SUDOKURAG: EFFICIENT RAG PREFILLING VIA COORDINATED INTRA- AND INTER-DOCUMENT ATTENTION SPARSITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) can incur substantial prefill overhead when multiple retrieved documents are incorporated into the model input. Existing approaches accelerate RAG prefilling through parallel encoding, but often require additional training or incur accuracy degradation. We observe that intra-doc attention patterns can be profiled offline and reused across prompts, whereas inter-doc selection must adapt to the current context. Building on this insight, we introduce SUDOKURAG, a training-free design that coordinates offline intra-doc masks and online inter-doc pruning within the tiled attention loop, using current prefix and intra-doc softmax statistics as guidance. Empirical results show that SUDOKURAG achieves 1.34–1.64× prefill-attention speedups over dense attention at approximately 50% sparsity while largely preserving task accuracy.

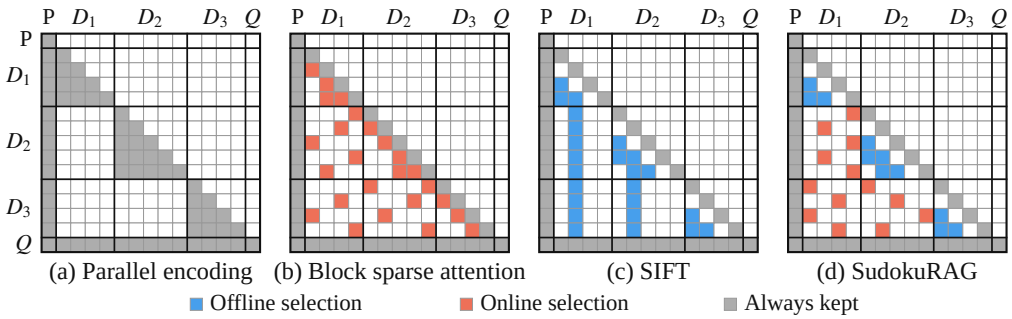


Figure 1: (a) **Parallel encoding** removes inter-doc interactions. (b) **Block sparse attention** selects within and across documents. (c) **SIFT** uses offline document profiles for both regions; selected inter-doc keys are fixed across receiving queries. (d) **SUDOKURAG** combines offline intra-doc masks (blue) with query-adaptive online inter-doc selection (coral). Gray tiles are always kept; white tiles are not computed, including causally masked regions. Schematic selection patterns, not measured sparsity.

1 INTRODUCTION

Retrieval-augmented generation (RAG) grounds a language model's responses in evidence retrieved from an external corpus (Lewis et al., 2020). To improve coverage of relevant evidence, RAG systems typically retrieve multiple passages for each query and incorporate them into the input prompt. However, these passages lengthen the prompt and increase prefill latency, leading to a longer time to first token (TTFT). Since dense attention incurs a computational cost quadratic in prompt length, prefill becomes increasingly expensive as the number or length of retrieved passages grows.

Existing approaches reduce this cost through parallel encoding or sparse attention. Parallel encoding processes context segments independently and combines their representations for the query (Ratner et al., 2023; Yen et al., 2024). For RAG, Block-Attention and TurboRAG precompute document KV

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.

states for reuse across queries (Ma et al., 2025; Lu et al., 2025). This removes inter-document attention during document encoding (Figure 1), and both methods use additional fine-tuning to adapt the model to the changed attention pattern. APE avoids fine-tuning by correcting the attention distribution, but its performance remains sensitive to the attention temperature and scaling factor, which require tuning on a validation set (Yang et al., 2025). Sparse attention instead reduces computation by selecting interactions within the concatenated prompt (Jiang et al., 2024; Lai et al., 2025; Xu et al., 2025). The challenge is to preserve important interactions while keeping selection overhead low and producing block-sparse masks that enable efficient GPU execution.

RAG offers additional structure for reducing sparse attention selection overhead: documents are available in advance, and some attention patterns remain stable when they appear in different prompts. This allows reusable selection decisions to be made offline, while prior knowledge of intra- and inter-document attention can guide online selection. SIFT explores this direction by profiling documents independently and using offline attention statistics to predict important intra- and inter-document interactions (Sanovar et al., 2026). However, our experiments reveal that its offline selection does not reliably capture important inter-document interactions.

In this work, we exploit the different roles of intra-doc and inter-doc attention to coordinate sparse prefill in RAG. Based on our observations, we introduce SUDOKURAG, a training-free framework that couples reusable offline intra-doc masks with adaptive online inter-doc pruning. Specifically, SUDOKURAG first profiles each document independently to cache intra-doc masks. During prefill, it computes prefix and retained intra-doc attention before visiting inter-doc blocks, using the accumulated softmax statistics to guide QK and PV skipping. We integrate these stages into a FlashAttention-3 kernel with shared K/V loads and pipelined execution. We evaluate task accuracy on ChatRAG, LOFT, and LaRA Paper and online prefill-attention efficiency on an NVIDIA H100 GPU. SUDOKURAG achieves speedups of 1.34×, 1.44×, and 1.64× over dense attention at prompt lengths of 32K, 64K, and 128K tokens, respectively, with nearly no accuracy loss at approximately 50% sparsity.

Our contributions are summarized as follows:

- We analyze intra-doc and inter-doc attention in RAG, identifying reusable offline selection decisions and how document structure can guide online sparsification.
- We introduce SUDOKURAG, a sparse prefill framework that uses prefix and intra-doc softmax statistics to guide inter-doc QK and PV skipping within a FlashAttention-3 kernel.
- We evaluate task accuracy on ChatRAG, LOFT, and LaRA Paper and benchmark online attention efficiency across prompt lengths, achieving up to 1.64× speedup over dense attention at approximately 50% sparsity, with nearly no accuracy loss.

2 BACKGROUND

2.1 DOCUMENT-AUGMENTED PROMPTS AND PREFILL COST

RAG augments a user query with retrieved documents (Lewis et al., 2020). We write the resulting *document-augmented prompt* as

$$X = [P, D_1, \ldots, D_M, Q], \qquad N = n_P + \sum_{r=1}^{M} n_r + n_Q. \tag{1}$$

Here, P is a shared instruction prefix of length $n_P = |P|$, D_r is the r-th retrieved document of length $n_r = |D_r|$, and Q denotes the user query, including any final instructions, with length $n_Q = |Q|$. Document lengths may differ, and boundaries are known from prompt construction.

During prefill, dense attention operates over the full document-augmented prompt, with a computational cost quadratic in its total length:

$$C_{\text{attn}}^{\text{prefill}} = \Theta(N^2) = \Theta\left(\left(n_P + \sum_{r=1}^{M} n_r + n_Q\right)^2\right). \tag{2}$$

As more or longer documents are retrieved, the quadratic growth in attention computation can make attention a bottleneck in RAG prefill. Accelerating prefill attention is therefore important for reducing time to first token (TTFT).

2.2 Parallel Encoding

Parallel encoding processes documents independently and combines their representations when processing the query (Ratner et al., 2023; Yen et al., 2024). In the self-attention variant shown in Figure 1a, each document attends to its own causal history and a shared prefix, but not to other documents. The query attends to all retrieved documents, bringing their evidence together only after document encoding.

Before cache reuse, this pattern changes the $\Theta(N^2)$ attention cost in Equation 2 to

$$C^{\text{prefill}}_{\text{attn,parallel}} = \Theta\left(n_P^2 + \sum_{r=1}^{M} n_r(n_P + n_r) + n_Q N\right). \tag{3}$$

The three terms account for encoding the shared prefix once, encoding documents independently, and processing the query, respectively. When document encoding is also independent of the query, document KV states can be precomputed and reused across requests (Ma et al., 2025; Lu et al., 2025; Yang et al., 2025). With these caches, online prefill avoids document encoding, although cache loading and prefix/query processing still incur costs.

However, directly applying this document-isolated mask to a pretrained LLM without adaptation can substantially degrade task accuracy (Ma et al., 2025; Yang et al., 2025). Block-Attention and TurboRAG recover accuracy through fine-tuning, adding training and deployment overhead (Ma et al., 2025; Lu et al., 2025). Training-free corrections such as APE instead rely on validation-set calibration or task-specific hyperparameter choices (Yang et al., 2025; Zhang et al., 2025b). These adaptations retain document isolation, motivating selective joint attention that preserves useful inter-doc interactions during prefill.

2.3 General Sparse Attention

Another line of work accelerates prefill for general long-context inputs by exploiting attention sparsity. Rather than encoding documents independently, these methods skip low-importance interactions within the full sequence. This reduces attention computation beyond the memory-traffic optimizations of exact kernels such as FlashAttention (Dao, 2023; Shah et al., 2024).

Training-free methods apply sparse masks to pretrained models at inference time (Xu et al., 2025; Zhang et al., 2025a; Lin et al., 2025). MInference and FlexPrefill construct input-dependent sparse masks by exploiting head-specific attention patterns (Jiang et al., 2024; Lai et al., 2025). BLASST and CoSA use online-softmax statistics to guide in-kernel pruning (Yuan et al., 2025; Xue et al., 2026).

Although these methods accelerate prefill, they target general long-context inputs and do not explicitly exploit RAG's prefix–document–query structure. This leaves document-aware selection and cross-request reuse of document-specific masks unexploited, motivating the analysis in Section 3.

3 Intra-doc and Inter-doc Sparsity

In this section, we study attention within and across documents during RAG prefill, deriving three key insights (I1), (I2), and (I3) on which interactions to preserve and when to select them.

3.1 Inter-Document Attention Matters

Parallel encoding (Section 2.2) retains dense attention within each document but removes attention between documents during document encoding. Its accuracy degradation without adaptation motivates examining what is lost through document isolation.

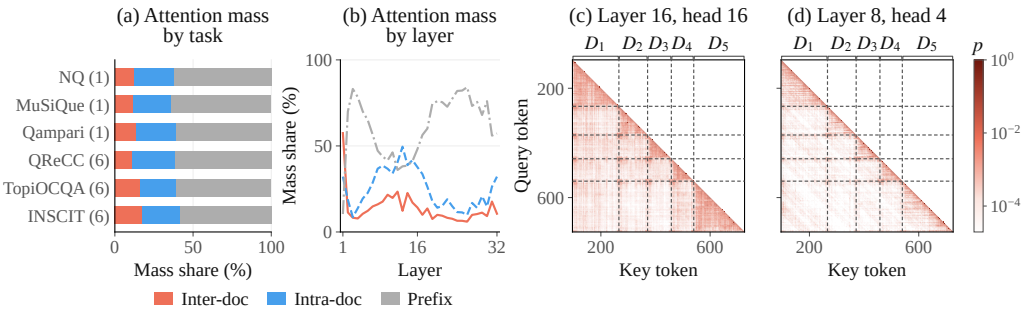


Figure 2: **Attention mass and patterns in RAG prefill.** (a) Stacked task averages and (b) layer-wise six-task averages of inter-doc (red), intra-doc (blue), and prefix (gray) attention as shares of total causal attention mass. Parentheses give context counts. (c,d) Attention maps over five LOFT NQ documents, showing intra-doc (diagonal) and inter-doc (off-diagonal) attention. Colors show probability p; dashed lines mark document boundaries.

Using D_r also for its token-position interval, we distinguish two regions for a query position $i \in D_r$: *intra-document* attention covers $D_r \cap \{j : j \leq i\}$, while *inter-document* attention covers $\bigcup_{t<r} D_t$. Both regions and the prefix keys share one softmax normalization.

We measure inter-doc attention mass during dense prefill with Llama-3.1-8B-Instruct on 21 distinct document contexts across six RAG tasks. For sampled document-token query positions, we compute native post-RoPE attention over all causal keys and measure the mass assigned to earlier documents, the current document, and the prefix. We average within each context, then within each task, and finally equally across the six tasks.

Figure 2(a,b) shows that inter-doc attention receives 13.81% of total mass on average (10.97%–17.35% across tasks), while prefix and intra-doc attention together receive 86.19%. The token-level maps in Figure 2(c,d) further show inter-doc interactions outside the diagonal document blocks—the interactions removed by document isolation. Together with the accuracy degradation under document isolation, these observations motivate our first insight: (I1) **Inter-doc attention matters, but prefix and intra-doc attention account for most of the attention mass on average.**

3.2 Sparsity in Inter-Document Attention

SIFT similarly identifies the loss of inter-document interactions as a limitation of isolated KV reuse and advocates selective recomputation (Sanovar et al., 2026). These interactions also exhibit sparsity: in our six-task measurements, entries with probability at least 0.01 occupy only 0.014% of the causal inter-doc region.

Offline key-frequency selection has low recall. SIFT relies on *cross-attention consistency*: keys that frequently receive high attention within a standalone document are expected to receive high attention from later documents. It therefore uses offline attention statistics to select KV pages for inter-doc recomputation (Sanovar et al., 2026).

We evaluate this rule by profiling each source document independently and checking which high-score inter-doc entries its selected keys cover during joint dense prefill. Using the same RAG inputs as in Section 3.1, we find that the standalone frequent-key selector recalls only 32.54% of online inter-doc entries with probability at least 0.01, at 10.61% retained density. In other words, it misses roughly two-thirds of the high-score inter-doc connections.

Why inter-doc selection should be online. Inter-doc attention changes with the other documents in the prompt and their order. Offline profiling processes each document in isolation, without this context. A mask fixed at that stage cannot adjust to changes in the prompt and may miss important inter-doc connections. At the same token

Table 1: **Matched-budget masks (%).**

Mask	Recall
SIFT-style keys	32.54
Leading keys	23.39
Middle keys	10.53
Last keys	20.80
Random (expected)	17.36
Oracle (online)	85.08

budget, static masks remain low-recall, whereas an online oracle that reselects keys per query using true attention reaches 85.08% (Table 1). Hence, **(I2) Inter-doc sparsity depends on the prompt context, so inter-doc masks should be selected online rather than fixed offline.**

3.3 Sparsity in Intra-Document Attention

For the remaining intra-doc work, SIFT observes that high-score attention locations remain stable across prepended contexts (Sanovar et al., 2026), suggesting offline reuse. Figure 3(a) illustrates the reuse of one intra-doc mask across different preceding contexts.

Using the same setup as Section 3.1, we profile each document preceded by a single BOS token and reuse its thresholded entry mask at matching document-relative positions in joint prompts. Figure 3(b) varies the offline threshold α, with online positives fixed at probability ≥ 0.001. At $\alpha = 10^{-3}$, these masks achieve 91.69% recall at 28.86% entry density. This tests entry reuse, not executable-block sparsity.

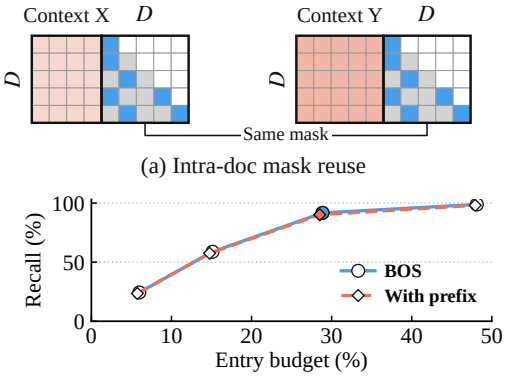


Figure 3: **Intra-doc mask reuse.** (a) Different preceding contexts, the same offline mask (schematic). (b) Recall versus retained entry density at four thresholds; filled markers denote $\alpha = 10^{-3}$.

We further investigate whether attention sinks (Xiao et al., 2024) during offline document profiling affect mask reuse. Figure 3(b) compares BOS-only profiling with a shared-prefix variant that reduces document-head bias. The two recall–budget curves are similar, with no aggregate recall benefit from the added prefix at the tested thresholds. We therefore use BOS-only profiling to generate offline masks. (I3) **Intra-doc masks can be profiled offline and reused across prompts.**

4 SudokuRAG

4.1 Overview

The three insights in Section 3 motivate SudokuRAG's structure-aware design. Following **(I3)**, we profile intra-doc masks once per document and reuse them across prompts. Inter-doc interactions must remain available **(I1)**, but their selection must adapt to the current prompt **(I2)**. This motivates **offline intra-doc sparsity** and **online inter-doc sparsity**.

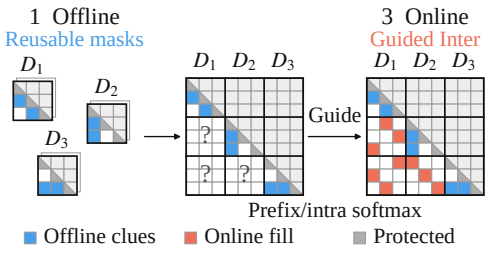


Figure 4: **Offline clues guide online selection.** (1) Maintain per-document intra-doc masks offline. (2) Assemble them into the attention map, leaving inter-doc choices unresolved (?). (3) Use current prefix/intra softmax statistics to guide online completion. Blue: offline clues; coral: online choices. Document-only schematic.

Our key insight is to *coordinate* these two forms of sparsity through RAG structure, rather than merely combine independently selected masks. Offline (Sec. 4.2), we profile each document independently and store reusable intra-doc masks as Sudoku-like clues (Figure 4). Online (Sec. 4.3), we first compute attention to the shared prefix and the intra-doc tiles retained by the offline mask. Their online-softmax state determines the pruning threshold for Q/K-mean inter-doc scores, rather than applying a fixed cutoff independent of the current prompt. **RAG structure thus turns intra-doc computation into online guidance for inter-doc sparsity.**

Algorithm 1 SUDOKURAG prefill

Input: Query/key/value matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V}$; mapped intra-doc mask $M^{\text{Intra-doc}}$; document boundaries; τ, $\lambda_{\text{QK}}, \lambda_{\text{PV}} \in (0, 1)$

Output: Sparse attention output $\widetilde{\mathbf{O}}$

Divide $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ into $B = 64$-token blocks; compute Q/K means and reliability using τ.
for each query block a in parallel **do**
  Load $\mathbf{Q}_a$; initialize $(m, \ell, \mathbf{A}) \leftarrow (-\infty, 0, 0)$
  Form prefix $\mathcal{J}_P$, retained intra-doc $\mathcal{J}_{\text{Intra-doc}}$, and earlier-document $\mathcal{J}_{\text{Inter-doc}}$ block sets using $M^{\text{Intra-doc}}$ and the protection rules
  for $\phi \in (P, \text{Intra-doc}, \text{Inter-doc})$ in order **do**
    if $\phi =$ Inter-doc **then**
      $\theta_a \leftarrow \min_i(m_i + \log \ell_i) + \log(\lambda_{\text{QK}}/B)$; prune eligible $b \in \mathcal{J}_\phi$ with $\widehat{s}_{ab} < \theta_a$
    end if
    for $b \in \mathcal{J}_\phi$ in increasing page order **do**
      Load $\mathbf{K}_b$
      $\mathbf{S} \leftarrow \mathbf{Q}_a \mathbf{K}_b^\top / \sqrt{d}$, with causal/padding masking
      if $\phi =$ Inter-doc and $\max(\text{rowmax}(\mathbf{S}) - m) < \log \lambda_{\text{PV}}$ **then**
        continue
      end if
      Load $\mathbf{V}_b$
      $m' \leftarrow \max(m, \text{rowmax}(\mathbf{S}))$; $\beta \leftarrow \exp(m - m')$
      $\mathbf{P} \leftarrow \exp(\mathbf{S} - m')$; $\ell \leftarrow \beta\ell + \text{rowsum}(\mathbf{P})$
      $\mathbf{A} \leftarrow \beta\mathbf{A} + \mathbf{P}\mathbf{V}_b$; $m \leftarrow m'$
    end for
  end for
  Write $\widetilde{\mathbf{O}}_a \leftarrow \mathbf{A}/\ell$
end for
return $\widetilde{\mathbf{O}}$

To co-design the algorithm with hardware execution, we adopt *block-sparse attention* (Guo et al., 2024) rather than an irregular entry-level mask. Both intra-doc and inter-doc sparsity use a common block size of $B = 64$ tokens along the query and key axes. Finally, we implement SUDOKURAG on top of FlashAttention (Shah et al., 2024) and optimize the kernel with command compaction, shared K/V loads, and pipelined execution (Sec. 4.4).

4.2 INTRA-DOC SPARSITY: OFFLINE DOCUMENT MASKS

Following SIFT (Sanovar et al., 2026), we profile each document D independently with a dense forward pass on `BOS + document`. Let $\pi_{ij}^{D,\ell,h}$ denote the causal attention probability between document tokens i, j at layer ℓ and query head h, with BOS included in normalization. We retain a document-relative block (I_a, I_b) if any causal entry meets the threshold α:

$$A_{ab}^{D,\ell,h} = \mathbf{1}\left[\max_{\substack{i \in I_a,\, j \in I_b \\ j \le i}} \pi_{ij}^{D,\ell,h} \ge \alpha\right]. \tag{4}$$

We cache the document-local masks as bit-masks for reuse across prompts, which incurs minimal additional storage overhead. At inference time, we align each mask with the document's position in the global attention matrix, yielding $M^{\text{Intra-doc}}$, and retain an intra-doc block if it overlaps any selected local block. Diagonal and boundary blocks are always retained.

4.3 INTER-DOC SPARSITY: SOFTMAX-GUIDED SKIPPING

RAG-ordered PV skipping. SUDOKURAG uses prefix and intra-doc computation to guide online inter-doc skipping. We start from the *PV skip* in BLASST and CoSA (Yuan et al., 2025; Xue et al., 2026): after computing a tile's QK scores $\mathbf{S}$, skip softmax and PV if $\max(\text{rowmax}(\mathbf{S}) - m) < \log \lambda_{\text{PV}}$, where m is the running row-wise maximum and $0 < \lambda_{\text{PV}} < 1$. Every valid query row must satisfy this test. FlashAttention's inner-loop order matters here: fixed-mask attention is order-independent in exact arithmetic, but online pruning depends on statistics accumulated so far (Dao,

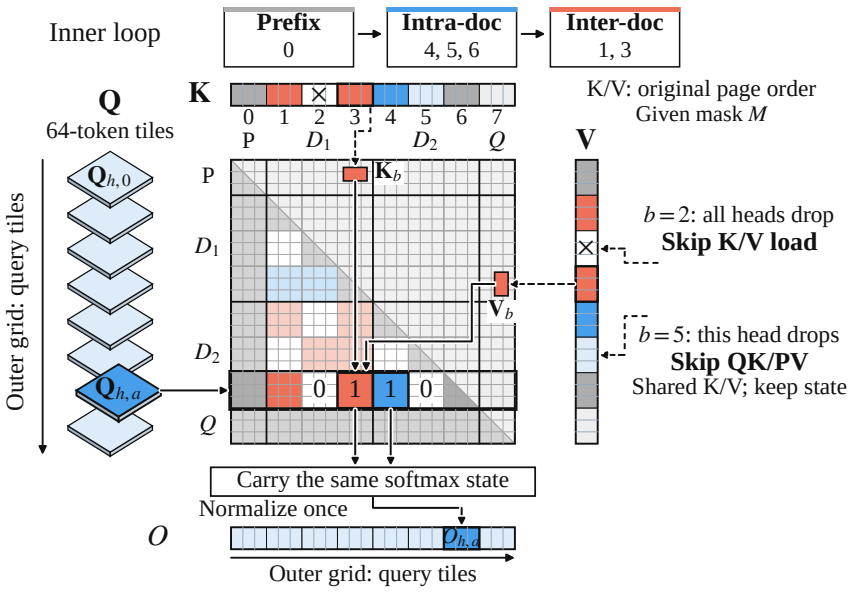


Figure 5: **SUDOKURAG kernel schedule.** The inner loop visits Prefix → Intra-doc → Inter-doc. Current prefix/intra softmax statistics set adaptive pruning thresholds for Q/K-mean inter-doc scores; all phases share the same softmax state. K/V strips show original page order, not visit order. For $a = 6$, page 2 is not fetched; page 5 is shared but skips QK/PV for this head. Normalize only after all phases (Algorithm 1). Each query or KV block contains 64 tokens; the mask and layout are schematic.

2023; Xue et al., 2026). BLASST studies local-first traversal, while CoSA prioritizes proxy-predicted row-max blocks to improve subsequent skipping (Yuan et al., 2025; Xue et al., 2026).

In RAG, **(I1)** provides a structural basis for this ordering: prefix and intra-doc interactions carry most attention mass on average. We therefore visit *Prefix → Intra-doc → Inter-doc*, rather than following prompt order, which reaches earlier-document keys before the current document's intra-doc keys. For each query tile and head, attention to the prefix and offline-retained intra-doc tiles supplies the initial running maxima and normalization statistics for inter-doc skipping. Both stages use this state; surviving inter-doc tiles update it, with one final normalization (Figure 5).

Mean-based QK skipping. PV skipping still computes full QK. We therefore estimate inter-doc scores from block means (Zhang et al., 2025a). Let q_i, k_j, v_j denote the current d-dimensional query, key, and value vectors, and $\bar{q}_a, \bar{k}_b$ the query and key means of blocks a, b. We use

$$\widehat{s}_{ab} = \bar{q}_a^\top \bar{k}_b / \sqrt{d}. \tag{5}$$

QK skipping applies only when both blocks are complete and their within-block average pairwise cosine similarity, including self-pairs, exceeds τ. Eligible key blocks must lie in documents preceding every document touched by the query block. We prune an eligible block when its estimated unnormalized weight $B \exp(\widehat{s}_{ab})$ is below $\lambda_{\mathrm{QK}} e^{m_i} \ell_i$ for every valid query row i, where $\lambda_{\mathrm{QK}} \in (0, 1)$ controls the per-block relative-weight threshold. Algorithm 1 evaluates this heuristic in log space, taking the minimum over valid query rows. Rejected tiles skip both QK and PV; all others, including unreliable blocks, compute QK and undergo the PV test.

Algorithm 1 gives the forward pass for one attention head. Subscripts a, b select query and key/value blocks of $\mathbf{Q}, \mathbf{K}, \mathbf{V}$; $\mathcal{J}_\phi$ lists key-block indices for phase $\phi \in (P, \text{Intra-doc}, \text{Inter-doc})$. The state $(m, \ell, \mathbf{A})$ holds row-wise running maxima, rescaled weight sums, and weighted-value sums. The updated maxima m' and rescaling factors β maintain this state; $\mathbf{P}$ contains unnormalized exponential weights, and $\widetilde{\mathbf{O}}$ is the normalized output. Row reductions act over key positions, with elementwise arithmetic and row-wise broadcasting. The PV test considers only valid query rows.

4.4 KERNEL OPTIMIZATIONS

Figure 5 illustrates the SUDOKURAG kernel schedule. The algorithm fits directly into the tiled execution template of FlashAttention-3 (FA-3) (Shah et al., 2024): prefix, intra-doc, and inter-doc blocks share the running softmax statistics and output accumulator. Prefix/intra statistics guide inter-doc selection inside the attention loop, while retained blocks use the standard QK, softmax, and PV operations. This allows structure-aware selection and attention to be fused without materializing the token-level attention matrix or launching separate attention kernels for the three regions.

At the kernel level, we compact retained pages into commands with per-head keep bits and share K/V loads across four query heads. Each query tile is loaded once; a three-buffer TMA/WGMMA pipeline overlaps K/V transfers with computation. Pages rejected before QK by all four heads are not fetched, while each head independently applies the QK and PV gates. Skipped heads preserve their accumulators and still participate in pipeline synchronization. BF16 operands and FP32 softmax statistics retain the usual mixed-precision computation, with one final normalization across all regions.

5 EXPERIMENTS

In this section, we conduct experiments to compare SUDOKURAG with state-of-the-art (SOTA) sparse prefill methods in terms of accuracy and efficiency on RAG tasks. We present the accuracy and efficiency results in Section 5.1 and Section 5.2, respectively.

5.1 ACCURACY EVALUATION

Benchmarks and Models. We evaluate task accuracy using two widely used models, Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen3-8B(thinking disabled) (Qwen Team, 2025). We use three groups of benchmarks: Doc2Dial, QReCC, and QuAC from ChatRAG Bench (Liu et al., 2024); QUEST, SciFact, ArguAna, TopiOCQA, and NQ for LOFT retrieval, together with NQ, QUEST, and TopiOCQA for LOFT RAG (Lee et al., 2024); and the Paper subset of LaRA (Li et al., 2025). These datasets cover question answering and document retrieval. For LaRA Paper, all methods use `deepseek-v4.1-flash` as the LLM judge, and we report macro accuracy over the four question types.

Baselines. We compare SUDOKURAG with full attention, MInference (Jiang et al., 2024), FlexPrefill (Lai et al., 2025), Block-Attention (without finetune) (Ma et al., 2025), and SIFT (Sanovar et al., 2026) on both models. MInference and FlexPrefill are general-purpose sparse attention methods, whereas Block-Attention and SIFT are RAG-specific sparse attention methods that exploit document structure. For MInference, we profile and calibrate each model using the official default configuration. For FlexPrefill, we use the official Llama-3.1-8B settings ($\gamma = 0.95$, $\tau = 0.1$) for Llama-3.1-8B-Instruct, and adopt the Qwen2-7B settings ($\gamma = 0.9$, $\tau = 0.1$) for Qwen3-8B.

Implementation Details. For SUDOKURAG, we use query/key blocks of $B = 64$ tokens, an intra-doc threshold $\alpha = 10^{-3}$, a cosine-similarity threshold $\tau = 0.6$, and a QK-skipping parameter $\lambda_{\mathrm{QK}} = 0.8$. These settings are fixed across both models and all benchmarks in the main accuracy comparison. The first two layers and all decoding steps use dense attention. This configuration yields approximately 50% attention sparsity.

Results. Table 2 shows that SUDOKURAG closely matches full attention on ChatRAG, with average F1 differences below 0.04 percentage points for both models. MInference and FlexPrefill also maintain ChatRAG performance, but exhibit larger average performance losses on LOFT retrieval. On this benchmark, SUDOKURAG achieves the highest average score among the evaluated sparse methods, exceeding the strongest sparse baseline by 4.60 and 1.80 percentage points on Llama-3.1-8B-Instruct and Qwen3-8B, respectively. On LOFT RAG, SUDOKURAG also achieves the highest average subspan EM among sparse methods on both models, exceeding the strongest sparse baseline by 2.00 and 0.67 percentage points, respectively. Relative to full attention, its average score is 1.67 percentage points higher on Llama and 1.67 percentage points lower on Qwen. On LaRA Paper, SUDOKURAG likewise achieves the highest average accuracy among sparse methods on both

Table 2: **Average task scores on RAG benchmarks (%).** We report unweighted means over the three ChatRAG datasets, five LOFT retrieval datasets, three LOFT RAG datasets, and two LaRA Paper context lengths. Bold marks the best score excluding full attention for each model and column. Detailed results are in Appendix A.

	Llama-3.1-8B-Instruct				*Qwen3-8B (thinking disabled)*			
Method	ChatRAG	LOFT Ret.	LOFT RAG	LaRA	ChatRAG	LOFT Ret.	LOFT RAG	LaRA
Full attention	31.7344	37.0000	29.3333	81.8700	30.0690	28.2000	30.3333	82.6450
MInference	31.7664	29.0000	18.0000	80.2500	30.0134	26.2000	27.6667	80.8400
FlexPrefill	31.6867	30.2000	29.0000	79.0500	**30.2595**	21.4000	28.0000	78.6150
Block-Attention (without finetune)	28.0511	0.0000	0.0000	42.4650	23.8631	0.0000	0.0000	54.9000
SIFT	31.3105	14.4000	24.0000	58.6900	29.8269	26.8000	28.0000	73.4300
SUDOKURAG	**31.7706**	**34.8000**	**31.0000**	**81.7200**	30.0312	**28.6000**	**28.6667**	**81.2200**

models, outperforming SIFT by 7.25–24.71 percentage points across the four model–context settings. Block-Attention without finetuning obtains zero scores on all evaluated LOFT tasks for both models.

5.2 EFFICIENCY EVALUATION

We benchmark the online prefill-attention path of SUDOKURAG on an NVIDIA H100 80GB HBM3 GPU using Llama-3.1-8B-Instruct (Grattafiori et al., 2024) with GQA, BF16, and batch size 1. We use three real LOFT NQ development prompts (Lee et al., 2024) sharing a document corpus, with full prefixes and question/few-shot suffixes preserved. The 32K, 64K, and 128K inputs contain exactly 32,768, 65,536, and 131,072 tokens; longer inputs use additional nonduplicated NQ documents, clipping only the final document as needed. The dense baseline is cuDNN SDPA, the fastest measured backend at each length among cuDNN SDPA, FlashAttention-2, and PyTorch Flash SDPA. Table 3 sweeps target sparsity from 0% to 95%, measured as the head-averaged fraction of causal 64×64 tiles skipped.

Table 3: **Online prefill-attention speedup (×).** H100, Llama-3.1-8B-Instruct; dense baseline: cuDNN SDPA.

Sparsity (%)	**32K**	**64K**	**128K**
0	0.75	0.79	0.90
25	0.95	1.01	1.13
50	1.34	1.44	1.64
75	2.28	2.53	2.95
90	4.23	4.85	5.81
95	**6.12**	**8.17**	**9.41**

Key Results. At 95% sparsity, SUDOKURAG achieves online prefill-attention speedups of 6.12×, 8.17×, and 9.41× over dense attention at 32K, 64K, and 128K tokens, respectively. The largest speedup in the displayed 0–95% sweep is 9.41× at 128K. Speedups increase with sparsity and prompt length: the 0% control is slower than dense attention (0.75–0.90×), while all three lengths benefit at 50% sparsity (1.34–1.64×).

6 CONCLUSION

We present SUDOKURAG, a training-free framework that coordinates reusable offline intra-doc masks with online inter-doc pruning guided by prefix and intra-doc softmax statistics. It closely matches full attention on ChatRAG and achieves the highest average scores among the evaluated sparse methods on LOFT and LaRA Paper. On H100, SUDOKURAG delivers 1.34–1.64× online prefill-attention speedups over dense attention at approximately 50% sparsity across 32K–128K-token prompts.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods and interpret experimental results. Additionally, we used generative AI tools for brainstorming, searching for and identifying relevant

literature, summarizing and analyzing existing literature, creating or editing software code, and drafting parts of this research paper.

We have reviewed all AI-assisted work. In particular, we tested AI-assisted code for correctness. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics Statement

We consider SudokuRAG's potential impacts in light of the ICLR Code of Ethics. Sparse attention can omit answer-critical evidence and does not address existing model or data biases. We report evaluation limitations and both favorable and unfavorable results. Model and dataset use and redistribution should respect licenses, attribution requirements, privacy, and confidentiality. Deployment requires application-specific evaluation of reliability and potential harms, particularly where incorrect answers could have serious consequences.

References

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning, 2023. URL https://arxiv.org/abs/2307.08691.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Junxian Guo, Haotian Tang, Shang Yang, Zhekai Zhang, Zhijian Liu, and Song Han. Block Sparse Attention. https://github.com/mit-han-lab/Block-Sparse-Attention, 2024.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention, 2024. URL https://arxiv.org/abs/2407.02490.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. FlexPrefill: A context-aware sparse attention mechanism for efficient long-sequence inference, 2025. URL https://arxiv.org/abs/2502.20766.

Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien M. R. Arnold, Vincent Perot, Siddharth Dalmia, Hexiang Hu, Xudong Lin, Panupong Pasupat, Aida Amini, Jeremy R. Cole, Sebastian Riedel, Iftekhar Naim, Ming-Wei Chang, and Kelvin Guu. Can long-context language models subsume retrieval, RAG, SQL, and more?, 2024. URL https://arxiv.org/abs/2406.13121.

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, 2020. URL https://arxiv.org/abs/2005.11401.

Kuan Li, Liwen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Shuai Wang, and Minhao Cheng. LaRA: Benchmarking retrieval-augmented generation and long-context LLMs – no silver bullet for LC or RAG routing, 2025. URL https://arxiv.org/abs/2502.09977.

Chaofan Lin, Jiaming Tang, Shuo Yang, Hanshuo Wang, Tian Tang, Boyu Tian, Ion Stoica, Song Han, and Mingyu Gao. Twilight: Adaptive attention sparsity with hierarchical top-p pruning, 2025. URL https://arxiv.org/abs/2502.02770.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. ChatQA: Surpassing GPT-4 on conversational QA and RAG, 2024. URL https://arxiv.org/abs/2401.10225.

Songshuo Lu, Hua Wang, Yutian Rong, Zhi Chen, and Yaohua Tang. TurboRAG: Accelerating retrieval-augmented generation with precomputed KV caches for chunked text. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6588–6601. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.334. URL https://aclanthology.org/2025.emnlp-main.334/.

Dongyang Ma, Yan Wang, and Tian Lan. Block-attention for efficient prefilling. In *International Conference on Learning Representations*, volume 2025, pp. 63774–63788, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a03037317560b8c5f2fb4b6466d4c439-Abstract-Conference.html.

Qwen Team. Qwen3-8B, 2025. URL https://huggingface.co/Qwen/Qwen3-8B.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6383–6402. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.352. URL https://aclanthology.org/2023.acl-long.352/.

Rya Sanovar, Srikant Bharadwaj, Hritvik Taneja, and Moinuddin Qureshi. SIFT: Selective-index for fast compute of RAG prefill by exploiting attention invariance, 2026. URL https://arxiv.org/abs/2606.09441.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision, 2024. URL https://arxiv.org/abs/2407.08608.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.17453.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. XAttention: Block sparse attention with antidiagonal scoring, 2025. URL https://arxiv.org/abs/2503.16428.

Yufei Xue, Lin Niu, Hong Liu, Siran Liu, Hanyong Shao, Wei Liu, Guanghua Yu, Jianchen Zhu, and Jun Zhang. CoSA: Accelerating long-context inference via proxy-kernel co-designed sparse attention, 2026. URL https://arxiv.org/abs/2607.25291.

Xinyu Yang, Tianqi Chen, and Beidi Chen. APE: Faster and longer context-augmented generation via adaptive parallel encoding, 2025. URL https://arxiv.org/abs/2502.05431.

Howard Yen, Tianyu Gao, and Danqi Chen. Long-context language modeling with parallel context encoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2588–2610. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.142. URL https://aclanthology.org/2024.acl-long.142/.

Jiayi Yuan, Cameron Shinn, Kai Xu, Jingze Cui, George Klimiashvili, Guangxuan Xiao, Perkz Zheng, Bo Li, Yuxin Zhou, Zhouhai Ye, Weijie You, Tian Zheng, Dominic Brown, Pengbo Wang, Markus Hoehnerbach, Richard Cai, Julien Demouth, John D. Owens, Xia Hu, Song Han, Timmy Liu, and Huizi Mao. BLASST: Dynamic BLocked attention sparsity via softmax thresholding, 2025. URL https://arxiv.org/abs/2512.12087.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. SpargeAttention: Accurate and training-free sparse attention accelerating any model inference, 2025a.

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 (Volume 1: Long Papers)*, pp. 9840–9855. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.acl-long.485. URL https://aclanthology.org/2025.acl-long.485/.

A Detailed Accuracy Results

Tables 4 and 5 provide the per-dataset ChatRAG and LOFT scores and per-context LaRA Paper scores underlying the averages in Table 2. Block-Attention obtains zero scores on all LOFT tasks for both models; its all-zero rows are omitted from Table 5.

Table 4: **Accuracy on ChatRAG and LaRA Paper (%).** Avg is the unweighted mean across the three ChatRAG datasets or the two LaRA context lengths. Bold marks the best score excluding full attention for each model and column.

	ChatRAG Bench				LaRA Paper		
Method	Doc2Dial	QReCC	QuAC	Avg	32K	128K	Avg
Metric	F1				Accuracy		
Llama-3.1-8B-Instruct							
Full attention	29.2967	34.6594	31.2471	31.7344	85.48	78.26	81.8700
MInference	**29.3204**	34.6636	31.3153	31.7664	**86.42**	74.08	80.2500
FlexPrefill	29.1464	34.5486	**31.3651**	31.6867	82.47	75.63	79.0500
Block-Attention (without finetune)	26.3244	32.2664	25.5626	28.0511	57.15	27.78	42.4650
SIFT	28.9337	34.2418	30.7561	31.3105	62.81	54.57	58.6900
SudokuRAG	29.2706	**34.7271**	31.3142	**31.7706**	84.16	**79.28**	**81.7200**
Qwen3-8B (thinking disabled)							
Full attention	25.5230	37.5130	27.1709	30.0690	88.88	76.41	82.6450
MInference	25.5692	37.2875	27.1835	30.0134	**87.09**	74.59	80.8400
FlexPrefill	**25.7637**	**37.6396**	**27.3752**	**30.2595**	82.00	75.23	78.6150
Block-Attention (without finetune)	20.7611	31.4338	19.3945	23.8631	63.38	46.42	54.9000
SIFT	25.6305	36.9196	26.9307	29.8269	79.37	67.49	73.4300
SudokuRAG	25.5893	37.3217	27.1825	30.0312	86.62	**75.82**	**81.2200**

Table 5: **Accuracy on LOFT retrieval and RAG (%).** Avg is the unweighted mean within each group. Bold marks the best available score excluding full attention for each model and column.

	LOFT retrieval						LOFT RAG			
Method	QUEST	SciFact	ArguAna	TopiOCQA	NQ	Avg	NQ	QUEST	TopiOCQA	Avg
Metric	MRecall@3		Recall@1				subspan_em			
Llama-3.1-8B-Instruct										
Full attention	5.0000	58.0000	15.0000	15.0000	92.0000	37.0000	54.0000	11.0000	23.0000	29.3333
MInference	4.0000	29.0000	16.0000	**14.0000**	82.0000	29.0000	32.0000	4.0000	18.0000	18.0000
FlexPrefill	2.0000	40.0000	**21.0000**	4.0000	84.0000	30.2000	52.0000	6.0000	**29.0000**	29.0000
SIFT	1.0000	4.0000	14.0000	1.0000	52.0000	14.4000	51.0000	1.0000	20.0000	24.0000
SudokuRAG	**5.0000**	**55.0000**	13.0000	10.0000	**91.0000**	**34.8000**	**54.0000**	**11.0000**	28.0000	**31.0000**
Qwen3-8B (thinking disabled)										
Full attention	1.0000	43.0000	15.0000	15.0000	67.0000	28.2000	54.0000	3.0000	34.0000	30.3333
MInference	2.0000	41.0000	12.0000	12.0000	64.0000	26.2000	50.0000	4.0000	29.0000	27.6667
FlexPrefill	**3.0000**	36.0000	**14.0000**	10.0000	44.0000	21.4000	50.0000	2.0000	**32.0000**	28.0000
SIFT	2.0000	34.0000	11.0000	**19.0000**	68.0000	26.8000	50.0000	**5.0000**	29.0000	28.0000
SudokuRAG	2.0000	**49.0000**	11.0000	12.0000	**69.0000**	**28.6000**	**52.0000**	2.0000	**32.0000**	**28.6667**