warm). By default, tokens for recomputation are selected as in Cacheblend, by K matrix deviation at layer 1. We also implement a query-driven variant (noted Q) that selects the tokens as in FusionRag by identifying the highest-attention tokens via a lightweight computation between the last-layer query and K caches. CacheCraft$-M$ runs a full prefill whenever it encounters a chunk with fewer than M stored warm caches; the case M=1 recovers ***LMCache***. We implement these variants in a single framework based on HuggingFace Transformers (Wolf et al., 2019). Unless stated otherwise, $R = 15\%$ of the tokens are selectively recomputed to recover cross-attention, a common setting in experimental evaluations.

4 Limitations of existing studies

4.1 Evaluation methodologies: pitfalls and our solution.

Current pitfalls. To the best of our knowledge, all experimental analysis of KV cache reuse systems report aggregate accuracy metrics, e.g., F_1, computed over every query of the evaluation set for the full prefill baseline and for the target system. We argue that this is the wrong evaluation methodology. The target metric for evaluating KV cache reuse should measure the ability to preserve full prefill accuracy on queries where the correct answer genuinely depends on the provided context. An aggregate over every query in a benchmark conflates this metric with three distinct phenomena: i) accidental regularization brought by approximated attention states, which may arise on queries that the model fails to answer with full prefill; ii) parametric leakage, which may arise on queries the model answers from parametric memory alone; and iii) chance correctness, which may happen on yes/no or disjunctive questions.

Proposed solution. We suggest to exclude three classes of queries from evaluation datasets: i) queries where full prefill achieves 0 accuracy score with the target context. On these queries there is no accuracy-preservation capability to evaluate; any score gain achieved by the reuse strategy is due to accidental regularization brought by the approximated attention states; ii) queries that the model can answer correctly without additional context. The model can answer such queries without looking at the KV caches, so they are unfit to capture the ability of KV cache reuse strategies to preserve the model's ability to extract information from the context; and iii) queries with a low-information answer (hereafter *low-info*), namely boolean yes/no queries and disjunctive queries of the form *who is richer, Elon Musk or Jeff Bezos?* The answer to such queries can be guessed correctly with probability 50% by either the baseline or the reuse strategy. Low-info queries may seem addressable by prompt-based mitigations ("use only the provided context"), but such instructions are not reliable (Pezeshkpour & Hruschka, 2024; Zheng et al., 2024a; Sclar et al., 2024).

We refer to the queries that survive all three filters as the *meaningful subset* of the benchmark, since they are the ones on which the loss of accuracy attributable to KV cache reuse is observable.

Impact of the evaluation methodology. We empirically quantify the impact of each filter on LongBench-QA, and show that reporting aggregate accuracy over all queries systematically overstates how well a KV cache reuse method preserves full prefill accuracy. We define three filters over the queries of a benchmark. **B0** drops queries on which the full prefill baseline scores $F_1 = 0$ on the

Table 2: Mean norm-F1 of different offline cross-chunk attention infusion modes. The provided cross-chunk attention helps significantly over a random baseline only if it aligns with the query's context or golden chunks.

	qwen			llama			mistral		
method	2wiki	hotpot	musique	2wiki	hotpot	musique	2wiki	hotpot	musique
CACHEBLEND (NO INFUSION)	0.13	0.21	0.10	0.38	0.35	0.19	0.14	0.19	0.13
RANDOM	0.39	0.52	0.35	0.48	0.55	0.43	0.32	0.48	0.27
SIMILARITY-NO-CTX	0.35	0.47	0.37	0.44	0.56	0.41	0.28	0.44	0.19
CONTEXT-NOGOLD	0.44	0.56	0.48	0.53	**0.66**	0.53	0.41	**0.62**	0.32
GOLDEN-ONLY	**0.63**	**0.67**	**0.52**	**0.57**	**0.66**	**0.56**	**0.52**	0.59	**0.34**
FUSIONRAG (SIMILARITY)	0.55	0.59	0.36	**0.57**	0.61	0.42	0.42	0.55	0.20

target context. **NC** drops queries that the model can answer without any context, identified as queries on which the model achieves $F_1 > 0.2$ when prompted with the question alone. **LI** drops yes/no and disjunctive queries. For each method, we measure the norm-F1 obtained on the dataset that results from progressively applying these filters, and report the ratio $\rho = \text{norm-F1}_{\text{filtered}}/\text{norm-F1}_{\text{whole}}$; $\rho < 1$ indicates that the filtered dataset reveals more accuracy loss than the unfiltered headline reports. We compute ρ for **CB**, **CB+Q**, **FR**, and **FR+Q**, which span cold (**CB**, **CB+Q**) and warm-offline (**FR**, **FR+Q**) reuse, to show that the conclusion is not specific to a single reuse strategy. We exclude warm-online methods (LM, CC) because, as we will show in Section 4.3, LongBench-QA does not exhibit the cross-query chunk recurrence that those methods require.

The results in Table 1 (left) show that ρ generally decreases as we restrict the evaluation dataset, indicating that each class of filtered-out queries contributes to inflating the accuracy score aggregated over the whole dataset. The B0 filter has the largest individual impact, so we provide an additional ablation (Table 1, right). For each (method, dataset, model) cell we count the queries on which both the baseline and the method score $F_1 = 0$ (**B0M0**), and those on which the baseline scores $F_1 = 0$ but the method scores $F_1 > 0$ (**B0M+**). The latter set is the queries on which the approximate reused cache flips the answer from totally wrong to at least partially right. We measure the percentage of the aggregate F_1 score attributable to B0M+ queries; by construction it equals $1 - \rho$ after B0, so B0 quantifies this inflation exactly, while NC and LI remove inflation not implied by construction.

The data shows that, first, a significant fraction of queries are unanswerable by either the baseline or the method, consistent with the low aggregate F_1 scores typically reported on LongBench-QA (between 0.2 and 0.5). Second, flipping queries contribute substantially to the aggregate F_1 score, reaching 42% on MuSiQue and 24.9% on 2WikiMQA, the two harder datasets. This reinforces the message that including baseline-zero queries inflates the F_1 scores reported by existing studies through dynamics that are orthogonal to the preservation of full prefill accuracy.

Appendix F shows that the results presented hold for any threshold in $[0, 0.5]$, reports the size of the meaningful subset per dataset and model, and discusses its model dependence. Appendix G shows that the F_1 inflation phenomenon persists, attenuated, on a $4\times$ larger model (Qwen3-32B).

Discussion. The classes of queries we filter out may capture useful side-effects of KV cache reuse, e.g., accidental regularization on long contexts, as documented in sparse attention and KV cache compression (Xiao et al., 2024) evaluation studies. In those cases, however, the effect of such regularization is infrequent and mild, which may justify an aggregate accuracy score over an unfiltered dataset. In our study, instead, such accidental regularization significantly impacts accuracy and shows up consistently across methods, models, and datasets. In the context of reuse, these phenomena deserve their own investigation, but existing evaluations provide no ablation studies in this direction. Our proposed methodology measures the accuracy degradation attributable to KV cache reuse in isolation.

4.2 DATASET LIMITATIONS FOR WARM REUSE - THE OFFLINE CASE

What cross-attention is needed? We run an experiment to evaluate the effect of injecting different cross-chunk attention to KV caches used to serve a query. For each chunk in a query, we construct its KV cache with a prefix of additional chunks; the prefix determines what cross-attention the cache stores. We test six prefix conditions per chunk in the context T: no prefix (plain *CacheBlend*); random chunks drawn from the dataset; semantically close chunks that are not in T; the golden chunks of T; non-golden chunks that appear in T; and ten semantically close chunks, as in *FusionRAG*. To isolate the effect of cross-attention infusion we set $R = 0$. Table 2 reports the resulting norm-F1 on the meaningful subset of queries identified in the previous section. Cross-attention infusion improves over plain Cacheblend, but only when the injected attention is drawn from chunks that

Table 3: **Left:** CC (GC) is the percentage of a query's context (golden) chunks identifiable by cosine similarity alone, without knowledge of the query. High values indicate that warm offline retrieval can identify the right cross-attention targets purely from chunk embeddings. **Middle:** mean cosine similarity between chunk pairs g: goldens of the query; c: non-golden context chunks of the query; o: chunks outside the query. Golden chunks and context chunks are clustered and easily separable from chunks of other queries. **Right:** reuse statistics. C: % of unique chunks that appear in ≥ 2 queries. G: among the reused chunks, the % that are golden in some query. F: % of queries whose entire context has already appeared in earlier queries.

Dataset	CC	GC	$g \to g$	$g \to c$	$g \to o$	$c \to g$	$c \to c$	$c \to o$	C	G	F
MuSiQue	0.54	0.69	0.77	0.73	0.70	0.73	0.74	0.70	12.6%	26.6%	0%
2WikiMQA	0.56	0.81	0.78	0.75	0.71	0.75	0.75	0.70	11.9%	0.7%	0%
HotpotQA	0.84	0.88	0.81	0.78	0.70	0.78	0.78	0.70	0.7%	16.7%	0%

actually appear in the target query, with golden chunks providing the largest gain. Cross-attention from semantically close chunks that do *not* appear in T offers no consistent benefit over random chunks. The random variant itself is already better than the no-infusion case, which we attribute to infusion suppressing the attention sinks that distort the dynamics of independently-prefilled KV caches (Xiao et al., 2024; Yang et al., 2025d; Hu et al., 2024).

Important cross-attention is easily identified. On LongBench-QA, identifying golden and context chunks offline is easy: the cosine-similarity neighborhood of a chunk C already contains most of the context of the queries in which C is used. For a query Q with context T_Q and golden subset $G_Q \subseteq T_Q$, let $\text{TOPK}(c)$ denote the k nearest neighbors of c in the corpus by cosine similarity, and $U_k(Q) = \bigcup_{c \in T_Q} \text{TOPK}(c)$ the union of these neighborhoods. We define *context coverage* $CC(Q) = |U_k(Q) \cap T_Q|/|T_Q|$ and *golden coverage* $GC(Q) = |U_k(Q) \cap G_Q|/|G_Q|$. Table 3 (left) reports both at k=10: offline neighbor-based infusion already provides cross-chunk attention, on average, to at least 54% of a query's context chunks and 69% of its golden chunks. Warm offline systems can therefore identify the right cross-attention targets without any knowledge of the query. The reason is a tight cluster structure: for each chunk in a query, we compute its cosine similarity to every other golden chunk of the same query, every non-golden context chunk of the same query, and every chunk outside the query. We then report the average broken down by the role of the two chunks in the pair: g (golden), c (non-golden context), o (outside the query). Table 3 (right) shows that golden chunks lie close to other golden chunks, and context chunks lie close to other context chunks, especially the golden ones. Appendix C.4 visualizes this through the distance CDFs. This tight clustering is not representative of a broad class of RAG workloads where the same chunk participates in queries that involve different contexts. Our analysis of industry RAG workloads (Appendix I) shows that individual chunks recur often across queries, but the chunks they appear with rarely do: two requests that share a chunk have a median Jaccard similarity of only about 0.1 between their chunk sets, and a chunk's actual co-used partner is among its 10 nearest neighbours computed offline in only 8% of cases. These workloads challenge the core assumption behind chunk-similarity-based approaches.

4.3 Dataset limitations for warm reuse - the online case

We analyse whether LongBench-QA is fit to evaluate warm online KV cache reuse methods. The main required property is that the same chunk appears in at least two queries: one occurrence seeds the warm KV cache, the other is the query on which reuse is evaluated; without it, no KV cache reuse happens.

We further identify two desirable properties. First, there must exist *queries whose chunks are all reused*. This enables steady-state evaluation of warm systems: on a chunk's first appearance, a warm system may run a partial or full prefill (Liu et al., 2025; Yao et al., 2025; Ye et al., 2025), hiding the accuracy degradation that would arise at steady state under full reuse. Full reuse is the regime that the evaluations of all cold methods reproduce, and it is also the common case in practice: in the RAGPulse production trace, 73% of first-week requests (93% after one week of warmup) consist entirely of previously seen chunks, as do 40–50% of the questions in the HERB enterprise benchmark. Second, *the reused chunks should be golden*. The purpose of reuse is to preserve/restore, the attention states that matter for answering the query. If reuse events are concentrated on *distractor* chunks, the evaluation cannot evaluate if a method can preserve/restore the meaningful cross-attention.

Existing datasets are unfit for warm KV cache reuse. Table 3 (right) reports three metrics computed on LongBench-QA: i) *chunk reuse* (C), i.e., the percentage of unique chunks that appear in

two or more queries; ii) *golden reuse* (G), i.e., the percentage of golden chunks among the reused chunks; and iii) *full-context reuse* (F), i.e., the fraction of queries whose chunks have *all* appeared already in previous queries. C lies between 0.7% and 12.6%, and re-used chunks are golden chunks only in a minority of the cases (between 0.7% and 26.6%). Finally, no query can be served entirely from re-used KV caches, meaning that warm online methods would run at least a partial prefill on every query of the dataset to populate the KV cache.

5 BOXOFFICE: A STRESS-TEST BENCHMARK FOR KV CACHE REUSE

`Boxoffice` is a tool that programmatically generates a synthetic dataset designed to evaluate KV cache reuse techniques. `Boxoffice` produces a corpus of structured records describing fake movies, and a set of queries on such corpus. Each record describes a fictional movie with a small set of fields, prominently a numeric box office field, plus short paragraphs that paraphrase the schema content. Each query asks to identify the movie with the highest box office value among a fixed-size candidate set provided in the context (whose default size is 10). This argmax template is the default one, which we discuss in detail in this Section. Appendix H presents two additional templates, a categorical uniqueness task and a two-hop pivot join, and shows that the dynamics we present generalize to other query templates. `Boxoffice` addresses all the limitations that we have discussed so far, and is intended as a controlled stress-test complement to existing benchmarks rather than as a replacement.

Meaningful by construction. Because the films are fictional, queries cannot be answered from parametric memory of the model. Moreover, `Boxoffice` includes in the produced datasets only queries that can be answered correctly with full prefill by the target model(s). Finally, with 10 candidates per query, the chance that a randomly guessed answer is correct is low (0.1).

Heterogeneous contexts. Queries draw from a shared pool of chunks, and the same chunk appears alongside different context chunks in different queries (on the released datasets, every candidate slot holds a reused chunk and the median Jaccard similarity between queries sharing a chunk is 0.11, as in the workloads of Section 4.2). The within-query semantic clustering identified in Section 4 cannot occur on `Boxoffice`. For a corpus of 100 movies, k=5 and 10 chunks per query, golden coverage at k=5 using chunk semantic similarity is 0.32, no better than the 0.37 that random sampling of 5 chunks from the 100-movie corpus would yield in expectation. The mean pairwise cosine similarity is 0.89 both within a query and from a golden chunk to chunks outside the query (see also Appendix C.4): query membership carries no embedding signal, so top-k similarity cannot distinguish in-query from outside-query chunks.

Programmatic golden reuse. `Boxoffice` splits queries into a *warmup* set and an *evaluation* set: every chunk used in the evaluation set has already appeared in at least one warmup query, so all evaluation queries rely on chunks with a re-usable KV cache. A configurable parameter M controls the minimum number of warmup appearances per evaluation chunk. This feature enables the evaluation of systems that maintain M different KV caches of the same chunk, such as CacheCraft. Note that in `Boxoffice` every chunk is effectively golden: the answer is the argmax of the box office field across the candidate set, so every candidate dossier contributes to answering correctly.

5.1 CACHE STALENESS, A NEW KV CACHE REUSE DYNAMIC

`Boxoffice` controls not only whether a chunk has been seen before but also the role under which it was seen. For any query, we call the *winner* the chunk whose box office value is maximal among the context candidates and the *nonwinners* the remaining ones. During warmup, a target chunk is cached after predecessor chunks whose box office values are either all strictly smaller than the target's, in which case the target was the winner of its warmup query, or all strictly larger, in which case the target was a nonwinner. The cached state therefore carries a directional signal: a winner-cached chunk implicitly encodes "I am the maximum among my predecessors", and a nonwinner-cached chunk encodes the opposite. We refer to a cached chunk as *stale* when the role it carries from warmup does not match its role in the query at hand.

Four warmup configurations. Every evaluation query falls into one of four warmup configurations, defined by the role under which each chunk was cached when it appeared for the *first* time in the warmup. Two binary axes determine the configuration: whether the evaluation winner was cached as a winner (W) or as a nonwinner (N), and whether all evaluation nonwinners were cached as winners or as nonwinners. We label the four configurations `aligned`, `all-W`, `all-N`, and

Table 4: norm-F1 on Boxoffice. Within each (model, configuration), the winning method is in bold.

	Llama										Qwen										Mistral									
Config	CB	CB+Q	FR	FR+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q	CB	CB+Q	FR	FR+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q	CB	CB+Q	FR	FR+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q
aligned	.10	.31	.11	.37	**.98**	.95	.17	.55	.19	.51	.39	.09	.44	.40	**.97**	.94	.17	.38	.18	.25	.20	.18	.33	.53	**.98**	.94	.14	.42	.21	.36
all-W	.06	.21	.05	.28	.14	.29	.13	.42	.24	**.50**	.30	.12	.28	.29	.11	.25	.13	.27	.24	**.32**	.24	.15	.15	.26	.17	.26	.14	.26	.31	**.34**
all-N	.06	.25	.13	.43	.07	.55	.16	.57	.23	**.57**	.39	.20	.30	**.39**	.05	.11	.18	.36	.24	.34	.29	.16	.24	**.57**	.09	.29	.16	.34	.32	.39
flipped	.09	.25	.06	.35	.00	.09	.11	.41	.23	**.54**	**.41**	.18	.23	.28	.00	.06	.15	.29	.28	.34	.25	.11	.18	**.34**	.00	.02	.13	.23	.30	.30
average	.08	.26	.09	.36	.30	.47	.14	.49	.22	**.53**	**.37**	.15	.31	.34	.28	.34	.16	.33	.24	.31	.24	.15	.23	**.42**	.31	.38	.14	.31	.28	.35
std	.03	.04	.02	.03	.03	.03	.05	.07	.04	.06	.05	.03	.04	.04	.02	.02	.05	.05	.04	.04	.03	.04	.03	.04	.02	.04	.05	.06	.07	.05

flipped; Table 6 describes each one, the cached signals it presents to the reuse method, and the qualitative difficulty it imposes. The evaluation set is balanced across configurations.

Staleness with multiple cached versions per chunk. When the warmup is configured with $M>1$ cached versions per chunk, the first cached version reflects the role the chunk plays at its first warmup appearance, and the following $M-1$ appearances balance the role distribution: by the end of warmup, at least $\lfloor M/2 \rfloor$ appearances are in the winner role and at least $\lceil M/2 \rceil$ in the nonwinner role, or vice versa. Each chunk therefore has at least one warmup appearance whose role matches its role in an evaluation query and at least one whose role does not (e.g., with $M=2$, a chunk first cached as a winner is later placed as a nonwinner). This construction evaluates whether a method that stores $M>1$ versions per chunk, such as *CacheCraft*, can pick the version whose cached signals best match the current query, and hence recover from staleness that a single-version method can only address by selective recomputation.

5.2 Evaluating KV cache reuse with Boxoffice

We evaluate the full set of KV cache reuse methods introduced in Section 2 on Boxoffice. Table 4 reports norm-F1 (which equals F_1 by construction), providing the average value across all the evaluation queries, and a breakdown by configuration. The evaluation set includes 160 queries (40 per configuration); warmup queries are roughly 250 and guarantee that each chunk in the evaluation set appears at least 4 times each during warmup, to enable the evaluation of $CC-M4$. We average the results across 10 different generated datasets; sample standard deviations across seeds and results with $R = 0.05$ are reported in Appendix C.5.

Result 1. Boxoffice's warmup elicits configuration-dependent reuse dynamics. Boxoffice's configuration-driven warmup biases each chunk's cached KV strongly enough to drive the reuse dynamics visible in Table 4. The clearest evidence is in the two extreme configurations on LM: on aligned, where every cached KV encodes the same role the chunk plays at query time, *LM* achieves near-perfect norm-F1 across all three models (≥ 0.97); on flipped, where every cached role contradicts the role the chunk plays at query time, *LM* drops to 0.00 on every model. The intermediate configurations sit between the two extremes: the cached cross-attention is partially incorrect, and the recomputation budget can only partially repair it. The same collapse and recovery appear on the two additional templates (Appendix H) and with Qwen3-32B (Appendix G).

Result 2. Chunk similarity alone is not effective for warm offline KV cache reuse on Boxoffice. *FR* is on average on par with *CB* on LLama and Mistral, and worse than it on Qwen. This is a direct consequence of Boxoffice's non-clustered chunk distribution (Section 5). The Q-augmented variant *FR+Q* outperforms *CB+Q* across all three models on average; we attribute this to the warmup itself providing better KV cache conditioning (e.g., removing attention sinks (Hu et al., 2024; Yang et al., 2025d)) that +Q can leverage at recomputation time, not to *FR*'s neighbour-based mechanism: the same dynamic arises under random offline warmup (Section 4.2).

Result 3. Query-driven token selection helps the warm methods. On average, the +Q variant improves over the deviation-based default on every warm method (*FR*, *LM*, *CC-M2*, *CC-M4*) and every model. The picture is different for the cold method *CB*: +Q helps on Llama but regresses on Qwen and Mistral, plausibly because cold KV caches encode distorted attention patterns such as attention sinks (Xiao et al., 2024) that confound a selection mechanism relying on attention scores against the cached KVs. Even on the warm methods, +Q does not close the gap with the full prefill baseline, although Boxoffice makes the key information for each query identifiable: better token-selection strategies are needed, and Boxoffice provides a controlled testbed to develop them.

Result 4. Multi-version KV caching is powerful but its selection criterion matters. Multi-version methods reach the highest overall norm-F1 on Llama, so storing multiple cached versions per chunk pays off when the right version is selected at query time. However, on `aligned` queries, *LM* ($M{=}1$) outperforms every $M{>}1$ configuration, because with $M{>}1$ the system can pick the wrong cached version. *CC* selects a cache by prefix similarity, a criterion oblivious to whether the chosen cache is aligned with the role the chunk plays in the current query; with 10 chunks per query, even a small per-chunk probability of mis-selection compounds, so at least one chunk's cache is likely to break the perfect alignment.

Result 5. No single strategy is robust across models. On average, the winner differs on every model: *CC-M4+Q* on Llama , *CB* on Qwen, and *FR+Q* on Mistral. These three winners are heterogeneous in the hot/cold and online/offline taxonomy, so no family of KV cache reuse strategies is preferable across models, and conclusions drawn from a single model risk misgeneralising.

6 Related work

KV cache reuse methodologies and datasets. Existing evaluations of KV cache reuse (Yao et al., 2025; Wang et al., 2026a; Agarwal et al., 2025; Liu et al., 2025) report aggregate accuracy on standard QA benchmarks, primarily from LongBench (Bai et al., 2024) (as shown in Appendix A). We show in Section 4.1 that this inflates reported retention, substantially overstating the effectiveness of the techniques. To simulate the cross-query reuse that standard QA datasets lack, two simple workarounds have been used: LLM-generated questions over a fixed context (Yao et al., 2025; Feng et al., 2025), and partitioning a single long document into smaller chunks queried by multiple questions (Jiang et al., 2025). Both have limitations that `Boxoffice` addresses. They yield clustered contexts, since paraphrased questions re-target the same chunks and chunks split from one document share a topic by construction; they do not enforce golden-chunk reuse; and they do not generate queries composed entirely of reused chunks. The evaluation of CacheCraft (Agarwal et al., 2025) relies on a proprietary workload that is not released. `Boxoffice` enables the reproducible evaluation of multi-versioned KV cache reuse strategies such as CacheCraft, enforcing parametric reuse of chunks so that enough versions accumulate.

Evaluation beyond KV cache reuse. Our methodology aligns with prior evidence that aggregate accuracy can be inflated by queries answered from dataset artifacts or memorised test items (Gururangan et al., 2018; McCoy et al., 2019; Sainz et al., 2023; Magar & Schwartz, 2022), with calls for per-capability rather than single-score evaluation (Hooker et al., 2020; Ribeiro et al., 2020; Chen et al., 2024), and with work that separates the contribution of the context from that of parametric memory (Mallen et al., 2023; Tao et al., 2024; Longpre et al., 2021; Xiong et al., 2024); synthetic or freshly sourced data has likewise been used to prevent parametric leakage (Zhang et al., 2024; White et al., 2024; Eyuboglu et al., 2025; Wei et al., 2023). See Appendix B for details.

These results and insights back the methodology we propose to clearly measure the accuracy loss attributable to KV cache reuse. `Boxoffice` follows this methodology by construction, and enables the evaluation of KV cache reuse strategies on challenging cross-query reuse dynamics.

7 Conclusions

We show limitations in the methodology and the datasets currently used to evaluate KV cache reuse strategies, propose remedies for both, and release `Boxoffice`, a tool that enables the evaluation of KV cache reuse under challenging and controllable reuse patterns, towards a more thorough and reproducible evaluation of KV cache reuse strategies.

Limitations. First, `Boxoffice` is built around three query templates over one synthetic corpus (Appendix H), whereas suites such as LongBench span multiple tasks. Second, our main evaluation covers three open-source $\sim$ 8B models; Appendix G verifies on Qwen3-32B that the accuracy inflation and the staleness dynamic persist, but other architectures may be affected quantitatively differently by the filtered-out queries, even though the methodological argument is unchanged. Third, we focus on recomputation-based KV cache reuse, the family currently seeing industry adoption (Liu et al., 2025; Agarwal et al., 2025; ModelEngine-Group, 2025; Wang et al., 2025a). The inflation analysis and the meaningful subset apply unchanged to approaches based on fine-tuning or post-training (Lu et al., 2024; Ma et al., 2025; Yang et al., 2025c), since the filters only look at the answers a method produces; the staleness dimension is instead specific to methods that store context-conditioned caches, for which `Boxoffice` serves as a leakage-free testbed.

AI USE STATEMENT

In this work, we used generative AI tools for editing (grammar, spelling, word choice), for retrieval and discovery of related work, and for assisting with code development (helper scripts for data processing, filtering, and figure generation); all AI-assisted code was reviewed and tested by the authors, and all results were produced by running the released code. We have not used generative AI tools to generate synthetic datasets (the `Boxoffice` corpus and queries are produced programmatically by the released generator), to develop theoretical models, to propose hypotheses, to design experiments, or to interpret results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The code for every method, the LongBench-QA enrichment scripts, and the `Boxoffice` generator (including the two additional templates of Appendix H) are publicly available in an anonymized repository, `https://github.com/boxoffice1280/boxoffice1280`, and the exact generated datasets used in this paper in an anonymized dataset release, `https://huggingface.co/datasets/Boxoffice1280/boxoffice1280` (Anonymous Authors, 2026a;b). Section 3 lists models, embedding model, metric, and recomputation budget; Appendix C gives the exact prompts and the generation pseudocode; Appendix C.5 reports per-seed variability and the runtime of an end-to-end run.

REFERENCES

Shubham Agarwal, Sai Sundaresan, Subrata Mitra, Debabrata Mahapatra, Archit Gupta, Rounak Sharma, Nirmal Joshua Kapu, Tong Yu, and Shiv Kumar Saini. Cache-craft: Managing chunk-caches for efficient retrieval-augmented generation. *Proc. ACM Manag. Data*, 3(3):136:1–136:28, 2025. doi: 10.1145/3725273. URL `https://doi.org/10.1145/3725273`.

Anonymous Authors. Boxoffice1280 dataset: released seed datasets for the BoxOffice benchmark. `https://huggingface.co/datasets/Boxoffice1280/boxoffice1280`, 2026a. Anonymized dataset release for double-blind review.

Anonymous Authors. Boxoffice: Code and data for cache-reuse experiments on BoxOffice and LongBench-QA. `https://github.com/boxoffice1280/boxoffice1280`, 2026b. Anonymized repository for double-blind review.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3119–3137, 2024. URL `https://aclanthology.org/2024.acl-long.172/`.

Ziyi Cao, Qingyi Si, Jingbin Zhang, and Bingquan Liu. Sparse attention across multiple-context KV cache. *arXiv preprint arXiv:2508.11661*, 2025.

Samuel Cestola, Tianxiang Xia, Zheng Weiyan, Zheng Pengfei, and Diego Didona. An experimental study of KV cache reuse strategies in chunk-level caching systems, 2026. URL `https://arxiv.org/abs/2603.20218`.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. URL `https://arxiv.org/abs/2309.01431`.

Prafulla Kumar Choubey, Xiangyu Peng, Shilpa Bhagavath, Kung-Hsiang Huang, Caiming Xiong, and Chien-Sheng Wu. Benchmarking deep search over heterogeneous enterprise data, 2025. URL `https://arxiv.org/abs/2506.23139`.

Dvir Cohen, Lin Burg, Sviatoslav Pykhnivskyi, Hagit Gur, Stanislav Kovynov, Olga Atzmon, and Gilad Barkan. WixQA: A multi-dataset benchmark for enterprise retrieval-augmented generation, 2025. URL `https://arxiv.org/abs/2505.08643`.

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 (NAACL-HLT)*, pp. 4599–4610. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.naacl-main.365/.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Will Tennien, Atri Rudra, James Zou, Azalia Mirhoseini, and Christopher Ré. Cartridges: Lightweight and general-purpose long context representations via self-study. *arXiv preprint arXiv:2506.06266*, 2025. URL https://arxiv.org/abs/2506.06266.

Shaoting Feng, Yuhan Liu, Hanchen Li, Xiaokun Chen, Samuel Shen, Kuntai Du, Zhuohan Gu, Rui Zhang, Yuyang Huang, Yihua Cheng, Jiayi Yao, Qizheng Zhang, Ganesh Ananthanarayanan, and Junchen Jiang. EVICPRESS: Joint KV-cache compression and eviction for efficient LLM serving, 2025. URL https://arxiv.org/abs/2512.14946.

In Gim, Guojun Chen, Seung-Seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference. In Phillip B. Gibbons, Gennady Pekhimenko, and Christopher De Sa (eds.), *Proceedings of the Seventh Annual Conference on Machine Learning and Systems, MLSys 2024, Santa Clara, CA, USA, May 13-16, 2024*. mlsys.org, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/a66caa1703fe34705a4368c3014c1966-Abstract-Conference.html.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 107–112, 2018. URL https://aclanthology.org/N18-2017/.

Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058*, 2020. URL https://arxiv.org/abs/2010.03058.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Junhao Hu, Wenrui Huang, Haoyi Wang, Weidong Wang, Tiancheng Hu, Qin Zhang, Hao Feng, Xusheng Chen, Yizhou Shan, and Tao Xie. EPIC: efficient position-independent context caching for serving large language models. *CoRR*, abs/2410.15332, 2024. doi: 10.48550/ARXIV.2410.15332. URL https://doi.org/10.48550/arXiv.2410.15332.

Zhengding Hu, Vibha Murthy, Zaifeng Pan, Wanlu Li, Xiaoyi Fang, Yufei Ding, and Yuke Wang. Hedrarag: Co-optimizing generation and retrieval for heterogeneous RAG workflows. In Youjip Won, Youngjin Kwon, Ding Yuan, and Rebecca Isaacs (eds.), *Proceedings of the ACM SIGOPS 31st Symposium on Operating Systems Principles, SOSP 2025, Lotte Hotel World, Seoul, Republic of Korea, October 13-16, 2025*, pp. 623–638. ACM, 2025. doi: 10.1145/3731569.3764806. URL https://doi.org/10.1145/3731569.3764806.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825. URL https://doi.org/10.48550/arXiv.2310.06825.

Yinsicheng Jiang, Yeqi Huang, Liang Cheng, Cheng Deng, Xuan Sun, and Luo Mai. Contextpilot: Fast long-context inference via context reuse, 2025. URL https://arxiv.org/abs/2511.03475.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1601–1611. Association for Computational Linguistics, 2017. URL https://aclanthology.org/P17-1147/.

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*, 6:317–328, 2018. URL https://aclanthology.org/Q18-1023/.

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 Jason Flinn, Margo I. Seltzer, Peter Druschel, Antoine Kaufmann, and Jonathan Mace (eds.), *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pp. 611–626. ACM, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Yang Liu, Yunfei Gu, Liqiang Zhang, Chentao Wu, Guangtao Xue, Jie Li, Minyi Guo, Junhao Hu, and Jie Meng. CacheSlide: Unlocking cross position-aware KV cache reuse for accelerating LLM serving. In *24th USENIX Conference on File and Storage Technologies (FAST 26)*. USENIX Association, 2026a.

Yuhan Liu, Hanchen Li, Yihua Cheng, Kuntai Du, Jiayi Yao, Shan Lu, Ion Stoica, and Junchen Jiang. CacheGen: KV cache compression and streaming for fast large language model serving. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pp. 38–56. Association for Computing Machinery, 2024. doi: 10.1145/3651890.3672274.

Yuhan Liu, Yihua Cheng, Jiayi Yao, Yuwei An, Xiaokun Chen, Shaoting Feng, Yuyang Huang, Samuel Shen, Rui Zhang, Kuntai Du, and Junchen Jiang. Lmcache: An efficient kv cache layer for enterprise-scale llm inference. *arXiv preprint arXiv:2510.09665*, 2025. URL https://arxiv.org/abs/2510.09665.

Yuhan Liu, Yuyang Huang, Jiayi Yao, Shaoting Feng, Zhuohan Gu, Kuntai Du, Hanchen Li, Yihua Cheng, Junchen Jiang, Shan Lu, Madan Musuvathi, and Esha Choukse. Droidspeak: Kv cache sharing across fine-tuned model variants. In *23rd USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2026b.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. URL https://aclanthology.org/2021.emnlp-main.565/.

Songshuo Lu, Hua Wang, Yutian Rong, Zhi Chen, and Yaohua Tang. Turborag: Accelerating retrieval-augmented generation with precomputed KV caches for chunked text. *CoRR*, abs/2410.07590, 2024. doi: 10.48550/ARXIV.2410.07590. URL https://doi.org/10.48550/arXiv.2410.07590.

Dongyang Ma, Yan Wang, and Tian Lan. Block-attention for efficient prefilling. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=7zNYY1E2fq.

Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 157–165, 2022. URL https://aclanthology.org/2022.acl-short.18/.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 9802–9822, 2023. URL https://aclanthology.org/2023.acl-long.546/.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. URL https://aclanthology.org/P19-1334/.

ModelEngine-Group. UCM: Unified cache management – persist and reuse KV cache to speedup your LLM. https://github.com/ModelEngine-Group/unified-cache-management, 2025.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024. URL https://arxiv.org/abs/2308.11483.

Siddhant Ray, Rui Pan, Zhuohan Gu, Kuntai Du, Shaoting Feng, Ganesh Ananthanarayanan, Ravi Netravali, and Junchen Jiang. METIS: fast quality-aware RAG systems with configuration adaptation. In Youjip Won, Youngjin Kwon, Ding Yuan, and Rebecca Isaacs (eds.), *Proceedings of the ACM SIGOPS 31st Symposium on Operating Systems Principles, SOSP 2025, Lotte Hotel World, Seoul, Republic of Korea, October 13-16, 2025*, pp. 606–622. ACM, 2025. doi: 10.1145/3731569.3764855. URL https://doi.org/10.1145/3731569.3764855.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020. URL https://aclanthology.org/2020.acl-main.442/.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023. URL https://arxiv.org/abs/2310.18018.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.11324.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Yixuan Tang and Yi Yang. MultiHop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries. In *First Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=t4eB3zYWBK.

Yufei Tao, Adam Hiranandani, Siyuan Wang, and Bryan Hooi. When context leads but parametric memory follows in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. URL https://aclanthology.org/2024.emnlp-main.234/.

Xin Teng, Canyu Zhang, Shaoyi Zheng, Danyang Zhuo, Tianyi Zhou, and Shengjie Wang. InfoFlow KV: Information-flow-aware KV recomputation for long context, 2026. URL https://arxiv.org/abs/2603.05353.

C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979. ISBN 0-408-70929-4.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Jiahao Wang, Weiyu Xie, Mingxing Zhang, Boxing Zhang, Jianwei Dong, Yuening Zhu, Chen Lin, Jinqi Tang, Yaochen Han, Zhiyuan Ai, Xianglin Chen, Yongwei Wu, and Congfeng Jiang. From prefix cache to fusion rag cache: Accelerating llm inference in retrieval-augmented generation. *arXiv preprint arXiv:2601.12904*, 2026a. URL https://arxiv.org/abs/2601.12904.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2022.

Qian Wang, Zahra Yousefijamarani, Morgan Lindsay Heisler, Rongzhi Gu, Xiaolong Bai, Yizhou Shan, Wei Zhang, Wang Lan, Ying Xiong, Yong Zhang, and Zhenan Fan. MEPIC: Memory efficient position independent caching for LLM serving, 2025a. URL https://arxiv.org/abs/2512.16822.

Shihao Wang, Jiahao Chen, Yanqi Pan, Hao Huang, Yichen Hao, Xiangyu Zou, Wen Xia, Wentao Zhang, Chongyang Qiu, and Pengfei Wang. Prophetkv: User-query-driven selective recomputation for efficient kv cache reuse in retrieval-augmented generation. *arXiv preprint arXiv:2602.02579*, 2026b. URL https://arxiv.org/abs/2602.02579.

Zhengchao Wang, Yitao Hu, Jianing Ye, Zhuxuan Chang, Jiazheng Yu, Youpeng Deng, and Keqiu Li. RAGPulse: An open-source RAG workload trace to optimize RAG serving systems, 2025b. URL https://arxiv.org/abs/2511.12979.

Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V. Le. Simple synthetic data reduces sycophancy in large language models. *arXiv preprint arXiv:2308.03958*, 2023. URL https://arxiv.org/abs/2308.03958.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Sandeep Singh Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. Livebench: A challenging, contamination-limited llm benchmark. *arXiv preprint arXiv:2406.19314*, 2024. URL https://arxiv.org/abs/2406.19314.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. URL https://aclanthology.org/2024.findings-acl.372/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang,

Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025a. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Bin Yang, Qiuyu Leng, Jun Zeng, and Zhenhua Wu. Cacheclip: Accelerating RAG with effective KV cache reuse. *CoRR*, abs/2510.10129, 2025b. doi: 10.48550/ARXIV.2510.10129. URL https://doi.org/10.48550/arXiv.2510.10129.

Jingbo Yang, Bairu Hou, Wei Wei, Yujia Bao, and Shiyu Chang. Kvlink: Accelerating large language models via efficient KV cache reuse. *CoRR*, abs/2502.16002, 2025c. doi: 10.48550/ARXIV.2502.16002. URL https://doi.org/10.48550/arXiv.2502.16002.

Xinyu Yang, Tianqi Chen, and Beidi Chen. APE: faster and longer context-augmented generation via adaptive parallel encoding. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025d. URL https://openreview.net/forum?id=yUC8pU508S.

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*, EuroSys '25, pp. 94–109, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400711961. doi: 10.1145/3689031.3696098. URL https://doi.org/10.1145/3689031.3696098.

Hancheng Ye, Zhengqi Gao, Mingyuan Ma, Qinsi Wang, Yuzhe Fu, Ming-Yu Chung, Yueqian Lin, Zhijian Liu, Jianyi Zhang, Danyang Zhuo, and Yiran Chen. KVCOMM: Online cross-context KV-cache communication for efficient LLM-based multi-agent systems. *arXiv preprint arXiv:2510.12872*, 2025.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. ∞bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL https://aclanthology.org/2024.acl-long.814/.

Zhisong Zhang, Yan Wang, Xinting Huang, Tianqing Fang, Hongming Zhang, Chenlong Deng, Shuaiyi Li, and Dong Yu. Attention entropy is a key factor: An analysis of parallel context encoding with full-attention-based pre-trained language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. URL https://arxiv.org/abs/2412.16545.

Shiju Zhao, Junhao Hu, Jiaqi Zheng, and Guihai Chen. You need an encoder for native position-independent caching, 2026. URL https://arxiv.org/abs/2602.01519.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024a. URL https://arxiv.org/abs/2309.03882.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark W. Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024b. URL http://papers.nips.cc/paper_files/paper/2024/hash/724be4472168f31ba1c9ac630f15dec8-Abstract-Conference.html.

Yuechi Zhou, Yi Su, Jianxin Zhang, Juntao Li, Qingrong Xia, Zhefeng Wang, Xinyu Duan, and Baoxing Huai. A^3: Attention-aware accurate KV cache fusion for fast large language model serving, 2025. URL https://arxiv.org/abs/2511.17560.

Table 5: Use of LongBench-QA datasets in state-of-the-art papers on KV cache reuse.

Paper	HotpotQA	2WikiMQA	MuSiQue
CacheBlend (Yao et al., 2025)	✓	✓	✓
Cache-Craft (Agarwal et al., 2025)	×	✓	✓
CacheClip (Yang et al., 2025b)	✓	✓	✓
EPIC (Hu et al., 2024)	✓	✓	✓
SamKV (Cao et al., 2025)	✓	✓	✓
ContextPilot (Jiang et al., 2025)	×	×	×
FusionRAG (Wang et al., 2026a)	✓	✓	✓
Droidspeak (Liu et al., 2026b)	✓	✓	✓
ProphetKV (Wang et al., 2026b)	✓	✓	✓
CacheSlide (Liu et al., 2026a)	✓	×	×
KVLink (Yang et al., 2025c)	✓	✓	✓
TurboRAG (Lu et al., 2024)	✓	✓	✓
BlockAttention (Ma et al., 2025)	✓	✓	✓
APE (Yang et al., 2025d)	✓	✓	✓
CacheGen (Liu et al., 2024)	✓	✓	✓
LMCache (Liu et al., 2025)	✓	✓	✓
A^3 (Zhou et al., 2025)	✓	✓	✓
COMB (Zhao et al., 2026)	✓	✓	✓
EVICPRESS (Feng et al., 2025)	✓	✓	✓
InfoFlow KV (Teng et al., 2026)	✓	✓	✓
SEL (Zhang et al., 2025)	✓	×	×
MEPIC (Wang et al., 2025a)	×	×	×
Cestola et al. (2026)	×	✓	✓
Papers using benchmark	19/23	19/23	19/23

A Appendix - Prevalence of LongBench Datasets

Table 5 shows which LongBench-QA datasets are used in the evaluation of state-of-the-art papers that investigate KV cache reuse. We find that such datasets are used each in more than 80% of the papers we reviewed. Other datasets that are used with less frequency are TriviaQA (Joshi et al., 2017), Qasper (Dasigi et al., 2021), NarrativeQA (Kočiský et al., 2018), Ruler (Hsieh et al., 2024), MultihopRag (Tang & Yang, 2024). These datasets present each some nondesirable property for evaluating KV cache reuse in the RAG case, e.g., being single-hop, or having a single long chunk with multiple (similar and clustered) queries over it, or having only very few (e.g., 3) long chunks as context. As said in Section 2 we do not consider summarization tasks because there is no reuse in those datasets, and agentic workload datasets because they target different re-use dynamics.

B Extended related work

This appendix expands the second part of Section 6.

A line of work counters dataset contamination through synthetic data or fresh sourcing: InfiniteBench (Zhang et al., 2024) replaces core entities in real books with fictional ones, and LiveBench (White et al., 2024) sources fresh questions over time, both to prevent answers from leaking via parametric memory. Synthetic-corpus generation has also been used for adjacent purposes, e.g., training compact long-context representations (Eyuboglu et al., 2025) or surfacing specific behaviors such as sycophancy (Wei et al., 2023).

Work on natural language inference shows that aggregate accuracy can be inflated by queries the model answers from dataset artifacts rather than from actual reasoning about the query (Gururangan et al., 2018; McCoy et al., 2019). Work on benchmark contamination (Sainz et al., 2023; Magar & Schwartz, 2022) documents the analogous effect of test items leaking into pretraining, where aggregate scores are inflated by memorisation. Hooker et al. (2020) show how a subset of queries can disproportionately affect an aggregate metric when evaluating compressed models. Ribeiro et al. (2020) make the same case for natural language processing evaluation more broadly, arguing that a single aggregate score collapses distinct capabilities and proposing per-capability tests. In the RAG setting, RGB (Chen et al., 2024) stratifies accuracy into noise robustness, negative rejection, information integration, and counterfactual robustness, rather than a single aggregate.

Prior work has measured when retrieval helps vs. when parametric memory suffices (Mallen et al., 2023), quantified the relative contribution of context vs. parameters in long-form generation (Tao

et al., 2024), and used entity substitution to force context reading (Longpre et al., 2021). MIRAGE (Xiong et al., 2024) reports a no-retrieval baseline alongside RAG accuracy on medical QA.

C Appendix - Boxoffice

C.1 Example Prompt Components

We show below the exact prompt components used by the `Boxoffice` seeds: the instruction/system prompt chunk, one example of movie dossier chunk, and the final question suffix. The text is taken verbatim from a `s7` seed row. Each query includes the instruction chunk, then 10 dossier chunk, and finally the question suffix.

Instruction chunk

```
Use only the synthetic movie dossiers below.  Ignore outside/world
knowledge.
For ranking, compare only the BOX_OFFICE_MUSD integer fields across
all named
candidates.  Return exactly one FILM-ID and no other text.
```

Example dossier chunk

```
BENCHMARK_DOSSIER
ENTITY_ID: FILM-1008
TITLE: The Midnight Ledger
DIRECTOR: Elara Vance
RELEASE_YEAR: 2009
STARLIGHT_AWARDS: 4
BOX_OFFICE_MUSD: 966
GENRE: Sci-Fi
STUDIO: Aperture Seven
COUNTRY: Norland
RUNTIME_MIN: 142
CAST: Lyra Boone, Sage Monroe, Rin Carter

CATALOG_ENTRY
Overview:  The Midnight Ledger is a 2009 sci-fi feature directed by Elara
Vance for Aperture Seven.  In short directory listings, the film is usually
identified through its Norland market line and its association with Novastad.
Cast and setting:  The credited cast is Lyra Boone, Sage Monroe, Rin Carter.
Reference copy keeps Novastad attached to the title as its standing location
marker rather than expanding into a long plot synopsis.
Release profile:  The entry presents the film as a full-length commercial
feature with a runtime of 142 minutes.  The dossier keeps the release identity
tied to studio, market, cast, and setting in the style of a compact film
guide.
Commercial note:  The recorded box office for The Midnight Ledger is 966 million
USD, and the listing treats that figure as part of the title's market profile
rather than as a review note.

RELEASE_AND_REFERENCE_NOTES
SETTING_CITY: Novastad
SETTING_CITY_POPULATION_MIL: 9.2
Studio filing:  Aperture Seven remains the credited company throughout the
record.
Market filing:  Norland is the territory used for release and indexing
reference.
Cast filing:  Lyra Boone, Sage Monroe, Rin Carter remains the recurring cast
line attached to the title.
Setting filing:  Novastad is the place-name consistently used to anchor the
film in short reference prose.
Directory note:  The title is grouped with current genre features rather than
with repertory or specialty listings.
Exhibition note:  The entry reads like a film associated with standard
multiplex play.
Reference note:  The record keeps to production identity, market placement,
cast, setting, runtime, and gross without adding review language.
Index note:  The Midnight Ledger is filed in a director-led pattern that makes
the title easy to locate in long title lists.
Program note:  The note block favors release and reference detail over plot
description.
Library note:  The overall effect is closer to a compact distribution-reference
record than to a pressbook or capsule review.
Release note:  The title is presented as a studio-backed sci-fi feature rather
than as a limited event item.
Record line:  The runtime and release year remain part of the short production
summary for the film.
```

Question suffix

```
Question:  Valid candidates:  FILM-1008, FILM-1022, FILM-1002,
FILM-2000,
FILM-2001, FILM-1000, FILM-1044, FILM-2002, FILM-1032, FILM-2003
Which valid FILM-ID has the maximum BOX_OFFICE_MUSD among all listed
candidates?
Read the BOX_OFFICE_MUSD field from the dossiers; do not infer values
from
titles.
Return exactly one valid FILM-ID.
```

C.2 WARMUP CONFIGURATIONS

Table 6: Configurations of a `Boxoffice` query, by the role each chunk played during warmup (Winner vs Nonwinner).

Configuration	Winner was	Nonwinners were	Description
`aligned`	W	N	Every chunk's cached state matches its role in the eval query. The winner's cache encodes "I am the maximum"; each nonwinner's cache encodes "I am not". **Easy case.**
`all-W`	W	W	The winner's cache is aligned, but each nonwinner's cache also encodes "I am the maximum". The model must override the "winner" signal carried by nonwinner caches. **Medium case.**
`all-N`	N	N	Every nonwinner's cache is aligned, but the winner's encodes "I am not the maximum". The model must identify the winner despite every cache claiming nonwinner status. **Medium case.**
`flipped`	N	W	The signals are fully inverted relative to the eval roles. The model must recover the correct ranking from a reversed set of cached signals. **Hard case.**

Table 6 lists the four warmup configurations of Section 5, the cached signal each presents to the reuse method, and the qualitative difficulty it imposes.

C.3 DATASET GENERATION

Pseudocode for generating one `Boxoffice` seed. Algorithm 1 summarizes the workflow implemented in the released `Boxoffice` bundle for a single seed s.

Algorithm 1: Generate one `Boxoffice` seed s **Input:** seed s; film corpus; similarity metadata W; filter model set $\mathcal{M}$

Output: `full.jsonl`, `eval.jsonl`, manifest, validation counts

1: rebuild similarity metadata W from the corpus if missing
2: load film records; build warmup schedule with W/N roles
3: **for all** $c \in \{$`aligned`, `all-W`, `all-N`, `flipped`$\}$ **do**
4: oversample candidate 10-film queries using seed s
5: compute similarity statistics on candidates via W
6: retain retrieval-balanced candidate pool for c
7: **end for**
8: extract eval rows; build question-only copies
9: **for all** $m \in \mathcal{M}$ **do**
10: run full-context and question-only baselines on eval rows
11: **end for**
12: keep rows with full-context $F_1{=}1$ and question-only $F_1{\neq}1$ for *every* $m \in \mathcal{M}$
13: per configuration, retain a fixed number of surviving rows
14: augment warmup so every eval chunk has ≥ 2 cached *smaller-than* and ≥ 2 *greater-than* predecessor configurations
15: concatenate augmented warmup with filtered eval; reindex
16: write `full.jsonl`, `eval.jsonl`, manifest, and validation counts

Steps 9–12 implement the selection that preserves the "meaningful by construction" property by keeping only rows that all filter models solve with full context but not without context. Step 14 realizes the directional warmup balancing discussed in Section 5: every evaluation chunk is given both "smaller-than" and "greater-than" cached predecessors, which yields the balanced seeds used in the paper.

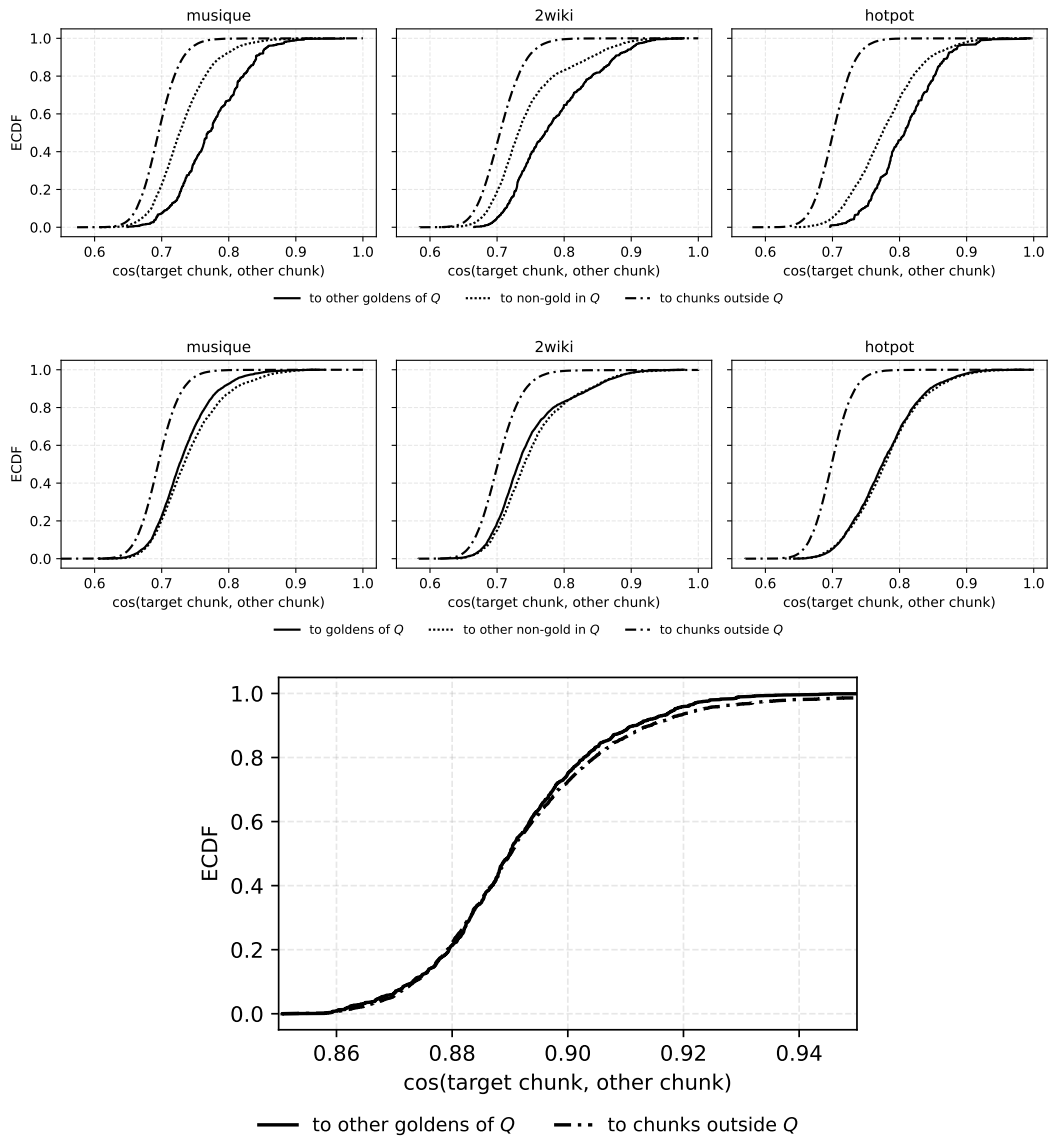


Figure 2: **Top**: similarity between a *golden* target chunk in query Q of LongBench-QA w.r.t. other golden chunks Q, other non-golden chunks in Q, and other chunks in the dataset. **Middle**: same as top but done for target *non-golden* chunks. **Bottom**: similarity between a chunk in `Boxoffice` in Q w.r.t. other chunks in Q and other chunks in the dataset.

C.4 Chunk distance CDFs

Figure 2 shows the distribution of the distances between chunk in the context of a query Q and other chunks in the context of Q or outside of Q. The main result is that in LongBench-QA golden chunks are clearly closer to other chunks in the same query, and especially golden (top). A non-golden chunk that appear in Q is also more close to other chunks Q than to other chunks. Hence, LongBench-QA is highly clustered. On the contrary, chunks in `Boxoffice` (which are all golden ones) have the same similarity distribution towards any other chunk in the dataset.

C.5 Boxoffice additional evaluation

Table 7 extends the results of Section 5.2 for $R = 0.15$ with standard deviation information per cell, computed on the results of 10 seeds. Table 8 the result obtained on `Boxoffice` for $R = 0.05$. The most interesting result is that at $R = 0.05$, the winner on average for each model is $LM + Q$. The reason is that at $R = 0.05$ the information encoded in the KV cache during warmup cannot

Table 7: Norm-F1 on `Boxoffice` at R=0.15, mean±sample std across 10 generated datasets (seeds 7, 11, 13, 17, 19, 23, 29, 31, 47, 73). Within each (model, configuration), the winning method is in bold.

	Config	CB	CB+Q	FR	FR+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q
Qwen	aligned	.39 ± .07	.09 ± .05	.44 ± .06	.40 ± .10	**.97 ± .03**	.94 ± .05	.17 ± .13	.38 ± .11	.18 ± .09	.25 ± .10
	all-W	.30 ± .06	.12 ± .06	.28 ± .07	.29 ± .09	.11 ± .06	.25 ± .05	.13 ± .10	.27 ± .08	.24 ± .11	**.32 ± .07**
	all-N	.39 ± .06	.20 ± .07	.30 ± .06	**.39 ± .11**	.05 ± .03	.11 ± .03	.18 ± .09	.36 ± .11	.24 ± .09	.34 ± .07
	flipped	**.41 ± .12**	.18 ± .07	.23 ± .07	.28 ± .06	.00 ± .00	.06 ± .04	.15 ± .08	.29 ± .09	.28 ± .11	.34 ± .10
	average	**.37 ± .05**	.15 ± .03	.31 ± .04	.34 ± .04	.28 ± .02	.34 ± .02	.16 ± .05	.33 ± .05	.24 ± .04	.31 ± .04
Llama	aligned	.10 ± .03	.31 ± .09	.11 ± .05	.37 ± .07	**.98 ± .02**	.95 ± .03	.17 ± .11	.55 ± .11	.19 ± .06	.51 ± .08
	all-W	.06 ± .04	.21 ± .03	.05 ± .04	.28 ± .06	.14 ± .08	.29 ± .07	.13 ± .11	.42 ± .12	.24 ± .09	**.50 ± .12**
	all-N	.06 ± .05	.25 ± .12	.13 ± .06	.43 ± .08	.07 ± .04	.55 ± .06	.16 ± .10	.57 ± .13	.23 ± .09	**.57 ± .10**
	flipped	.09 ± .04	.25 ± .08	.06 ± .04	.35 ± .08	.00 ± .00	.09 ± .06	.11 ± .08	.41 ± .13	.23 ± .11	**.54 ± .10**
	average	.08 ± .03	.26 ± .04	.09 ± .02	.36 ± .03	.30 ± .03	.47 ± .03	.14 ± .05	.49 ± .07	.22 ± .04	**.53 ± .06**
Mistral	aligned	.20 ± .08	.18 ± .08	.33 ± .06	.53 ± .10	**.98 ± .02**	.94 ± .03	.14 ± .10	.42 ± .13	.21 ± .08	.36 ± .10
	all-W	.24 ± .06	.15 ± .07	.15 ± .08	.26 ± .06	.17 ± .05	.26 ± .09	.14 ± .09	.26 ± .08	.31 ± .09	**.34 ± .07**
	all-N	.29 ± .08	.16 ± .07	.24 ± .07	**.57 ± .09**	.09 ± .04	.29 ± .09	.16 ± .08	.34 ± .15	.32 ± .15	.39 ± .14
	flipped	.25 ± .09	.11 ± .05	.18 ± .05	**.34 ± .10**	.00 ± .00	.02 ± .02	.13 ± .09	.23 ± .14	.30 ± .13	.30 ± .10
	average	.24 ± .03	.15 ± .04	.23 ± .03	**.42 ± .04**	.31 ± .02	.38 ± .04	.14 ± .05	.31 ± .06	.28 ± .07	.35 ± .05

Table 8: norm-F1 on `Boxoffice` at R=0.05, mean±sample std across 10 generated datasets (seeds 7, 11, 13, 17, 19, 23, 29, 31, 47, 73). Within each (model, configuration), the winning method is in bold.

	Config	CB	CB+Q	FR	FR+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q
Qwen	aligned	.03 ± .02	.07 ± .04	.36 ± .05	.34 ± .06	.95 ± .02	**.97 ± .02**	.13 ± .13	.18 ± .12	.12 ± .07	.10 ± .05
	all-W	.04 ± .03	.07 ± .03	**.23 ± .05**	.23 ± .04	.11 ± .05	.18 ± .05	.10 ± .07	.12 ± .08	.21 ± .08	.20 ± .06
	all-N	.07 ± .04	.09 ± .04	**.28 ± .04**	.28 ± .06	.04 ± .03	.08 ± .03	.14 ± .08	.15 ± .07	.15 ± .07	.18 ± .09
	flipped	.11 ± .06	.11 ± .06	.19 ± .05	**.19 ± .05**	.00 ± .00	.00 ± .00	.12 ± .08	.14 ± .07	.19 ± .09	.19 ± .08
	average	.07 ± .02	.09 ± .02	.27 ± .03	.26 ± .02	.28 ± .02	**.31 ± .01**	.12 ± .04	.15 ± .04	.17 ± .03	.17 ± .03
Llama	aligned	.11 ± .06	.12 ± .06	.09 ± .03	.09 ± .05	**.98 ± .02**	.96 ± .03	.13 ± .11	.21 ± .11	.10 ± .04	.16 ± .07
	all-W	.04 ± .04	.06 ± .03	.04 ± .03	.05 ± .03	.13 ± .08	.18 ± .05	.11 ± .10	.15 ± .10	.19 ± .09	**.22 ± .11**
	all-N	.11 ± .05	.08 ± .04	.15 ± .06	.12 ± .05	.04 ± .03	**.46 ± .08**	.14 ± .10	.21 ± .11	.15 ± .09	.20 ± .09
	flipped	.04 ± .03	.07 ± .04	.05 ± .03	.05 ± .03	.00 ± .00	.01 ± .01	.07 ± .04	.14 ± .08	.16 ± .08	**.19 ± .09**
	average	.07 ± .03	.08 ± .02	.08 ± .02	.08 ± .02	.29 ± .03	**.40 ± .03**	.11 ± .04	.18 ± .04	.15 ± .03	.19 ± .04
Mistral	aligned	.07 ± .05	.09 ± .04	.35 ± .07	.34 ± .09	.98 ± .02	**.98 ± .02**	.11 ± .10	.23 ± .11	.17 ± .06	.19 ± .08
	all-W	.05 ± .04	.05 ± .03	.13 ± .06	.15 ± .06	.17 ± .06	.22 ± .08	.11 ± .08	.14 ± .08	.23 ± .07	**.25 ± .09**
	all-N	.08 ± .02	.17 ± .05	.21 ± .04	**.26 ± .06**	.04 ± .02	.14 ± .04	.13 ± .10	.20 ± .12	.23 ± .11	.24 ± .14
	flipped	.03 ± .01	.04 ± .03	.16 ± .04	.19 ± .05	.00 ± .00	.00 ± .01	.10 ± .06	.14 ± .09	**.22 ± .10**	.21 ± .10
	average	.06 ± .02	.09 ± .02	.21 ± .02	.23 ± .03	.29 ± .02	**.33 ± .03**	.11 ± .04	.18 ± .05	.21 ± .05	.22 ± .06

be adequately overwritten by means of selective recomputation, and hence decisively influence the accuracy achieved. $LM + Q$ achieves near-perfect accuracy on the `aligned` cells, which skews the average results.

We report that running an end-to-end experiment from dataset generation to evaluation on all the involved methods takes roughly 5 hours on an accelerator with ∼376 TFLOPS BF16 dense compute, 64 GB HBM2E, and ∼1.6 TB/s memory bandwidth per device.

Figure 3 provides additional insights on the accuracy dynamics (top) for warm online methods, describing how additional versions available for the KV cache of a chunk may change the configuration of a query from aligned to nonaligned/mixed (left) and from nonaligned to mixed/aligned.

For the aligned case, when $M > 1$ it suffices that a single KV cache is picked with the wrong warmup conditioning to shift a query from aligned to mixed. With 10 chunks in a query, even $M = 2$ suffices to break the perfect alignment of more than 75% of the queries, which explains the sudden drop in accuracy. For the nonaligned case the dual happens: having more versions to pick from enables CacheCraft to turn some nonaligned queries to mixed ones, which partially recovers accuracy.

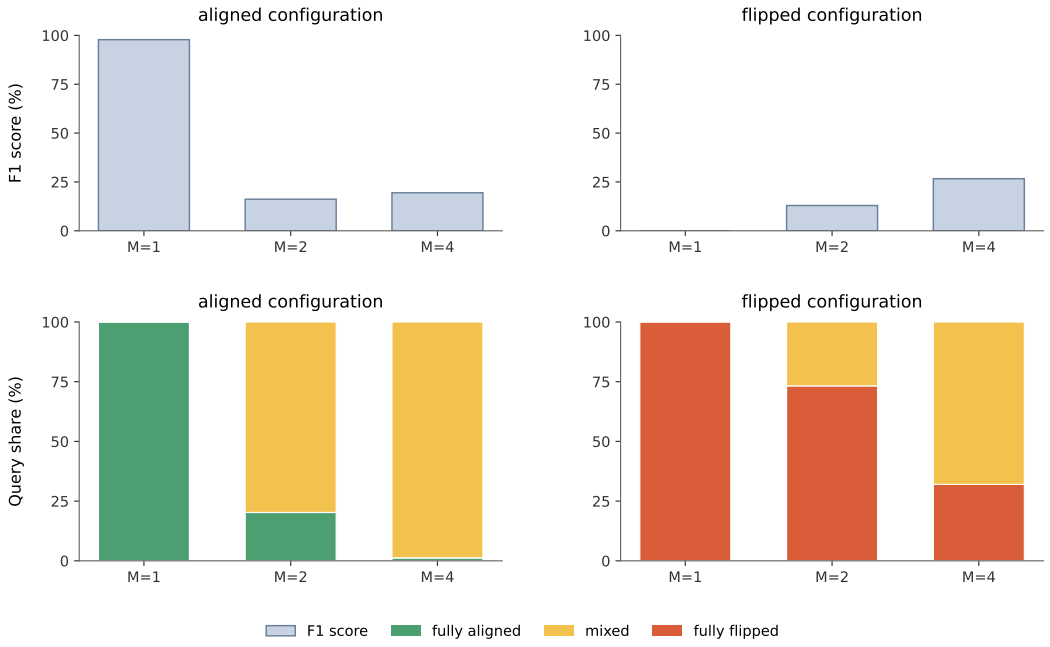


Figure 3: LMCache ($M = 1$) and CacheCraft ($M = 2, 4$) performance and query-level orientation breakdown on the ten `Boxoffice` seeds at recompute ratio R=0.15. The top row shows mean F1 across all three models for the aligned and flipped configurations as the number of stored variants grows from M=1 to M=4. The bottom row classifies each evaluation query by the orientation of the selected versions across all 10 chunks: *fully aligned* means every selected winner/nonwinner version has the role-consistent orientation, *mixed* means the query contains both properly oriented (aligned) and misoriented (flipped) selected versions, and *fully flipped* means every selected version is misoriented (flipped).

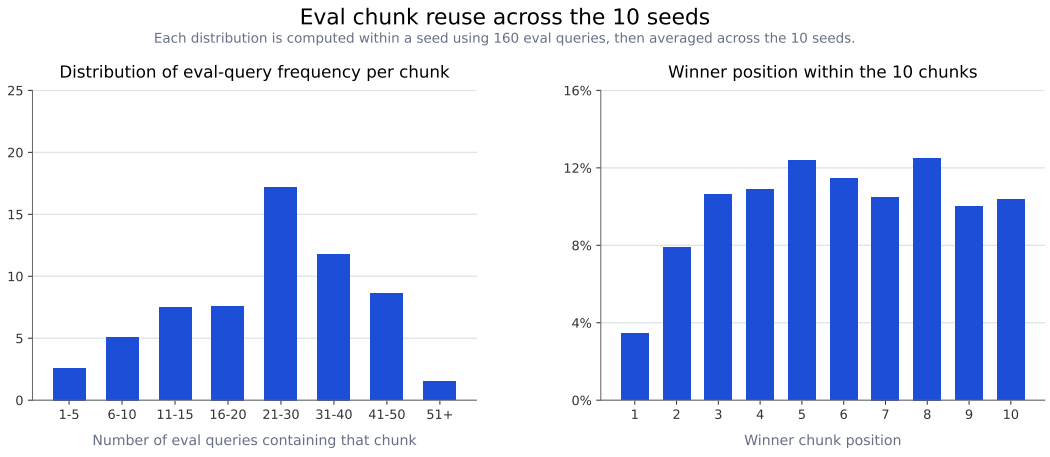


Figure 4: Evaluation queries chunk reuse structure on ten `Boxoffice` seeds. The left panel shows the distribution of eval query frequencies per unique chunk, split into winner chunks and nonwinner chunks. The right panel shows the winner's position among the 10 candidate chunks.

D REUSE CHARACTERIZATION

Figure 4 provides additional information about the reuse of the evaluation chunks. For each seed, chunk frequency is computed from its 160 evaluation queries, and the plotted distributions are then averaged across seeds. Figure 4 (left) shows for each unique chunk in the evaluation set, how many distinct evaluation queries contain it: the x-axis bins chunks by their evaluation-query frequency, and the y-axis counts how many unique chunks fall in each bin. The plot shows that there is a high degree of reuse of chunks. The plot does not reflect the warmup queries, which already guarantee, in our experiments, that each evaluation-chunk appears at least $M = 4$ times. Figure 4 (right) shows that winner chunks are placed at arbitrary positions in the prompts so as to avoid positional bias.

E OFFLINE WARMUP DETAILS

Figure 5 provides details on the conditioning of KV caches in FR. Both winners and nonwinners are conditioned to a mix of chunks with larger and smaller box office values, so FR has no clear advantages in any query configurations.

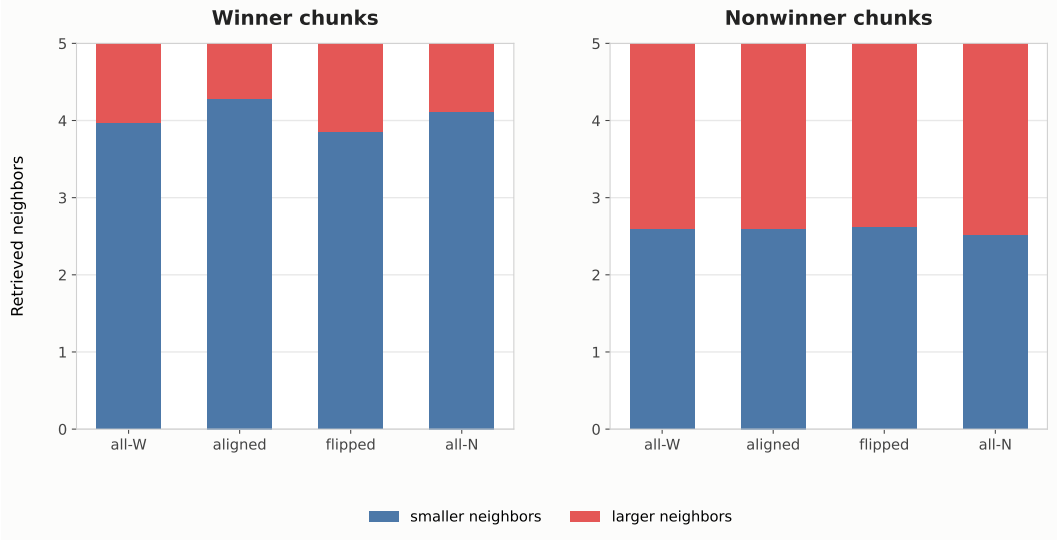


Figure 5: Warm offline retrieval conditioning composition on the ten seeds `Boxoffice` data as done in FR. The stacked bars show the average composition of the retrieved warm neighbors for k=5, split into smaller versus larger box-office movies relative to the target chunk. Winner chunks by construction have relatively high box office value, so they retrieve mostly smaller neighbors in every configuration. Nonwinner chunks have a more balanced neighborhood. In both cases, the KV cache encodes mixed information about the ranking of a chunk with respect to its neighbors.

F SENSITIVITY TO THE NO-CONTEXT THRESHOLD AND SIZE OF THE MEANINGFUL SUBSET

The NC filter of Section 4.1 classifies a query as answerable without context when the model scores $F_1 > \tau$ on the question alone, with τ=0.2 in the main text. Table 9 reports ρ_{F_1}, averaged over the nine dataset×model cells, for τ between 0 and 0.5. ρ remains well below 1 for every method at every threshold, and grows slowly with τ because a looser threshold filters out fewer queries, so the filtered and unfiltered datasets overlap more. The conclusion of Section 4.1, that measuring F_1 on the unfiltered dataset inflates the reported accuracy preservation, therefore does not depend on the choice of τ.

Table 10 reports how many of the 200 queries of each LongBench-QA dataset survive each filter, per model. The meaningful subset is model-dependent by construction: accuracy preservation is defined relative to a specific model's full-prefill ceiling, so a query that one model answers from parametric memory may be meaningful for another. All cross-method comparisons in this paper are within-model and are therefore unaffected; to compare across models on a common set, one can take

Table 9: Sensitivity of ρ_{F_1} (mean over the 9 dataset×model cells, R=0.15) to the no-context threshold τ that defines the NC filter. Left: after the NC filter; right: after NC and LI, i.e., on the meaningful subset. The row for B_0 alone does not depend on τ: CB 0.87, CB+Q 0.85, FR 0.83, FR+Q 0.83. The paper uses τ=0.2.

	+NC				+LI (meaningful subset)			
τ	CB	CB+Q	FR	FR+Q	CB	CB+Q	FR	FR+Q
0.00	0.81	0.74	0.76	0.78	0.80	0.72	0.73	0.77
0.05	0.81	0.74	0.76	0.78	0.80	0.72	0.73	0.77
0.10	0.82	0.74	0.77	0.78	0.80	0.73	0.73	0.77
0.15	0.82	0.75	0.76	0.78	0.81	0.73	0.74	0.77
0.20 (paper)	0.82	0.75	0.77	0.78	0.81	0.73	0.74	0.77
0.25	0.82	0.76	0.77	0.79	0.81	0.74	0.75	0.79
0.30	0.82	0.76	0.78	0.79	0.81	0.74	0.75	0.79
0.35	0.82	0.77	0.78	0.80	0.82	0.76	0.76	0.79
0.40	0.82	0.77	0.78	0.80	0.81	0.76	0.76	0.79
0.50	0.83	0.78	0.80	0.81	0.82	0.78	0.78	0.81

Table 10: Number of queries that survive each filter on LongBench-QA (200 queries per dataset). B_0 is independent of τ; +NC and +LI are reported at the paper's τ=0.2, and the size of the meaningful subset (+LI) is additionally reported for $\tau \in \{0.1, 0.3, 0.5\}$.

				τ=0.2		+LI at other τ		
dataset	model	all	B_0	+NC	+LI	0.1	0.3	0.5
MuSiQue	Qwen	200	79	47	47	44	52	69
	Llama	200	79	57	57	52	60	70
	Mistral	200	68	55	55	55	55	64
2WikiMQA	Qwen	200	96	42	38	37	41	51
	Llama	200	114	65	46	43	50	57
	Mistral	200	127	81	58	57	59	67
HotpotQA	Qwen	200	136	68	65	64	68	84
	Llama	200	134	70	65	60	66	79
	Mistral	200	127	80	74	73	75	84

the intersection of the per-model meaningful subsets. `Boxoffice` sidesteps the issue: its generator retains only the queries that every filter model answers with full context and none answers without it (Algorithm 1, steps 9–12), so each released dataset is a single evaluation set shared by all models, and generating a set that is meaningful for a new group of models is fully automated.

G LARGER MODELS: QWEN3-32B

Accuracy inflation. To check whether the inflation documented in Section 4.1 survives at larger scale, we re-ran the evaluation of Table 1 with Qwen3-32B on MuSiQue. Table 11 reports ρ_{F_1} and the B_0 ablation. Out of 200 queries, 100 are answerable with full prefill, 68 of them are also grounded, and none is low-information, so the meaningful subset has 68 queries. ρ_{F_1} stays below 1 for every method, at τ=0.2 as well as at τ=0.5, and the share of the aggregate F_1 that comes from baseline-zero queries ranges from 9% to 24%. The inflation is attenuated with respect to the 8B models (compare with the MuSiQue rows of Table 1), as expected: a larger model answers correctly from the context a larger fraction of the queries, so fewer unanswerable rows remain to inflate the metric. The inflation is thus not an artifact of small models; it persists at 4× the scale.

Cache staleness. We also ran the `Boxoffice` protocol of Section 5.2 with Qwen3-32B (Table 12). The staleness dynamic is unchanged at this scale: on `aligned`, LM matches the full-prefill baseline (norm-F1 $\geq$ 1), on `flipped` it scores 0.00, and the mixed configurations sit close to `flipped`, while the cold method CB is flat across configurations. Multi-version caching with M=4 lifts the three non-aligned configurations (0.33–0.42 for CC-$M4$ and CC-$M4$+Q, versus 0.00–0.11 for LM) at the cost of the aligned one, mirroring Result 4 of Section 5.2; query-driven token selection helps LM marginally and hurts the cold method on this model, mirroring Result 3.

Table 11: Qwen3-32B on MuSiQue at R=0.15 (200 queries; 100 survive B_0, 68 survive +NC and +LI at τ=0.2). Left: ρ_{F_1} under the progressive filters at τ=0.2 and at τ=0.5. Right: B_0 ablation, as in Table 1.

	τ=0.2			τ=0.5		B_0 ablation		
method	B_0	+NC	+LI	+NC	+LI	B_0M_0	B_0M_+	$\%F_1$
CB	0.86	0.82	0.82	0.82	0.82	82	18	13.8
CB+Q	0.76	0.72	0.72	0.71	0.71	73	27	23.6
FR	0.91	0.86	0.86	0.89	0.89	81	19	9.3
FR+Q	0.88	0.81	0.81	0.85	0.85	76	24	11.7

Table 12: norm-F1 of Qwen3-32B on the `Boxoffice` argmax template at R=0.15, per warmup configuration. CB and LM are averaged over 6 generated datasets (seeds 7, 11, 13, 17, 19, 23); the remaining methods over the 3 datasets on which they were run (seeds 7, 11, 13). FR is not reported because its offline pool was not built for this model. The datasets were filtered with the three 8B models, so the Qwen3-32B full-prefill baseline is not exactly 1 (mean F_1 between 0.94 and 0.98 per dataset) and norm-F1 can slightly exceed 1.

Config	CB	CB+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q
aligned	.26	.08	1.03	**1.04**	.10	.13	.19	.27
all-W	.28	.07	.11	.14	.21	.21	.39	**.42**
all-N	.35	.05	.08	.09	.16	.19	**.37**	.33
flipped	.33	.05	.00	.00	.14	.15	**.41**	.34
average	.31	.06	.31	.32	.15	.17	.34	**.34**

H ADDITIONAL QUERY TEMPLATES

The main text evaluates the default `Boxoffice` template, an argmax over a numeric field. To verify that the dynamics we report are not specific to numeric comparison, we implemented two further templates over the same fictional corpus and with the same generation pipeline (joint three-model filtering, 40 evaluation queries per configuration, warmup balanced to M=4); the generators are released in the `templates_ooak_pj` folder of the code repository and the generated datasets in the `templates` folder of the dataset release (Anonymous Authors, 2026a;b).

Genre uniqueness. The query asks to identify the only candidate whose genre is not shared by any other candidate in the context. Answering requires equality matching across all candidates and uniqueness detection, rather than a numeric comparison. The warmup roles are defined by genre: a chunk cached after predecessors none of which shares its genre carries the signal "I am unique among my predecessors", and a chunk cached after at least one same-genre predecessor carries the opposite signal. With g the unique-genre candidate, the four configurations are: `aligned`, g was cached with no same-genre predecessor and every other candidate was cached with at least one same-genre predecessor; `flipped`, the opposite; `all-W`, every candidate was cached with no same-genre predecessor; `all-N`, every candidate was cached with at least one same-genre predecessor.

Pivot join. The query names a *pivot* movie in the context and asks for the other candidate directed by the same director; exactly one such candidate exists. This is a strictly two-hop task: the model must locate the pivot's dossier, read its director, and then find the candidate that matches it, so the second step depends on what is extracted in the first. The golden chunks are the pivot and its match, and a chunk cached after a same-director predecessor carries the signal that a match exists: `aligned`, pivot and match were both cached with at least one same-director predecessor and every other candidate was cached with none; `flipped`, the opposite; `all-W`, every candidate was cached with at least one same-director predecessor; `all-N`, every candidate was cached with none.

Results. Tables 13 and 14 report norm-F1 at R=0.15 on four generated datasets per template, for the same methods as Table 4. Both templates are intrinsically harder than the argmax one, so absolute values are lower, but the dynamics of Section 5.2 carry over. The staleness dynamic is reproduced: LM peaks on `aligned` (0.51–0.66 on genre uniqueness, 0.24–0.42 on pivot join) and collapses on `flipped` (0.00–0.05 and 0.01–0.06), whereas the cold method CB is flat across configurations, since its caches carry no warmup conditioning. Multi-version reuse (CC-$M2$, CC-$M4$) progressively recovers from the collapse, and query-driven token selection ($+Q$) improves every warm method on every model on average. Finally, as in Result 5 of Section 5.2, the best method differs across models.

Table 13: norm-F1 on the `Boxoffice` *genre-uniqueness* template at R=0.15, mean over 4 generated datasets (seeds 7, 11, 13, 17; 40 evaluation queries per configuration per seed). Within each (model, configuration), the winning method is in bold.

	Llama										Qwen										Mistral									
Config	CB	CB+Q	FR	FR+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q	CB	CB+Q	FR	FR+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q	CB	CB+Q	FR	FR+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q
aligned	.69	.40	.08	.31	.51	.68	.35	.65	.33	**.72**	.27	.09	.11	.34	.56	**.65**	.21	.36	.30	.35	.23	.07	.10	.14	.66	**.72**	.29	.42	.35	.43
all-W	**.72**	.49	.10	.31	.15	.60	.19	.65	.31	.61	.25	.12	.15	**.37**	.11	.15	.18	.29	.32	.28	.26	.09	.14	.19	.20	.31	.22	.28	.36	**.42**
all-N	.68	.38	.08	.23	.13	.66	.28	.66	.35	**.70**	.34	.17	.12	**.42**	.11	.39	.26	.34	.31	.39	.24	.08	.12	.18	.18	.27	.26	.36	.40	**.45**
flipped	.64	.44	.07	.24	.05	.52	.19	**.65**	.30	.59	.25	.16	.13	.30	.00	.09	.18	.31	**.32**	**.32**	.26	.08	.09	.17	.04	.08	.26	.33	.36	**.37**
average	**.68**	.43	.08	.27	.21	.61	.25	.65	.33	.66	.28	.13	.13	**.36**	.20	.32	.21	.33	.31	.33	.24	.08	.11	.17	.27	.35	.26	.35	.37	**.42**

Table 14: norm-F1 on the `Boxoffice` *pivot-join* template at R=0.15, mean over 4 generated datasets (seeds 7, 11, 13, 17; 40 evaluation queries per configuration per seed). Within each (model, configuration), the winning method is in bold.

	Llama										Qwen										Mistral									
Config	CB	CB+Q	FR	FR+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q	CB	CB+Q	FR	FR+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q	CB	CB+Q	FR	FR+Q	LM	LM+Q	CC-M2	CC-M2+Q	CC-M4	CC-M4+Q
aligned	**.62**	.59	.09	.21	.24	.52	.15	.40	.23	.46	.50	.29	.14	.29	.42	**.56**	.27	.47	.29	.46	.39	.08	.19	.25	.28	**.56**	.24	**.56**	.23	.49
all-W	**.60**	.56	.12	.20	.15	.39	.18	.45	.23	.53	**.52**	.26	.15	.23	.14	.21	.28	.49	.31	.43	.42	.08	.14	.26	.09	.31	.31	.54	.36	**.55**
all-N	.67	**.68**	.14	.23	.10	.38	.14	.49	.14	.51	**.48**	.25	.06	.27	.10	.44	.22	.42	.29	.44	.43	.08	.16	.31	.13	.46	.22	**.48**	.35	**.48**
flipped	**.63**	.59	.22	.23	.06	.34	.19	.52	.21	.54	.47	.31	.11	.21	.01	.08	.13	.32	.22	**.50**	.46	.08	.16	.31	.03	.32	.19	.52	.39	**.56**
average	**.63**	.61	.14	.22	.14	.40	.17	.46	.21	.51	**.49**	.28	.11	.25	.17	.32	.22	.43	.28	.46	.42	.08	.17	.28	.13	.41	.24	**.52**	.33	**.52**

I Evidence from production RAG workloads

`Boxoffice` generates, by design, workloads in which the same chunk recurs across queries while its context differs at every recurrence. We checked whether this regime is representative of deployed RAG systems using the industry-oriented data we are aware of: RAGPulse (Wang et al., 2025b), a production trace of a deployed RAG service; WixQA (Cohen et al., 2025), real enterprise support queries over a shared knowledge base; HERB (Choubey et al., 2025), an industry-authored enterprise-workflow benchmark; and the production-workload analysis published with Cache-Craft (Agarwal et al., 2025), whose underlying data is not released.

All four indicate that chunk reuse is pervasive. In RAGPulse, 97% of the retrieved-passage slots land on passages used by at least two requests; in WixQA, popular articles serve up to 14 distinct user queries; HERB exhibits pervasive cross-question reuse of golden evidence chunks; Cache-Craft reports that 75% of the chunks retrieved per query are reprocessed ones and that the top 5% of the chunks serve 60% of the requests. On the three multi-chunk resources (RAGPulse, about 6 retrieved passages per request; HERB, about 9 evidence chunks per question; Cache-Craft, 5–15 chunks per query), the reuse is not clustered: the same chunk keeps returning, but its context differs at every recurrence. In RAGPulse, requests that share a chunk have a median Jaccard similarity of 0.09 between their chunk sets; since RAGPulse releases only chunk hashes, the offline-similarity assumption of Section 4.2 cannot be tested on it. In HERB, where the text is available, the Jaccard similarity between questions that share a chunk is likewise about 0.1, and we tested the offline-similarity assumption directly: after embedding all 8,348 chunks, a chunk's actually co-used partner is among its top-10 offline nearest neighbors in only 7.9% of the 54,974 co-used pairs. Cache-Craft reports the same from the systems side: chunks recur constantly but identical combinations rarely do (the chunks of a query typically come from three or more past requests), and its production deployment stores up to 11 differently-contextualized cache variants of a single chunk. WixQA is excluded from this second analysis by construction: its answers draw on 1.15–1.29 articles on average, a near-single-chunk regime in which cross-chunk composition barely arises.

Production systems thus handle chunks that are highly prone to reuse and that are combined at query time in ways that are hard to predict offline. This is exactly the regime that `Boxoffice` generates by design: on the full `Boxoffice` workload (about 250 warmup and 160 evaluation queries per seed), 100% of chunk slots land on chunks used by at least two queries, and queries that share a chunk have a median Jaccard similarity of 0.11 in every seed, close to RAGPulse's 97% and 0.09.

The number of cached versions M, the mix of warmup configurations, the candidate-set size, and the recomputation budget R are settable to match such measured statistics.