

Beyond Chunking: Efficient Global Pooling for Holistic Long-Document Representation

Hoin Jung, Xiaoqian Wang

Elmore Family School of Electrical and Computer Engineering

Purdue University

West Lafayette, IN 47907

{jung414, joywang}@purdue.edu

Abstract

Effectively representing long documents is a persistent challenge in natural language processing, as foundational encoders are constrained by limited context windows. Prevailing methods like truncation and independent chunk aggregation can discard content or lose cross-chunk context, hindering downstream performance. To address this problem, we introduce **Spectral Attention Token Pooling (SATPool)**, a novel, encoder-agnostic module that generates a single, fixed-size vector from variable-length sequences of locally encoded tokens. SATPool operates in two stages: it first uses efficient linear attention to model global token interactions after chunk encoding, then employs **Spectral Token Compression (STC)** to compress these globally-aware token representations into a compact, context-aware vector. Across long-document classification, Retrieval-Augmented Generation (RAG), and multimodal RAG, SATPool achieves strong overall improvements and robust top-tier performance across datasets and encoder backbones. The pretrained encoder remains frozen, while the lightweight SATPool module and task-specific heads are trained for each downstream task.¹

1 Introduction

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has become a standard paradigm for enhancing the capabilities of Large Language Models (LLMs) (Li et al., 2024a) and, more recently, Large Multimodal Models (LMMs) by grounding them in external knowledge (Marino et al., 2019). These systems leverage vast repositories of information, from web pages to specialized legal (Wiratunga et al., 2024), medical (S et al., 2024), or technical documents (Khemakhem et al., 2024), to generate more accurate and contextually relevant responses. However, a significant and persistent challenge

¹<https://github.com/HoinJung/SATPool>

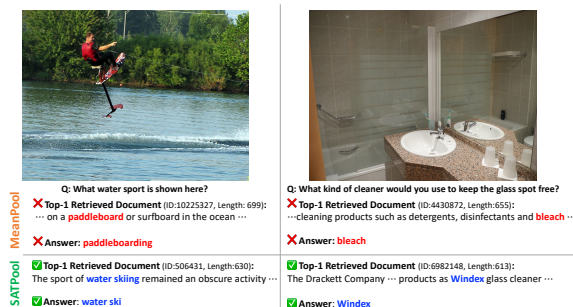


Figure 1: **Qualitative comparison for Multimodal RAG.** SATPool generates correct answers by retrieving semantically relevant documents that MeanPool misses. (Left) MeanPool retrieves an irrelevant document about paddleboarding based on broad associations with the water-sport category. (Right) MeanPool relies on a superficial match to “cleaner” but misses the specific “glass” context. These examples illustrate how broader document context can improve retrieval; they do not imply that document-level retrieval always supersedes passage-level retrieval.

arises when these source documents are longer than the input capacity of the text encoders used for retrieval. Foundational encoders like BERT (Devlin et al., 2019) and CLIP (Radford et al., 2021), which are widely used for their robust representations, are limited to relatively short contexts (e.g., 512 and 77 tokens, respectively), forcing a compromise when processing long-form text. Table 1 demonstrates that long documents are prevalent across datasets and tasks.

The predominant strategy to handle long documents is chunking, where a document is segmented into smaller, manageable pieces and a final representation is created from the chunks. Some methods compute one document vector by averaging chunk embeddings. RAG pipelines may instead retrieve each independently encoded chunk as a passage (Guu et al., 2020), or determine document relevance by the maximum similarity over its chunks (Dai and Callan, 2019). These strategies

Task	Dataset	# Documents	Avg. Length	Max. Length	% of Long Doc.
Text Classification	20NewsGroups	18,846	325.93	138,055	8.78
	EURLEX	60,000	1,509.01	248,252	60.18
RAG	NQ	39,415	3,766.93	68,842	89.56
	HotPotQA	88,287	2,202.71	49,503	72.22
Multimodal RAG	OKVQA	114,809	145.50	926	0.13
	EVQA	50,205	248.56	26,726	10.49

Table 1: Dataset analysis shows many documents exceed the text encoder’s capacity (larger than 512 tokens), highlighting the need for global context pooling in long documents.

can miss information distributed across chunks or context needed to interpret an isolated passage, as detailed in Section 3 and Fig. 2. However, chunk-in-isolation, chunk-in-context, and document-level representations remain distinct: SATPool targets compact document-level representation and is complementary to context-aware passage retrieval.

To address this challenge, we introduce **Spectral Attention Token Pooling (SATPool)**, a novel pooling method that creates a single, fixed-size vector from an assembled sequence of locally encoded document tokens. It functions as an encoder-agnostic, plug-and-play module that enhances performance without requiring a specialized long-context model or modification of the pretrained encoder. SATPool operates in two stages: first, it employs efficient linear attention to model interactions across the assembled token sequence. Second, it uses Spectral Token Compression (STC) to distill the attention-reweighted embeddings into a compact representation that preserves semantic content and coarse positional dynamics. Here, “plug-and-play” means that the base encoder remains frozen; SATPool and the downstream task head still require task-specific training. We validate SATPool through extensive experiments across classification, RAG, and multimodal RAG.

Crucially, our work adopts a **representation-centric** perspective. While we validate our method on downstream tasks like RAG and text classification, our primary objective is not to optimize the generative components (e.g., the LLM in RAG), but to construct a **dense, holistic representation** for the document itself.

This perspective is particularly appropriate when the prediction target or indexing unit is the complete document, as in document classification and document-level retrieval. It differs from chunk-in-context approaches, which retain passage-level retrieval units while incorporating surrounding context into each chunk representation. Such ap-

proaches may be preferable when a generator needs a small, localized evidence span. SATPool instead prioritizes a compact global vector, which can reduce index size and represent evidence distributed across several chunks. We therefore view the two approaches as complementary design choices rather than mutually exclusive alternatives.

Our primary contributions are as follows:

1. We introduce **SATPool**, a novel, encoder-agnostic pooling method that models cross-chunk token interactions before producing a single, context-aware document representation. Its two-stage architecture captures global semantics and coarse positional dynamics.
2. We demonstrate through extensive experiments that SATPool achieves strong overall improvements across long-document classification, RAG, and multimodal RAG. Component and length-stratified analyses clarify the respective contributions of linear attention and STC and identify the settings in which the gains are largest.

Ultimately, SATPool provides a practical and effective solution to fully leverage pre-trained encoders on long-form text, improving performance and fidelity across a range of applications.

2 Related Work

2.1 Text Encoders for Long Context

The challenge of processing long sequences has spurred the development of specialized transformer architectures. Models like Longformer (Beltagy et al., 2020) and LongT5 (Guo et al., 2022) extend the context window to several thousand tokens (e.g., 4096 tokens) using efficient attention patterns. Benchmarks have consistently emphasized the need for such capabilities (Park et al., 2022). However, these models still operate under a fixed token limit. Longtriever (Yang et al.,

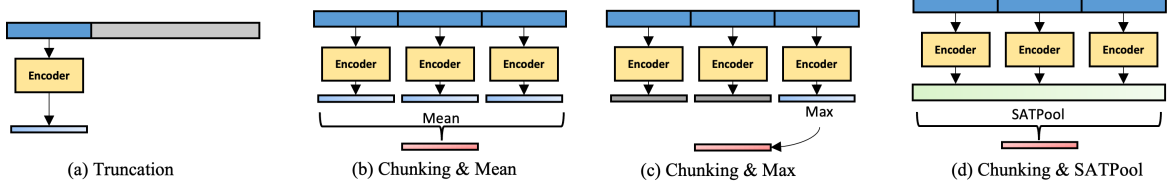


Figure 2: **(a)** Text encoders are constrained by limited input size, necessitating truncation or chunking. Common approaches **(b)** average representations across chunks or **(c)** select the maximum-scoring chunk in RAG settings. Such aggregation can discard inter-chunk dependencies. In contrast, **(d)** SATPool models interactions across the assembled token sequence before Spectral Token Compression. Its aggregation cost grows linearly with the number of assembled tokens.

2023) is specifically pretrained for long-context retrieval but is tied to its architecture. Other approaches, such as LTR-BERT (Wang et al., 2024a) and CFIR (Long et al., 2024), require retraining the text encoder, which is computationally expensive and task-specific. Late Chunking (Günther et al., 2024) produces context-aware chunks but assumes an encoder capable of processing the document jointly. SATPool is encoder-agnostic and has no fixed aggregation-stage context window, although its runtime and memory grow linearly with the assembled sequence length.

2.2 Handling Long Context in RAG

Within RAG research, handling long documents has been a persistent theme. Foundational RAG models for NLP (Guu et al., 2020; Lewis et al., 2020; Liu et al., 2024) and multimodal tasks (Zhang et al., 2025) commonly treat independently encoded chunks as passages for similarity search. This approach can miss relevant information when the context needed to interpret a passage lies in another chunk. More advanced methods create longer retrieval units while retaining internal chunk-level scoring (Jiang et al., 2024), or reconstruct context from initially retrieved chunks (Zhao et al., 2024). SATPool instead creates a single vector per document. This is useful for document-level indexing or distributed evidence, but it is not intended to establish that document retrieval generally replaces fine-grained passage or chunk-in-context retrieval.

While modern LLMs can handle longer input sequences, they aren’t a complete solution to the problem. Research has shown that their performance drops significantly when relevant information is buried in the middle of a long context, known as the “lost in the middle” problem (Liu et al., 2023; Jin et al., 2024). This issue stems from a different aspect of long-context challenges, specifically related to generation tasks rather than creating rep-

resentations.

3 Problem Definition

Given a document D represented as a long sequence of tokens with length L , and a pre-trained transformer-based encoder $\mathcal{E}$ with a maximum input capacity of L_{limit} tokens, a fundamental challenge arises when $L \gg L_{\text{limit}}$.

Prevailing approaches resort to suboptimal strategies like truncation or chunking. With truncation, the document is simply cut off at the context limit, $D' = D_{1:L_{\text{limit}}}$, and its representation becomes $\mathbf{z}_{\text{trunc}} = \mathcal{E}(D')$. This approach leads to a catastrophic loss of any information beyond the initial segment. A more common strategy is chunking, where the document D is partitioned into a set of N smaller chunks, $D = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N\}$, such that $|\mathbf{c}_i| \leq L_{\text{limit}}$. The encoder processes each chunk independently, and the resulting representations are aggregated. For instance, in text classification, a document vector is often computed by averaging chunk representations, $\mathbf{z}_{\text{mean}} = \frac{1}{N} \sum_{i=1}^N \mathcal{E}(\mathbf{c}_i)$. Similarly, in RAG and Multimodal RAG, a document’s relevance to a query $\mathbf{q}$ is typically determined by the maximum similarity score over its constituent chunks, $\text{score}(D, \mathbf{q}) = \max_{i=1, \dots, N} \text{sim}(\mathcal{E}(\mathbf{c}_i), \mathcal{E}(\mathbf{q}))$. However, both truncation and chunking-based aggregation are fundamentally flawed. Truncation discards vast portions of the text, while chunking severs long-range dependencies and discards the overarching narrative, resulting in a fragmented representation of the document’s global context, as illustrated in Fig. 2.

Our objective is to design a pooling function, f_{pool} , that acts as a lightweight, plug-and-play module operating on top of a frozen encoder. This function maps the full sequence of token embeddings $\mathbf{H} \in \mathbb{R}^{L \times d}$ to a single, fixed-size vector $\mathbf{z} \in \mathbb{R}^d$, a **dense document representation**. Unlike standard permutation-invariant pooling, $\mathbf{z}$ should retain both

semantic content and coarse positional variation across the document.

4 Proposed Method

To address the challenges of processing long documents, we propose **Spectral Attention Token Pooling (SATPool)**, a two-stage method that transforms a variable-length sequence of locally encoded tokens into a fixed-size, globally contextualized document representation. SATPool operates on the hidden states of a standard frozen pretrained encoder and can therefore be integrated into existing classification and retrieval pipelines without modifying the encoder architecture. Fig. 4 provides an overview of the complete SATPool pipeline, from local chunk encoding to global interaction and spectral compression.

Stage 0: Local encoding and assembly. We split the document into sliding-window chunks and encode each chunk independently. Hidden states from overlapping windows are mapped back to their original token positions and averaged, producing an assembled sequence $\mathbf{H} \in \mathbb{R}^{L \times d}$. This stage preserves the encoder’s dense local contextualization while retaining every document position for subsequent global interaction and compression.

4.1 Stage 1: Linear Attention for Global Context

Given the assembled hidden states $\mathbf{H} \in \mathbb{R}^{L \times d}$, we model interactions across all L positions using Performer-style generalized linear attention (Choromanski et al., 2021).

The attention mechanism (Vaswani et al., 2017) is $\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\mathbf{Q}\mathbf{K}^\top / \sqrt{d})\mathbf{V}$, which requires $\mathcal{O}(L^2d)$ time and $\mathcal{O}(L^2)$ memory for $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{L \times d}$. Performer-style attention uses a feature map $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^r$ to enable linear attention:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \approx \frac{\phi(\mathbf{Q})(\phi(\mathbf{K})^\top \mathbf{V})}{\phi(\mathbf{Q})(\phi(\mathbf{K})^\top \mathbf{1}_L)}, \quad (1)$$

reducing the time and memory complexity to $\mathcal{O}(Ldr)$ and $\mathcal{O}(Lr)$, respectively. The denominator is applied row-wise, once per query token. See Appendix A.1 for details.

Query, key, and value matrices are projected to low rank: $\mathbf{Q}' = \mathbf{H}\mathbf{W}_q$, $\mathbf{K}' = \mathbf{H}\mathbf{W}_k$, and $\mathbf{V}' = \mathbf{H}\mathbf{W}_v \in \mathbb{R}^{L \times r}$, where $r \ll d$, using trainable projection matrices $\mathbf{W}_{\{q,k,v\}} \in \mathbb{R}^{d \times r}$. Following Performer’s generalized-attention formulation, we

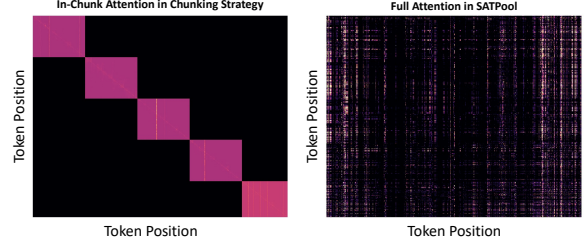


Figure 3: **A comparison of attention mechanisms on a long document.** (Left) Independent chunk encoding confines interactions within each chunk, producing a block-diagonal attention pattern. (Right) SATPool applies linear attention over the assembled sequence, allowing every token position to exchange information across chunk boundaries.

use the positive feature map $\phi(\mathbf{x}) = \text{ReLU}(\mathbf{x}) + \epsilon$. The resulting globally reweighted sequence is

$$\begin{aligned} \mathbf{H}^{\text{attn}} &= \hat{\mathbf{D}}^{-1} \phi(\mathbf{Q}') \left[\phi(\mathbf{K}')^\top \mathbf{V}' \right] \in \mathbb{R}^{L \times r}, \quad (2) \\ \hat{\mathbf{D}} &= \text{diag} \left(\phi(\mathbf{Q}') \left[\phi(\mathbf{K}')^\top \mathbf{1}_L \right] \right). \end{aligned}$$

This linear formulation overcomes the quadratic bottleneck of standard attention, making global interaction scalable to very long sequences. As illustrated in Fig. 3, the linear-attention stage bridges the fragmented chunk representations before document-level compression occurs. It scans the entire assembled sequence and propagates information between distant regions, allowing evidence introduced in one chunk to influence representations in every other chunk. SATPool can therefore model document-wide dependencies while retaining linear time and memory complexity in L .

4.2 Stage 2: Spectral Token Compression

To transform the variable-length token sequence $\mathbf{H}^{\text{attn}} \in \mathbb{R}^{L \times r}$ into a compact, fixed-size vector representation, we propose *Spectral Token Compression (STC)*, a pooling strategy that captures both semantic content and structural dynamics of the document. Unlike mean pooling or [CLS]-token aggregation, which treat the sequence as an unordered set of embeddings, STC preserves position-sensitive information by interpreting the sequence as a multidimensional discrete signal.

Specifically, each token embedding $\mathbf{H}_t^{\text{attn}} \in \mathbb{R}^r$ at position $t = 1, \dots, L$ is treated as a point sampled from an r -dimensional signal. STC first captures global semantic content through the mean

$$\bar{\mathbf{H}}^{\text{attn}} = \frac{1}{L} \sum_{t=1}^L \mathbf{H}_t^{\text{attn}}.$$

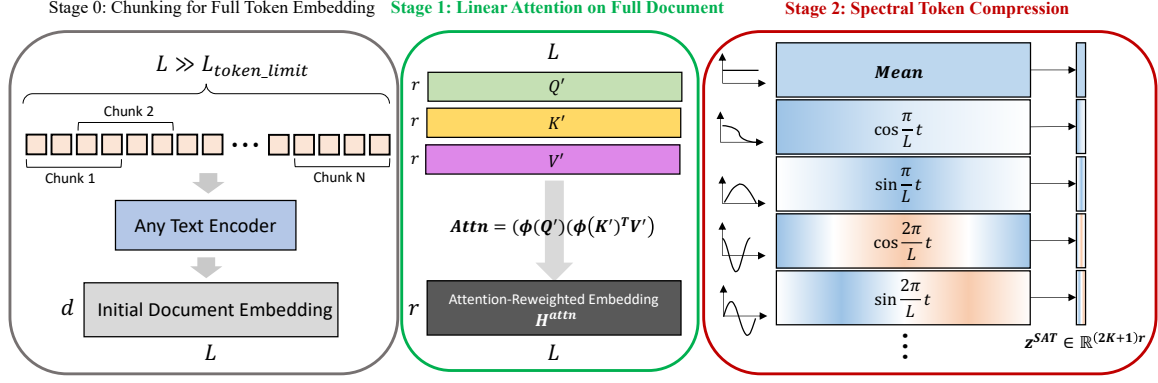


Figure 4: **Overview of SATPool.** **Stage 0:** Independently encode chunks and assemble locally contextualized token embeddings, averaging overlaps at their original positions. **Stage 1:** Apply linear attention across the assembled sequence to obtain $\mathbf{H}^{\text{attn}}$. **Stage 2:** STC concatenates the mean with projections onto K fixed sine and cosine bases, yielding $\mathbf{z}^{\text{SAT}} \in \mathbb{R}^{(2K+1)r}$.

It then captures positional structure by projecting the sequence onto K sine and cosine bases with frequencies $\omega_k = \frac{\pi k}{L}$. For $k = 1, \dots, K$,

$$\mathbf{C}_k = \alpha_L \sum_{t=1}^L \mathbf{H}_t^{\text{attn}} \cos(\omega_k t), \quad \mathbf{S}_k = \alpha_L \sum_{t=1}^L \mathbf{H}_t^{\text{attn}} \sin(\omega_k t) \quad (3)$$

The fixed sine and cosine bases provide a compact set of positional patterns spanning different frequencies. Because $\mathbf{H}^{\text{attn}}$ is produced by learned, content-dependent attention, its coefficient vectors $\{\mathbf{C}_k, \mathbf{S}_k\}$ encode how semantic features are distributed across relative document positions. Low-frequency components capture broad transitions between regions such as the introduction, body, and conclusion, while higher-frequency components capture finer changes. Their combination enables STC to represent multiple salient regions without collapsing the sequence into an order-invariant average.

We use $\alpha_L = 1$ in the main experiments. Table 8 compares this raw-sum formulation with the alternative length-normalized formulation, $\alpha_L = 1/L$. Both formulations outperform mean pooling across all evaluated configurations, and neither consistently dominates the other. Finally, the mean vector and the $2K$ spectral coefficient vectors are concatenated:

$$\mathbf{z}^{\text{SAT}} = [\bar{\mathbf{H}}^{\text{attn}}, \mathbf{C}_1, \mathbf{S}_1, \dots, \mathbf{C}_K, \mathbf{S}_K] \in \mathbb{R}^{(2K+1)r} \quad (4)$$

This fixed-size vector preserves global semantics through the mean and document structure through the spectral components. A subsequent projection layer $\mathbf{W}_o \in \mathbb{R}^{(2K+1)r \times d}$ learns the task-specific combination of these complementary features and

produces the final document embedding:

$$\mathbf{z} = \mathbf{z}^{\text{SAT}} \mathbf{W}_o \in \mathbb{R}^d.$$

Its dimensionality is independent of the input length L , making it directly applicable to classification and dense retrieval.

4.3 Analyzing STC with Visualization

Fig. 5 visualizes how STC transforms a document into spectral components. (a) The example document describes “Earth,” with the first two paragraphs serving as its introduction. Each paragraph occupies a contiguous interval in the token sequence, providing a direct correspondence between document content and position.

(b) STC projects the attention-reweighted sequence onto the fixed sine and cosine bases in Eq. (3). Each frequency captures a different positional pattern: low frequencies describe broad document-scale variation, whereas higher frequencies capture more localized changes.

(c) Each projection produces a vector coefficient $\mathbf{C}_k$ or $\mathbf{S}_k \in \mathbb{R}^r$. We summarize its magnitude using the ℓ_2 norm, shown in the top bar chart, and multiply the corresponding basis by that magnitude, for example, $\|\mathbf{C}_k\|_2 \cos(\omega_k t)$. The middle plot shows these weighted components together with their sum as the black curve, and the bottom heatmap presents the same reconstructed pattern along the document. The strongest response falls on the introductory region, showing how the combined spectral components preserve a content-linked positional pattern within the compact document representation. Appendix B repeats the analysis after reversing the paragraph order and shows

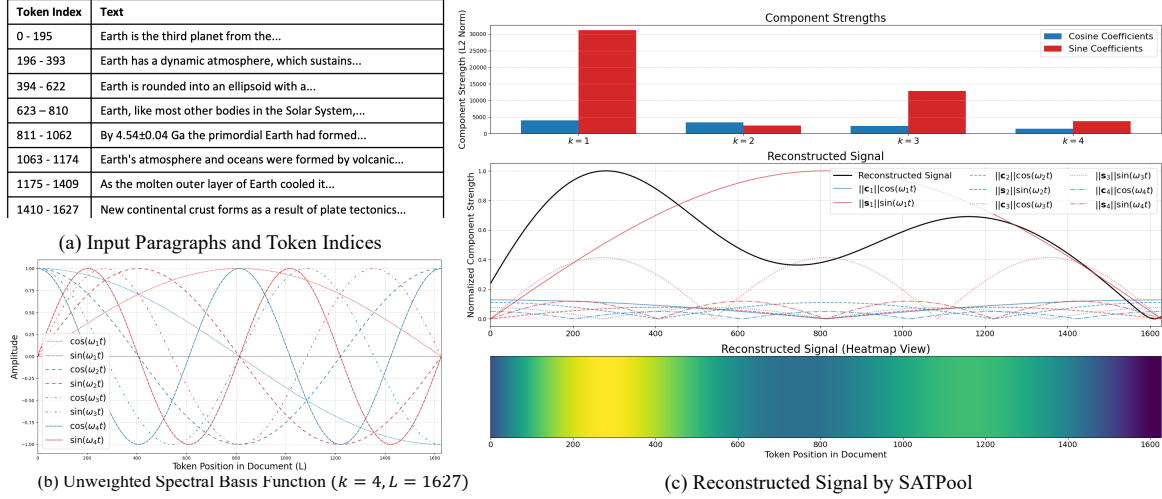


Figure 5: **Illustration of Spectral Token Compression (STC).** (a) The input document is represented as a sequence of paragraphs and token positions. (b) STC projects the attention-reweighted token sequence onto fixed sine and cosine basis functions. (c) The ℓ_2 norms of the resulting coefficient vectors summarize the strength of each spectral component (top); their norm-weighted combination forms the reconstructed pattern (middle), also shown as a heatmap (bottom). In this example, the strongest response aligns with the introductory paragraphs.

that the response moves with the introductory content rather than remaining fixed at the beginning of the sequence.

5 Implementation Details

We validate SATPool on text classification, RAG, and Multimodal RAG, using frozen pretrained encoders so that the comparison isolates pooling and aggregation. SATPool and the task head are trained separately for each dataset. Classification uses cross-entropy for 20NewsGroups and binary cross-entropy for EURLEX; retrieval uses supervised contrastive learning over annotated positives and nonmatching in-batch documents. The Qwen generators are used without fine-tuning. Across all main experiments, SATPool and the task head are trained for 50 epochs with batch size 16 using Adam; the learning rate is 10^{-4} for classification and CLIP-DPR, and 10^{-5} for text RAG and RAVQA. Additional dataset and optimization details are provided in Appendix C. We report mean performance here and provide 1,000-resample bootstrap analyses in Appendix L.

5.1 Comparisons

We evaluate SATPool against two main categories of methods for handling long documents. First, we compare against specialized long-context models that are architecturally designed to process long sequences. For text classification, this includes models like Longformer (Beltagy et al., 2020) and

LongT5 (Guo et al., 2022), as well as foundational models BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019).

Second, we compare against model-agnostic approaches that are designed as plug-and-play modules for standard encoders like SATPool. Our primary comparison in this category is Late Chunking (Günther et al., 2024), a versatile method that we evaluate across text classification, RAG, and multimodal RAG tasks. We also compare against hierarchical aggregation methods, specifically PARADE (Li et al., 2023a) and TrLDC (Dai et al., 2022), which utilize transformer-based layers to aggregate chunk information. Additionally, we benchmark against standard baseline strategies, including simple mean pooling, MaxP (Dai and Callan, 2019), and using the [CLS] token representation from the encoder. The details of these comparison methods are introduced in Appendix F.

5.2 Details in Generator

To demonstrate flexibility with modern generative backbones, we use Qwen2.5-3B-Instruct (Yang et al., 2024) for RAG and the Large Multimodal Model Qwen2.5-VL-3B-Instruct (Wang et al., 2024b) for multimodal RAG, providing retrieved documents as context without fine-tuning either generator.

5.3 Evaluation Metrics

We use task-specific metrics following standard practice, with full definitions provided in Ap-

Methods	20NewsGroups				EURLEX			
	BERT	RoBERTa	Longformer	LongT5	BERT	RoBERTa	Longformer	LongT5
CLS	0.5916	0.5890	0.5798	0.2606	0.5645	0.4975	0.4933	0.4411
Mean	<u>0.6223</u>	<u>0.6315</u>	<u>0.6386</u>	<u>0.5637</u>	<u>0.6856</u>	0.6515	0.6644	<u>0.6921</u>
LC	0.6166	0.6115	0.6219	0.4909	0.6763	<u>0.6708</u>	<u>0.6731</u>	0.4960
PARADE	0.5819	0.6070	0.6258	0.2318	0.6318	<u>0.6484</u>	<u>0.6609</u>	0.3857
TrLDC	0.5754	0.5705	0.5733	0.2203	0.6277	0.6341	0.6431	0.4969
SATPool	0.6371	0.6570	0.6450	0.6195	0.7124	0.7266	0.7232	0.7306

Table 2: Experimental results (F1 score) for text classification on the 20NewsGroups and EURLEX datasets. **Bold** indicates the best result in each setting, while underline denotes the second best.

Method	NQ				HotPotQA			
	BERT		GTR		BERT		GTR	
	Recall@5	AnswerRecall	Recall@5	AnswerRecall	Recall@5	AnswerRecall	Recall@5	AnswerRecall
CLS	0.2038	0.1984	<u>0.8099</u>	0.3511	0.0628	0.2231	0.5926	0.3309
Mean	0.4595	0.2601	0.7874	<u>0.3528</u>	0.2148	0.2611	0.5135	0.3139
MaxP	0.3145	0.2266	0.6359	0.3165	0.1340	0.2455	0.1340	0.2455
LC	<u>0.4688</u>	<u>0.2605</u>	0.6418	0.3014	<u>0.2345</u>	<u>0.2620</u>	0.4228	0.2971
PARADE	0.3425	<u>0.2281</u>	0.7110	0.3275	0.1019	<u>0.2261</u>	0.2790	0.2582
TrLDC	0.3060	0.2323	0.3380	0.2330	0.0914	0.2239	0.0271	0.2143
SATPool	0.5622	0.2940	0.8175	0.3676	0.2466	0.2693	<u>0.5744</u>	<u>0.3221</u>

Table 3: Experimental results for RAG on the NQ and HotPotQA datasets. **Bold** indicates the best result for each setting, while underline denotes the second best.

pendix D. For text classification and factual consistency tasks, we report F1 scores because ground-truth labels are available. For retrieval-augmented tasks, we evaluate retrieval and generation separately. Specifically, we use Recall@K for datasets with annotated relevant passages and PRRecall@K for OKVQA, where retrieving at least one relevant passage is often sufficient for answering the query. For generation quality, we report Answer Recall rather than Exact Match or F1, as LLM-generated responses may contain the correct answer within longer free-form outputs.

6 Result Analysis

6.1 Experimental Results

As shown in Tables 2, 3, and 4, SATPool achieves strong overall performance across classification, RAG, and multimodal RAG. The gains are particularly clear in classification and retrieval, where success depends directly on whether the pooled representation preserves information distributed throughout the document. These results demonstrate that modeling global token interactions before compression is more effective than aggregating independently constructed chunk representations.

The qualitative examples in Fig. 1 illustrate this distinction. SATPool retrieves documents that match the complete semantic intent of the query,

including relationships expressed across multiple phrases, whereas mean pooling is more likely to retrieve documents sharing only broad topics or isolated keywords. This behavior supports the central motivation of SATPool: a document representation should preserve interactions among distant pieces of evidence rather than merely summarize their average content.

The controlled NQ analysis in Table 6 further highlights the practical advantage of document-level pooling. SATPool represents each document with a single vector while achieving stronger parent-document retrieval than chunk-in-isolation indexing, which requires a substantially larger search index. SATPool therefore improves both representation quality and index efficiency by integrating document-wide evidence before retrieval.

The design analysis in Appendix I explains these empirical gains through three properties of SATPool. First, dense local encoding preserves fine-grained semantic information within each chunk, and linear attention subsequently propagates that information across chunk boundaries. Second, STC retains coarse positional variation that is removed by permutation-invariant mean pooling. Third, SATPool follows an “Attend-then-Pool” strategy, delaying compression until global interactions have been modeled, unlike hierarchical “Pool-then-

Method	OKVQA				EVQA			
	CLIP-DPR		RAVQA		CLIP-DPR		RAVQA	
	PRR@5	AnswerRecall	PRR@5	AnswerRecall	PRR@5	AnswerRecall	PRR@5	AnswerRecall
CLS	0.1371	0.4065	0.0551	0.3983	0.2588	0.2496	0.0535	0.1117
Mean	0.1424	0.4048	0.0817	0.4045	0.2338	0.2367	0.0843	0.1378
MaxP	0.0745	<u>0.4101</u>	0.0676	0.4054	0.1730	0.2005	0.0445	0.1168
LC	0.1321	0.4022	0.0727	0.4009	0.2385	0.2350	0.0767	0.1231
PARADE	0.1249	0.3965	0.0855	0.4166	0.2407	0.2368	0.0908	<u>0.1383</u>
TrLDC	0.1421	0.4100	<u>0.0864</u>	0.4045	0.2200	0.2293	0.0797	0.1244
SATPool	0.1564	0.4148	0.0964	<u>0.4115</u>	<u>0.2585</u>	0.2592	<u>0.0900</u>	0.1444

Table 4: Experimental results for multimodal RAG on the OKVQA and EVQA datasets. **Bold** indicates the best result for each setting, while underline denotes the second best.

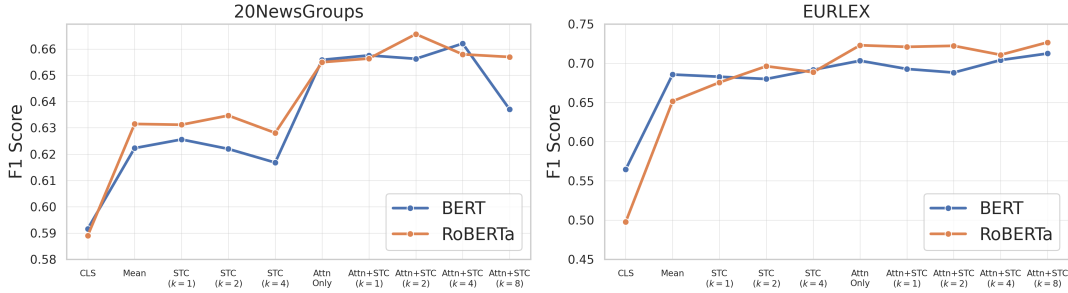


Figure 6: **Ablation results on 20NewsGroups and EURLEX.** Adding full attention and STC improves classification performance over [CLS] pooling with frozen BERT and RoBERTa.

Attend” approaches that compress each chunk before document-level aggregation.

The computational analysis in Appendix H shows that these gains do not require the cost of a specialized long-context encoder. SATPool maintains near mean-pooling runtime with modest memory overhead, while remaining substantially more efficient than heavier long-context alternatives. Its computational cost is primarily governed by the low-rank attention dimension r , whereas the number of spectral components K can be adjusted with little effect on overall FLOPs.

6.2 Ablation Study

To validate the contribution of each SATPool component, including linear attention, STC, and the choice of K , we compare [CLS], mean pooling, STC only, attention only, and SATPool variants across multiple datasets and encoder backbones. Fig. 6 presents the complete classification ablation, while Table 5 summarizes the contribution of STC beyond the Attn-only configuration in both classification and retrieval.

As shown in Fig. 6, linear attention accounts for a substantial share of the classification improvement, demonstrating the importance of modeling

interactions across the assembled token sequence. Table 5(a) further shows that adding STC improves the best classification result in every setting. The optimal value of K varies across datasets and backbones, indicating that STC provides complementary positional information at a task-dependent spectral resolution.

Table 5(b) shows that the contribution of STC extends to retrieval. On NQ, Attn+STC improves over Attn-only for every evaluated K with both BERT and GTR. The corresponding OKVQA component ablation in Appendix G.1 shows that Attn+STC consistently improves pseudo-relevance recall, while answer recall varies with spectral resolution. Together, these results indicate that linear attention provides cross-chunk interactions and that STC adds complementary positional structure. Full classification, NQ, and OKVQA component ablations are provided in Appendix G.1.

Table 6 makes the index-size trade-off explicit. SATPool achieves the strongest parent-document retrieval while indexing a single vector per document, whereas the chunk-in-isolation baselines substantially expand the index by representing each chunk separately. This result shows that SATPool

(a) Classification component contribution				(b) NQ retrieval component contribution			
Setting	Attn-only	Attn+STC (best K)	Δ	Method	BERT		GTR
20News, BERT	0.6559	0.6621	+0.0062		R@5	R@10	R@5 R@10
20News, RoBERTa	0.6550	0.6657	+0.0107	Attn-only	0.1798	0.2627	0.6898 0.7640
EURLEX, BERT	0.7032	0.7124	+0.0092	Attn+STC, $K = 1$	0.1900	0.2672	0.7757 0.8424
EURLEX, RoBERTa	0.7229	0.7266	+0.0037	Attn+STC, $K = 2$	0.1858	0.2699	0.7384 0.8078
				Attn+STC, $K = 4$	0.2164	0.3019	0.7780 0.8451

Table 5: **Component contributions in classification and retrieval.** (a) Linear attention provides most of the improvement, while the best Attn+STC configuration across the evaluated K values adds a smaller positive gain in each classification setting. (b) On NQ, adding STC to linear attention improves retrieval for every reported K with both encoder backbones. Full classification, NQ, and OKVQA component ablations are provided in Appendix G.1.

Method	Units	R@1	R@5	R@10
CLS document	1,153	0.1367	0.2883	0.3950
SATPool document	1,153	0.1450	0.3450	0.4667
CLS chunk-512	10,128	0.0883	0.2617	0.3658
CLS chunk-256	19,600	0.1083	0.2842	0.3692
CLS chunk-128	38,621	0.0883	0.3217	0.4367

Table 6: **Controlled NQ retrieval-unit diagnostic.** Document rows index one vector per document; chunk rows index independently encoded chunks and use the maximum chunk score for parent-document ranking.

Length	n	CLS R@5	SATPool R@5	Δ
≤ 512	208	0.0994	0.1611	+0.0617
513–1,024	334	0.1003	0.1411	+0.0408
1,025–2,048	483	0.0932	0.1534	+0.0602
2,049–4,096	688	0.0790	0.1793	+0.1003
$> 4,096$	1,124	0.1021	0.2231	+0.1210

Table 7: **Length-stratified NQ retrieval.** SATPool improves R@5 in every document-length group, with the largest gains occurring in the two longest groups. Full NQ answer-recall results and the corresponding 20NewsGroups analysis are provided in Appendix G.2.

can preserve document-wide evidence within a compact retrieval representation.

Table 7 shows that SATPool’s R@5 gains are largest in the two longest NQ document groups. This result is consistent with the motivation for global pooling, as longer documents may provide more opportunities for relevant information to occur across distant chunks. In contrast, the 20NewsGroups analysis in Appendix G.2 shows positive gains in every length group, but no monotonic increase with document length. Taken together, these results show that SATPool remains beneficial across the evaluated length groups, while document length alone does not consistently predict the magnitude of its improvement. Full NQ answer-recall results are also provided in Appendix G.2.

In Eq. (3), the raw-sum formulation uses $\alpha_L = 1$, whereas the length-normalized formulation uses

Backbone	K	Mean	Raw (gain)	Norm. (gain)
BERT	2	0.6223	0.6563 (+0.0340)	0.6681 (+0.0458)
BERT	4	0.6223	0.6621 (+0.0398)	0.6424 (+0.0201)
BERT	8	0.6223	0.6371 (+0.0148)	0.6366 (+0.0143)
RoBERTa	2	0.6315	0.6657 (+0.0342)	0.6590 (+0.0275)
RoBERTa	4	0.6315	0.6580 (+0.0265)	0.6593 (+0.0278)
RoBERTa	8	0.6315	0.6570 (+0.0255)	0.6682 (+0.0367)

Table 8: **STC coefficient-scaling ablation on 20NewsGroups.** Raw and Norm. denote the raw-sum and length-normalized formulations, respectively. Both formulations outperform mean pooling for every evaluated K and backbone; values in parentheses report absolute F1 gains over the corresponding mean-pooling baseline.

$\alpha_L = 1/L$, dividing each sine and cosine coefficient by the sequence length. Table 8 shows that both formulations outperform mean pooling across all evaluated spectral resolutions and backbones. Neither scaling consistently dominates the other, indicating that SATPool’s improvement is not tied to a particular coefficient-scaling convention.

7 Conclusion

In this work, we address the challenge of constructing document-level representations when pre-trained encoders have limited context windows. We introduce Spectral Attention Token Pooling (SATPool), a lightweight, two-stage, plug-and-play module for pretrained text encoders. SATPool applies efficient linear attention over assembled, locally encoded tokens and then uses Spectral Token Compression to produce a fixed-size vector. Experiments across long-document classification, RAG, and multimodal RAG demonstrate strong overall performance. Component analyses identify linear attention as a major contributor and STC as a complementary, task-dependent component. SATPool thus provides a practical document-level aggregation method, rather than a replacement for full-document encoding or passage-level retrieval.

Limitations

While SATPool demonstrates strong empirical performance, we acknowledge the following limitations:

Stage-0 Information and Practical Length

SATPool operates on independently encoded chunks. It can model interactions after local encoding, but it cannot recover information that the frozen encoder failed to capture within a chunk and is not equivalent to full-document self-attention from the outset. Moreover, SATPool retains and processes the assembled token sequence, so runtime and memory grow linearly with document length and may become substantial for extremely long documents.

Language Scope Our evaluations are primarily conducted on English datasets, such as 20News-Groups, EURLEX, NQ, and HotPotQA. Because STC uses a harmonic positional prior, its effectiveness may vary across languages, genres, discourse organizations, and weakly ordered document types. These settings remain to be evaluated.

Generative Context Constraints In our RAG and Multimodal RAG pipelines, SATPool is designed strictly to improve the retrieval phase by supplying more accurate and contextually intact documents. However, it does not modify the architecture of the downstream generator. Consequently, the overall system performance remains dependent on the generative backbone’s inherent ability to effectively process and attend to the extended context provided by the concatenated retrieved documents.

Base Model and Retrieval Scope SATPool operates on top of frozen encoders, so performance remains constrained by their semantic quality, vocabulary, and biases. SATPool also produces document-level vectors rather than fine-grained retrieval units; the present experiments do not show that it should replace chunk-in-context retrieval.

Ethical Considerations

SATPool avoids full-encoder fine-tuning in our experiments, but still requires task-specific training and additional computation over all assembled token representations.

Because SATPool represents content that truncation may omit, it can surface useful evidence but can also preserve harmful, biased, or low-quality content from later document regions. This may

affect which sources retrieval systems expose to users. Future deployments should audit retrieval quality and bias across languages, domains, and document positions, and consider how document-level pooling interacts with factuality verification and model alignment.

Acknowledgements

This work was partially supported by Purdue’s Elmore ECE Emerging Frontiers Center, and NSF IIS #2146091, IIS #2345235.

References

- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Ilias Chalkidis, Emmanouil Fergadiotis, Prodomos Malakasiotis, and Ion Androutsopoulos. 2019. [Large-scale multi-label text classification on EU legislation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6314–6322, Florence, Italy. Association for Computational Linguistics.
- Krzysztof Marcin Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021. [Rethinking attention with performers](#). In *International Conference on Learning Representations*.
- Xiang Dai, Ilias Chalkidis, Sune Darkner, and Desmond Elliott. 2022. Revisiting transformer-based models for long document classification. *arXiv preprint arXiv:2204.06683*.
- Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 985–988.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.
- Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. 2024. Late chunking: contextual chunk embeddings using long-context embedding models. *arXiv preprint arXiv:2409.04701*.
- Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. [LongT5: Efficient text-to-text transformer for long sequences](#). In *Findings of the Association for*

- Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.
- Ziyan Jiang, Xueguang Ma, and Wenhui Chen. 2024. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*.
- Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. 2024. Long-context llms meet rag: Overcoming challenges for long inputs in rag. *arXiv preprint arXiv:2410.05983*.
- Glenn Jocher, Jing Qiu, and Ayush Chaurasia. 2023. [Ultralytics YOLO](#).
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Mohamed Khemakhem, Houcem Eddine Rekik, and Oussama Bouaziz. 2024. [Enhancing Technical Knowledge Acquisition with RAG Systems: the TEI use case](#). In *Texts, Languages and Communities - TEI 2024*, Buenos Aires, Argentina. Universidad del Salvador (USAL), Buenos Aires, Argentina and Text Encoding Initiative (TEI).
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.
- Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:453–466.
- Ken Lang. 1995. Newsweeder: Learning to filter net-news. In *Machine learning proceedings 1995*, pages 331–339. Elsevier.
- Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoyebi, Bryan Catanzaro, and Wei Ping. 2025. [Nv-embed: Improved techniques for training llms as generalist embedding models](#). In *International Conference on Learning Representations*, volume 2025, pages 79310–79333.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- 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. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. [Llava-onevision: Easy visual task transfer](#). *Preprint*, arXiv:2408.03326.
- Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2023a. Parade: Passage representation aggregation for document reranking. *ACM Transactions on Information Systems*, 42(2):1–26.
- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.
- Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024b. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 881–893.
- Weizhe Lin and Bill Byrne. 2022. Retrieval augmented visual question answering with outside knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11238–11254.
- Weizhe Lin, Jingbiao Mei, Jinghong Chen, and Bill Byrne. 2024. Preflrmr: Scaling up fine-grained late-interaction multi-modal retrievers. *arXiv preprint arXiv:2402.08327*.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

- Weijie Liu, Zecheng Tang, Juntao Li, Kehai Chen, and Min Zhang. 2024. Memlong: Memory-augmented retrieval for long text modeling. *arXiv preprint arXiv:2408.16967*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Zijun Long, Xuri Ge, Richard McCreadie, and Joe-mon M Jose. 2024. Cfir: Fast and effective long-text to image retrieval for large corpora. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2188–2198.
- Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204.
- Thomas Mensink, Jasper Uijlings, Lluís Castrejon, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, André Araujo, and Vittorio Ferrari. 2023. Encyclopedic vqa: Visual questions about detailed properties of fine-grained categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3113–3124.
- Ramesh Nallapati, Bowen Zhou, Cicero Dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pages 280–290.
- Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.
- Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Hyunji Hayley Park, Yogarshi Vyas, and Kashif Shah. 2022. Efficient classification of long documents using transformers. *arXiv preprint arXiv:2203.11258*.
- Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Stewart Kirubakaran S, Jasper Wilsie Kathrine G, Grace Mary Kanaga E, Mahimai Raja J, Ruban Gino Singh A, and Yuvaraajan E. 2024. A rag-based medical assistant especially for infectious diseases. In *2024 International Conference on Inventive Computation Technologies (ICICT)*, pages 1128–1133.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. *arXiv preprint arXiv:2004.04228*.
- Junmei Wang, Jimmy X Huang, and Jinhua Sheng. 2024a. An efficient long-text semantic retrieval approach via utilizing presentation learning on short-text. *Complex & Intelligent Systems*, 10(1):963–979.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024b. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, Chunrui Han, and Xianguyu Zhang. 2024. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. *Preprint*, arXiv:2409.01704.
- Nirmalie Wiratunga, Ramitha Abeyratne, Lasal Jayawardena, Kyle Martin, Stewart Massie, Ikechukwu Nkisi-Orji, Ruvan Weerasinghe, Anne Liret, and Bruno Fleisch. 2024. Cbr-rag: case-based reasoning for retrieval augmented generation in llms for legal question answering. In *International Conference on Case-Based Reasoning*, pages 445–460. Springer.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and

43 others. 2024. [Qwen2 technical report](#). *Preprint*, arXiv:2407.10671.

Junhan Yang, Zheng Liu, Chaozhuo Li, Guangzhong Sun, and Xing Xie. 2023. Longtriever: a pre-trained long text encoder for dense document retrieval. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3655–3665.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zhengxuan Zhang, Yin Wu, Yuyu Luo, and Nan Tang. 2025. Fine-grained retrieval-augmented generation for visual question answering. *arXiv preprint arXiv:2502.20964*.

Qingfei Zhao, Ruobing Wang, Yukuo Cen, Daren Zha, Shicheng Tan, Yuxiao Dong, and Jie Tang. 2024. Longrag: A dual-perspective retrieval-augmented generation paradigm for long-context question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22600–22632.

A A Deeper Look into Performer and Linear Attention

A.1 Performer Attention (Choromanski et al., 2021)

For query, key, and value matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{L \times d}$, the original attention mechanism computes

$$\text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} \right) \mathbf{V}.$$

Equivalently, it can be written as

$$\begin{aligned} \text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \mathbf{D}^{-1} \mathbf{A} \mathbf{V} \\ \mathbf{A} &= \exp \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} \right) \\ \mathbf{D} &= \text{diag}(\mathbf{A} \mathbf{1}_L) \end{aligned}$$

This computation requires $\mathcal{O}(L^2 d)$ time and $\mathcal{O}(L^2)$ memory, which is prohibitive for long sequences. To reduce this cost, Performer approximates the softmax kernel $SM(x, y) = \exp(x^\top y)$ using a kernel of the form

$$K(x, y) = \mathbb{E}_\omega [\phi(x)^\top \phi(y)],$$

where $\phi : \mathbb{R}^d \rightarrow \mathbb{R}_+^r$ is a random feature map. This approximation is unbiased, i.e., the expectation of the dot product of the random features equals the softmax kernel. Choromanski et al. (2021) proposes a provable positive random feature map:

$$\phi(x) = \exp(\omega^\top x - \|x\|^2/2), \quad \omega \sim \mathcal{N}(0, I),$$

which satisfies the unbiasedness condition for approximating the softmax kernel.

In addition, Choromanski et al. (2021) explored generalized attention mechanisms, where other positive feature maps can be used for efficiency. For instance, they show that a simple choice like

$$\phi(x) = \text{ReLU}(x) + \epsilon$$

can be used as a kernel approximation, even though it does not approximate softmax theoretically. This choice is empirically effective, particularly for long sequence tasks like protein modeling.

Finally, using the decomposition $\mathbf{Q}' = \phi(\mathbf{Q})$, $\mathbf{K}' = \phi(\mathbf{K})$, the approximate attention becomes

$$\begin{aligned} \text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &\approx \hat{\mathbf{D}}^{-1} \mathbf{Q}' (\mathbf{K}'^\top \mathbf{V}) \\ \hat{\mathbf{D}} &= \text{diag}(\mathbf{Q}' (\mathbf{K}'^\top \mathbf{1}_L)) \end{aligned}$$

This formulation reduces the time complexity to $\mathcal{O}(Lrd)$ and the space complexity to $\mathcal{O}(Lr + Ld + rd)$, where r is the number of random features.

B Robustness of Spectral Token Compression

To validate the robustness of STC in identifying important regions within a document, we intentionally alter the order of the document introduced in Figure 5. Specifically, we reverse the order of its paragraphs, moving the original introduction to the second half of the text. Since STC previously highlighted the introduction at the beginning of the document, a robust model should now identify this same content in its new position.

As shown in Fig. 7, the illustrative curve shifts toward the relocated introductory content rather than remaining fixed at the beginning. This single qualitative example is consistent with content sensitivity, but does not establish that coefficient norms are calibrated token-importance scores.

C Implementation Details

Long Text Classification. For multi-class classification, we use the 20NewsGroups (Lang, 1995)

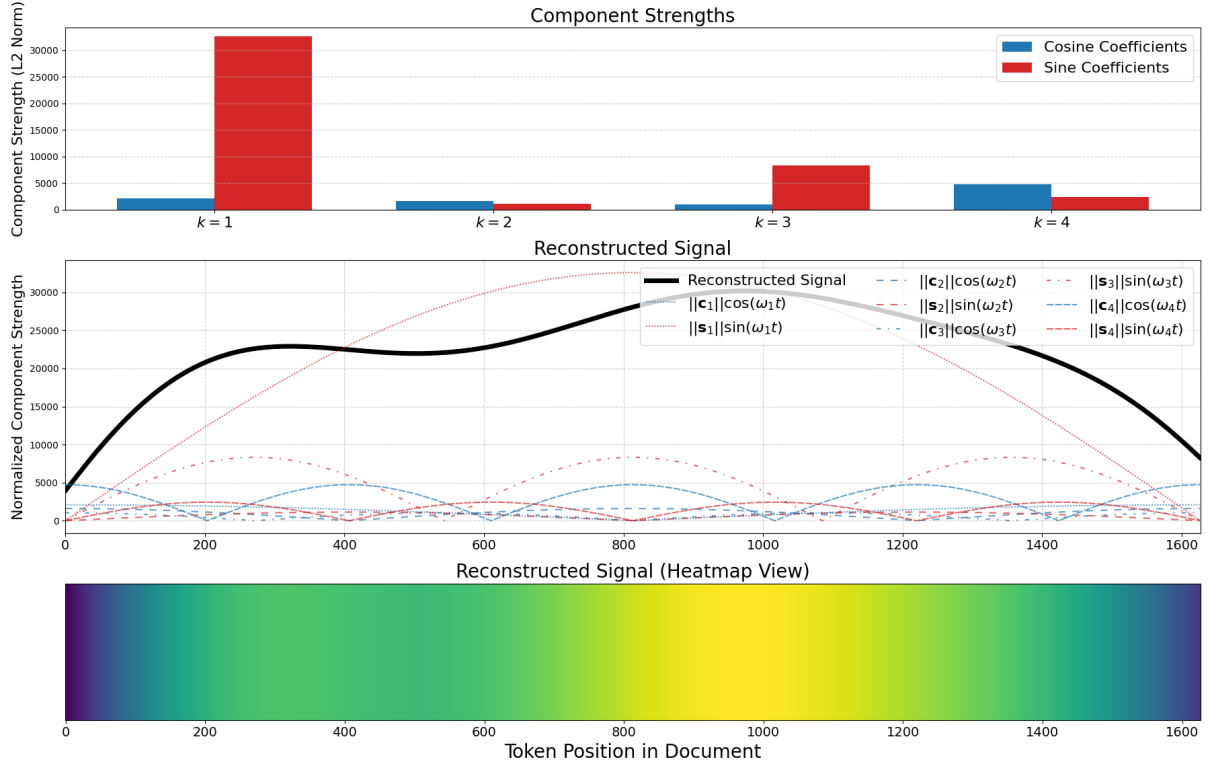


Figure 7: **Qualitative paragraph-order analysis of STC.** We reverse the paragraphs from Fig. 5, moving the original introduction to the latter half. The illustrative coefficient-norm-weighted curve shifts with that content. This example suggests content sensitivity, but is not a ground-truth importance evaluation.

dataset, which consists of 11.3k training and 7.5k test instances. Moreover, for multi-label classification, we use the EURLEX (Chalkidis et al., 2019) dataset of EU legal documents, as used in (Park et al., 2022). For the EURLEX dataset, we utilize the English version, which contains 55k training and 5k test instances; for computational efficiency, we sample 10k instances for training in all experiments.

Our classification setup uses frozen pretrained encoders (BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), Longformer (Beltagy et al., 2020), and LongT5 (Guo et al., 2022)) followed by SATPool and a three-layer MLP classifier with hidden dimension 256. SATPool and the classifier are trained separately for each dataset for 50 epochs using Adam, a learning rate of 10^{-4} , and a batch size of 16. We use cross-entropy for multiclass 20News-Groups classification and binary cross-entropy for multilabel EURLEX classification. The pretrained encoder remains frozen.

RAG with Long Documents. We evaluate our RAG pipeline on the Natural Questions (NQ) (Kwiatkowski et al., 2019) and HotPotQA (Yang

et al., 2018) datasets, as used in (Li et al., 2024b; Petroni et al., 2021). Both use Wikipedia as their knowledge source. For efficiency, we use a subset of 20k training samples for both datasets, while the evaluation is conducted on full test set. Our pipeline consists of retriever training followed by generation with a frozen Qwen2.5-3B-Instruct model.

For the retriever, we build a DPR-style (Karpukhin et al., 2020) model using frozen BERT and GTR-T5 (Ni et al., 2022) encoders. SATPool and three-layer MLP projection heads with output dimension 512 are trained for queries and documents for 50 epochs with batch size 16, Adam, and a learning rate of 10^{-5} . We use supervised contrastive (SupCon) loss (Khosla et al., 2020): dataset annotations define positive query–document pairs, and nonmatching documents in the minibatch serve as negatives.

For the generator, we adopt Qwen2.5-3B-Instruct (Yang et al., 2024) model without any fine-tuning. Since the full text of retrieved documents exceeds the generator’s input capacity, the top 5 retrieved documents are first summarized us-

ing a pre-trained BART (Lewis et al., 2019) model. These summaries are then concatenated and provided as context to the generator.

Multimodal RAG with Long Documents. For multimodal RAG, we use the OKVQA (Marino et al., 2019) and EVQA (Mensink et al., 2023) datasets, following the setup suggested in (Lin et al., 2024). The OKVQA dataset pairs 9.0k training and 5.1k test samples with 115k documents, while EVQA contains 167k training and 3.8k test samples paired with 50.2k documents.

We evaluate a CLIP-based DPR model and RAVQA (Lin and Byrne, 2022), whose query encoder incorporates OCR text (Wei et al., 2024), object detections (Jocher et al., 2023), and image descriptions (Li et al., 2022). We freeze the text encoder (CLIP for CLIP-DPR and BERT for RAVQA) and train SATPool and three-layer MLP projection heads with hidden dimension 512 for queries and documents. CLIP-DPR uses a learning rate of 10^{-4} and RAVQA uses 10^{-5} ; both use Adam, 50 epochs, batch size 16, and the same SupCon positive and in-batch-negative construction as text RAG. The Qwen2.5-VL-3B-Instruct generator (Wang et al., 2024b) is used without fine-tuning.

D Evaluation Metrics

For text classification and factual consistency tasks, we report the F1 score, as ground-truth labels are available in the test datasets. For both single-modal and multimodal retrieval-augmented tasks, we separately evaluate retrieval and generation performance, following (Lin and Byrne, 2022). Retrieval performance is measured using Recall@K or PRRecall@K, depending on the dataset structure, while generation performance is measured using Answer Recall.

D.1 Retrieval Metrics

Recall@K measures the fraction of ground-truth relevant documents that appear among the top- K retrieved candidates. Let P be the set of ground-truth passage IDs and $\hat{P}_K$ be the set of top- K retrieved passage IDs. Then,

$$\text{Recall@K} = \frac{|\hat{P}_K \cap P|}{|\hat{P}_K|}. \quad (5)$$

This metric provides a fine-grained view of how completely the model recovers all known relevant documents. It is particularly informative when multiple ground-truth passages are available.

Pseudo-Relevance Recall (PRRecall@K) is a binary variant of recall. It assigns a score of 1 if at least one ground-truth passage is retrieved among the top- K , and 0 otherwise:

$$\text{PRRecall@K} = \begin{cases} 1 & \text{if } \hat{P}_K \cap P \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

PRRecall@K focuses on whether the system retrieves any useful knowledge, regardless of the total number of relevant passages. This is especially useful in open-domain and knowledge-intensive VQA settings, where retrieving a single correct passage may be sufficient to answer the query. We use PRRecall@K exclusively for OKVQA, as its structure often provides multiple relevant documents for a single query.

D.2 Generation Metric

Answer Recall. We evaluate generated answers using Answer Recall, also known as Hit Rate. This metric measures whether the ground-truth answer appears in the generated response after text normalization. Specifically, we normalize both the ground-truth answer and the generated response by lowercasing, removing articles, and removing punctuation. A score of 1 is assigned if the normalized ground-truth answer appears as a substring of the normalized generated response, and 0 otherwise:

$$\text{Answer Recall}(y, \hat{y}) = \mathbb{1}[\text{norm}(y) \subseteq \text{norm}(\hat{y})], \quad (7)$$

where y denotes the ground-truth answer, $\hat{y}$ denotes the generated response, and $\text{norm}(\cdot)$ denotes the normalization function.

We use Answer Recall rather than Exact Match, token-level F1, or VQA Score because LLM-generated responses often contain the correct answer within longer free-form outputs. For example, the response “The capital is Paris” should be credited when the ground-truth answer is “Paris.” Answer Recall therefore captures whether the generated response includes the required answer while remaining stricter than purely semantic or paraphrase-based metrics.

E Additional Task: SATPool as a Long-Context Factuality Consistency Evaluator

In addition to creating representations for retrieval and classification, we repurpose SATPool to address a critical challenge in text generation:

factual-consistency evaluation. This task involves automatically estimating how faithfully a generated text, such as a summary, reflects the facts in its source document. Standard tools for this task include BERTScore (Zhang et al., 2019), which measures token-level semantic overlap using a pre-trained encoder, and FactCC-Eval (Kryściński et al., 2019), a classification model fine-tuned on BERT to predict whether a summary sentence is entailed or contradicted by the source (more details in BERTScore and FactCC are in Appendix F). However, a significant drawback of these approaches is that they inherit the 512-token input limit from their underlying BERT architecture. This forces them to truncate long source documents, meaning factual errors related to content beyond the initial segment may go undetected.

To overcome this, we introduce the **Dense Factuality Consistency Score (DFCScore)**, a model that leverages SATPool to process the entire document without truncation.

Method Given a document d and a candidate summary s , DFCScore computes a probability that the pair is factually consistent. Both inputs are processed by a frozen BERT encoder $\mathcal{E}$ to obtain token embeddings. These embeddings are then passed through separate processing heads, f_d and f_s , which each consist of a pooling layer followed by a three-layer MLP. For the document head f_d , we test both SATPool and a mean-pooling baseline as its pooling component. The final score is then calculated as:

$$\text{DFCScore}(d, s) = \sigma(w \cdot \cos(f_d(\mathcal{E}(d)), f_s(\mathcal{E}(s))) + b),$$

where σ is a sigmoid function, and w and b are learnable scalar parameters for calibration. The model is trained using weak supervision with positive (d, s) pairs from the CNN/DM dataset (Nallapati et al., 2016) and tested on the human-annotated FactCC-D (Kryściński et al., 2019) and QAGS (Wang et al., 2020) benchmarks.

Hypothesis If SATPool retains useful information beyond mean pooling, the resulting DFCScore should improve over a mean-pooled baseline and remain competitive with truncation-based BERTScore and FactCC-Eval.

Implementation Details To evaluate factual consistency, we utilize datasets from FactCC (Kryściński et al., 2019) and QAGS (Wang et al., 2020).

For the FactCC benchmark, we combine the publicly available validation (931 samples) and test (503 samples) splits into a single evaluation set. Similarly, we merge the human-annotated QAGS datasets for the CNN/DailyMail (CNN/DM) (Nallapati et al., 2016) and XSUM (Narayan et al., 2018) test sets to form our second evaluation set. For training our consistency model, we use the document-summary pairs from the CNN/DM training dataset. The model is trained for 100 epochs with a batch size of 64, using an Adam optimizer with a learning rate of 10^{-4} and a Binary Cross-Entropy loss function.

Result Analysis Table 9 shows that **DFCScore-SATPool** consistently improves over DFCScore-Mean and remains competitive with established factuality evaluators. The result indicates that document-wide pooling retains information useful for factuality assessment beyond simple averaging.

F1 Score	FactCC-D	QAGS
BERTScore	0.9274	0.8087
FactCC-Eval	0.9183	0.8075
DFCScore-Mean	0.8667	0.7845
DFCScore-SATPool	0.9275	0.8073

Table 9: Factuality consistency evaluation (F1). DFCScore with SATPool is competitive with established evaluators and improves over its mean-pooling counterpart.

Performance on Extremely Long Documents

As introduced previously, standard factuality consistency evaluators like BERTScore and FactCC-Eval are heavily constrained by the 512-token input limit of their underlying architectures. While these baselines perform reasonably well on average-length texts, their reliance on truncation suggests they should struggle on exceptionally long documents where critical factual context is discarded.

To explicitly test this hypothesis, we evaluated DFCScore-SATPool against the baselines on a much more challenging subset of our evaluation data: the top 10% longest documents from both the FactCC-D and QAGS datasets.

As shown in Table 10, DFCScore-SATPool consistently outperforms the truncation-based baselines on these extended texts. Notably, the performance of both BERTScore and FactCC-Eval drops more sharply on this long-document subset than in the full evaluation reported in Table 9.

Method	FactCC-D	QAGS
BERTScore	0.8923	0.6939
FactCC-Eval	0.8257	0.6067
DFCScore-SATPool	0.8966	0.7050

Table 10: Factuality consistency evaluation (F1 Score) restricted to the top 10% longest documents in each dataset. SATPool outperforms baselines by avoiding truncation and leveraging full-document processing.

F Details of Comparison Methods

F.1 Late Chunking (Günther et al., 2024)

The Late Chunking (Günther et al., 2024) method critiques standard fixed-length chunking, arguing that its arbitrary start and end points can disrupt the document’s semantic context. To address this, Late Chunking proposes segmenting the document along natural semantic boundaries, specifically sentences, which are identified by punctuation.

The process begins by generating an embedding for the entire document while simultaneously recording the token indices corresponding to sentence boundaries. This full sequence of token embeddings is then divided into context-aware chunks using the recorded indices. Finally, each chunk’s embedding is calculated via mean pooling, and these are averaged together to produce the single, final representation for the document.

F.2 MaxP (Dai and Callan, 2019)

The MaxP (Maximum Passage Similarity) method, used in LongRAG (Jiang et al., 2024), is a pooling strategy for determining a document’s relevance to a query. Unlike methods that aggregate information from all chunks, MaxP’s approach is selective, basing the document’s entire relevance on its single most similar passage to the query. The process involves computing a similarity score between the query and every chunk of the document. The final relevance score is then defined as the single maximum score from this set of comparisons. In essence, the document’s value is judged by its most relevant part, assuming that a single strong match is sufficient to make the entire document useful.

F.3 PARADE (Li et al., 2023a)

PARADE (Passage Representation Aggregation for Document Reranking) is a hierarchical method originally designed to aggregate passage representations for document ranking tasks. The method

operates by first encoding individual document chunks independently using a standard BERT encoder to obtain the [CLS] token embedding for each chunk. These chunk-level representations are then fed into a Transformer-based aggregator layer to model the interactions between passages and produce a final document relevance score. In our experiments, we adapt this aggregation strategy to pool chunk representations into a unified document embedding on top of the frozen backbone.

F.4 TrLDC (Dai et al., 2022)

TrLDC (Transformer-based Long Document Classification) proposes a hierarchical architecture specifically optimized for long document classification. Similar to PARADE, it begins by segmenting the document into chunks and encoding each to obtain a [CLS] representation. To better capture the document’s structure, TrLDC adds learnable segment position embeddings to these chunk vectors. The resulting sequence of chunk embeddings is processed by a Transformer encoder to capture inter-chunk dependencies, followed by a Max Pooling operation to derive the final global document representation.

F.5 BERTScore (Zhang et al., 2019)

BERTScore is an evaluation metric designed to judge the quality of a machine-generated sentence by comparing its semantic meaning to a correct, reference sentence (x), rather than just matching keywords. The process begins by using a language model, BERT, to understand the contextual meaning of each word in both the candidate ($\hat{x}$) and reference sentences. Following this, the metric computes a similarity score for every possible pairing of words and employs a simple greedy matching strategy. This strategy works by taking each word from one sentence and pairing it with the single, most similar-meaning word from the other sentence, based on their cosine similarity. These best-match scores are then used to calculate two key metrics. Recall measures if the generated text captured all the ideas from the reference; it does this by looking at each word in the reference sentence and finding its best match in the candidate. Precision, conversely, measures if all the words in the generated text were relevant; it looks at each word in the candidate sentence and finds its best match in the reference. These concepts are formally

Method	Hidden Dim.	# of Param.	Added Params. (% of Backbone, BERT/RoBERTa)	20News		EURLEX	
				BERT	RoBERTa	BERT	RoBERTa
Mean	256	0.20M	0.18%/0.06%	0.6223	0.6315	0.6856	0.6515
Mean-Large	1024	1.59M	1.45%/0.45%	0.6243	0.6289	0.6977	0.7039
SATPool ($k = 4$)	256	0.99M	0.90%/0.28%	0.6621	0.6580	0.7040	0.7107
SATPool ($k = 8$)	256	1.51M	1.38%/0.43%	0.6371	0.6570	0.7124	0.7266

Table 11: Comparison of SATPool against mean pooling baselines with varying parameter counts on the 20News and EURLEX datasets. The Mean-Large model uses an enlarged projection layer to provide a parameter matched baseline. Best F1 scores for each backbone and dataset are in **bold**.

F1 score	20NewsGroups				EURLEX			
	BERT	RoBERTa	Longformer	LongT5	BERT	RoBERTa	Longformer	LongT5
Baseline (CLS)	0.5916	0.5890	0.5798	0.2606	0.5645	0.4975	0.4933	0.4411
Mean	0.6223	0.6315	0.6386	0.5637	0.6856	0.6515	0.6644	0.6921
Late Chunking	0.6166	0.6115	0.6219	0.4909	0.6763	0.6708	0.6731	0.4960
SATPool	$k = 2$	<u>0.6563</u>	0.6657	<u>0.6548</u>	0.6361	0.6881	<u>0.7222</u>	0.7028
	$k = 4$	0.6621	<u>0.6580</u>	0.6634	<u>0.6347</u>	<u>0.7040</u>	0.7107	<u>0.7140</u>
	$k = 8$	0.6371	0.6570	0.6450	0.6195	0.7124	0.7266	0.7232

Table 12: Experimental results for text classification task for 20News and EURLEX datasets. **Bold** indicates the best result in each setting, while underline denotes the second best result.

defined as:

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^\top \hat{x}_j$$

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^\top \hat{x}_j$$

A final F1-score (F_{BERT}) is then calculated as the harmonic mean of Precision and Recall to provide a single, balanced measure.

F.6 FACT-CC (Kryściński et al., 2019)

FactCC is a weakly-supervised, model-based approach for verifying the factual consistency of an abstractive summary against a source document. The core of the FactCC system is a BERT-based classifier fine-tuned on a large, artificially generated dataset because no large-scale, human-annotated corpus for this specific task exists. The training data is created by first extracting sentences from source documents and then applying a series of rule-based textual transformations to generate both positive (factually consistent) and negative (inconsistent) examples. Semantically invariant transformations, such as paraphrasing via back-translation, are used to create consistent claims. Semantically variant transformations are used to create inconsistent claims and include negation, pronoun swapping, and randomly swapping enti-

ties and numbers with others found in the source document. The model architecture takes the full source document and a single summary sentence as input and performs a binary classification (CONSISTENT/INCONSISTENT) using the [CLS] token representation.

G Additional Ablation Studies

G.1 Component-Level Ablations

Table 13 separates the effects of global linear attention and STC. Attention-only is strong, and STC-only does not replace it. Nevertheless, after selecting K , Attn+STC gives the best result in all four classification settings. The benefit is not uniform for every K , so we interpret STC as complementary and task-dependent.

Tables 14 and 15 extend the analysis to retrieval. On NQ, Attn+STC improves over Attn-only for every reported K . On OKVQA, it consistently improves pseudo-relevance recall, while answer recall varies with K .

G.2 Length-Stratified Analysis

Tables 16 and 17 stratify evaluation examples by document length. SATPool improves over the corresponding baseline in every evaluated bucket. As shown in Table 17, the largest NQ retrieval gains occur in the two longest document groups. In contrast,

Method	20News BERT	20News RoBERTa	EURLEX BERT	EURLEX RoBERTa
[CLS]	0.5916	0.5890	0.5645	0.4975
Mean	0.6223	0.6315	0.6856	0.6515
STC-only, $K = 1$	0.6256	0.6312	0.6828	0.6755
STC-only, $K = 2$	0.6220	0.6347	0.6799	0.6962
STC-only, $K = 4$	0.6168	0.6281	0.6916	0.6885
Attn-only	0.6559	0.6550	0.7032	0.7229
Attn+STC, $K = 1$	0.6576	0.6564	0.6927	0.7209
Attn+STC, $K = 2$	0.6563	0.6657	0.6881	0.7222
Attn+STC, $K = 4$	0.6621	0.6580	0.7040	0.7107
Attn+STC, $K = 8$	0.6371	0.6570	0.7124	0.7266

Table 13: Classification component ablation (F1). The best result in each column is bold.

Method	BERT		GTR	
	R@5	R@10	R@5	R@10
STC-only, $K = 1$	0.1632	0.2450	0.5737	0.6634
STC-only, $K = 2$	0.1270	0.1858	0.4523	0.5383
STC-only, $K = 4$	0.0946	0.1338	0.2360	0.2616
Attn-only	0.1798	0.2627	0.6898	0.7640
Attn+STC, $K = 1$	0.1900	0.2672	0.7757	0.8424
Attn+STC, $K = 2$	0.1858	0.2699	0.7384	0.8078
Attn+STC, $K = 4$	0.2164	0.3019	0.7780	0.8451

Table 14: NQ retrieval component ablation.

Method	PRR@5	Ans.@5	PRR@10	Ans.@10
Mean	0.1424	0.4238	0.2015	0.5469
STC-only, $K = 1$	0.1427	0.4215	0.2031	0.5320
STC-only, $K = 2$	0.1444	0.4218	0.2095	0.5408
STC-only, $K = 4$	0.1557	0.4315	0.2266	0.5500
Attn-only	0.1626	0.4379	0.2327	0.5589
Attn+STC, $K = 1$	0.1695	0.4443	0.2388	0.5678
Attn+STC, $K = 2$	0.1829	0.4304	0.2553	0.5542
Attn+STC, $K = 4$	0.1815	0.4277	0.2574	0.5479

Table 15: CLIP-DPR component ablation on OKVQA. PRR denotes pseudo-relevance recall and Ans. denotes answer recall.

Table 16 shows positive but nonmonotonic gains across the 20NewsGroups length groups. Because the two longest 20NewsGroups groups contain only 54 and 42 examples, respectively, their estimates should be interpreted cautiously. The consistently positive NQ answer-recall gains in Table 17 further show that the retrieval improvement carries across document lengths.

G.3 Document-Level versus Chunk-Level Retrieval

We construct a controlled NQ diagnostic using validation questions and a fixed candidate pool of deduplicated gold source documents and randomly sampled distractors. For chunk retrieval, chunks are indexed independently and the parent-document score is the maximum score over its chunks. Ta-

Length	n	Mean F1	SATPool F1	Δ
≤ 512	6,894	0.6146	0.6465	+0.0319
513–1,024	392	0.6872	0.7230	+0.0358
1,025–2,048	150	0.7766	0.7846	+0.0080
2,049–4,096	54	0.7671	0.7883	+0.0212
$> 4,096$	42	0.7549	0.8838	+0.1289

Table 16: Length-stratified 20NewsGroups classification with BERT.

ble 18 shows that SATPool gives the highest parent-document recall with a substantially smaller index than naive chunk-in-isolation retrieval.

This experiment isolates the retrieval-unit question under a fixed candidate pool; it is not a replacement for the full NQ benchmark. It does not compare chunk-in-context representations or establish that document-level retrieval is generally preferable for RAG generation. Those alternatives retain fine-grained retrieval units and are complementary to SATPool.

G.4 Architectural Contribution vs. Parameter Count

SATPool introduces additional trainable parameters through linear attention and the STC projection. To separate architectural effects from model capacity, we compare SATPool with Mean-Large, a mean-pooling baseline whose enlarged classifier matches or exceeds SATPool’s trainable parameter count.

As shown in Table 11, SATPool outperforms the parameter-matched mean-pooling baseline across all evaluated settings while using a comparable number of trainable parameters. The improvement therefore reflects the global aggregation architecture rather than a simple increase in classifier capacity.

Length	n	CLS R@5	SATPool R@5	Δ R@5	CLS Ans.@5	SATPool Ans.@5	Δ Ans.@5
≤ 512	208	0.0994	0.1611	+0.0617	0.3173	0.4327	+0.1154
513–1,024	334	0.1003	0.1411	+0.0408	0.2904	0.3443	+0.0539
1,025–2,048	483	0.0932	0.1534	+0.0602	0.2692	0.3458	+0.0766
2,049–4,096	688	0.0790	0.1793	+0.1003	0.3038	0.4128	+0.1090
$> 4,096$	1,124	0.1021	0.2231	+0.1210	0.3719	0.4875	+0.1156

Table 17: Length-stratified NQ retrieval and generation with BERT. Ans. denotes answer recall.

Method	Index unit	# units	Parent R@1	Parent R@5	Parent R@10
BERT CLS full document	document	1,153	0.1367	0.2883	0.3950
BERT SATPool full document	document	1,153	0.1450	0.3450	0.4667
BERT CLS chunk-512	chunk	10,128	0.0883	0.2617	0.3658
BERT CLS chunk-256	chunk	19,600	0.1083	0.2842	0.3692
BERT CLS chunk-128	chunk	38,621	0.0883	0.3217	0.4367

Table 18: Controlled NQ retrieval-unit diagnostic using 100 validation questions and a candidate pool of 1,153 documents, including 1,000 randomly sampled distractors. Chunk rows use independently encoded chunks and max pooling to obtain parent-document scores.

G.5 Impact of the Number of Spectral Components K

Table 12 supplements the main classification results with additional values of K . Performance varies by dataset: lower spectral resolution is most effective for 20NewsGroups, whereas higher resolution performs best on EURLEX. We therefore treat K as a dataset-dependent hyperparameter rather than expecting a uniform benefit from adding components.

G.6 Impact of Low-Rank Dimension r

Given the contribution of linear attention shown in Figure 6, we further analyze its low-rank dimension r . Figure 8 shows that performance generally improves as r increases, exposing a direct capacity–efficiency trade-off. Because the added parameter count grows linearly as $(2K + 4)rd$, where d is the backbone hidden dimension, we use $r = 128$ in the main experiments as a balanced operating point.

G.7 Joint Analysis of Hyperparameters K and r

To characterize interactions between the number of spectral components (K) and the low-rank dimension (r), we conduct a comprehensive grid search using the RoBERTa backbone on 20NewsGroups. We perform three independent runs with different random initializations for each configuration and report the mean F1 score $\pm$ standard deviation.

The results, presented in Table 19, demonstrate the following characteristics:

Limited Interaction. Table 19 shows a broad stable region rather than a strong dependency in

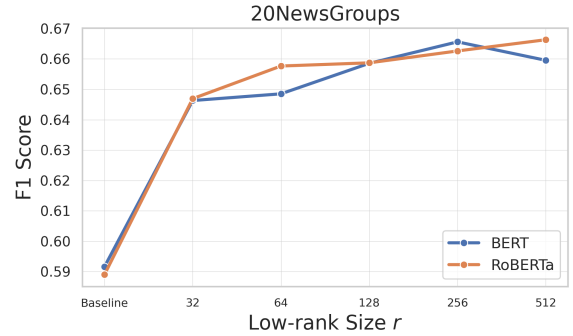


Figure 8: The effect of the low-rank dimension r in the linear attention module on the 20NewsGroups dataset. Performance scales positively with r , illustrating the trade-off between model accuracy and the number of parameters.

which the choice of K dictates a narrow optimal value of r .

Low Sensitivity. Performance varies only modestly across the grid, and the run-to-run deviations are small for most configurations. SATPool is therefore not highly sensitive to either hyperparameter within the evaluated range.

Performance Relative to Mean Pooling. Even the weakest configuration in the grid remains above the strongest RoBERTa mean-pooling result in Table 2. The improvement is therefore preserved without exhaustive joint tuning.

G.8 Sensitivity to Chunking Strategies

To evaluate sensitivity to chunk size and overlap, we compare SATPool with mean pooling on 20NewsGroups using BERT and Longformer. Ta-

$K \backslash r$	32	64	128	256	512
1	0.6470 ± 0.0022	0.6471 ± 0.0039	0.6465 ± 0.0055	0.6538 ± 0.0099	0.6425 ± 0.0067
2	0.6482 ± 0.0085	0.6508 ± 0.0101	0.6481 ± 0.0051	0.6439 ± 0.0042	0.6480 ± 0.0035
4	0.6378 ± 0.0090	0.6502 ± 0.0088	0.6454 ± 0.0062	0.6444 ± 0.0039	0.6381 ± 0.0023
8	0.6451 ± 0.0043	0.6449 ± 0.0055	0.6479 ± 0.0056	0.6399 ± 0.0061	0.6427 ± 0.0066

Table 19: Joint impact of spectral components (K) and low-rank dimension (r) on F1 Score (Mean $\pm$ Std) using RoBERTa on 20NewsGroups.

Backbone	Chunk Size	Stride = 0.5 (50% Overlap)		Stride = 1.0 (No Overlap)	
		Mean	SATPool	Mean	SATPool
BERT	128	0.6020	0.6394	0.6111	0.6224
	256	0.6150	0.6450	0.6142	0.6317
	512	0.6228	0.6387	0.6229	0.6316
Longformer	1024	0.6312	0.6481	0.6306	0.6618
	2048	0.6274	0.6445	0.6301	0.6570
	4096	0.6309	0.6356	0.6284	0.6573

Table 20: Impact of Chunk Size and Stride (Overlap) on F1 Score. SATPool consistently outperforms Mean Pooling across varying chunk sizes and overlap strategies.

ble 20 summarizes the results.

Consistent Advantage. SATPool outperforms mean pooling for every tested backbone, chunk size, and stride, showing that its benefit is not tied to a single preprocessing configuration.

Impact of Stride and Overlap.

- *Short-context encoder (BERT):* Overlapping chunks are generally preferable, suggesting that repeated boundary context helps preserve local evidence before global aggregation.
- *Long-context encoder (Longformer):* Non-overlapping chunks perform better, indicating that the encoder’s wider native context reduces the need for overlap.

Thus, the preferred stride depends on the backbone’s native context range, while SATPool’s advantage over mean pooling remains stable across both regimes.

G.9 Justification of Fixed Spectral Bases

To justify our design choice of fixed sinusoidal bases over learnable bases or Principal Component Analysis (PCA), we compare SATPool against two rigorous baselines:

1. **Learned Bases (Linear Interpolation):** To handle variable lengths robustly, we implement learned positional weights $W_{pos} \in \mathbb{R}^{N_{basis} \times 2048}$ and apply linear interpolation during the forward pass to match the input document length L . This isolates structural

overfitting from simple length mismatch issues.

2. **PCA Bases (Dynamic SVD):** We compute Principal Components on-the-fly for each document instance, projecting tokens onto the top N_{basis} eigenvectors of the covariance matrix. This serves as an ideal “Bag-of-Words” baseline that captures dominant semantic topics but discards sequential structure.

We choose fixed sinusoidal bases for theoretical reasons to provide a harmonic inductive bias, serving as a natural structural prior for sequential text. These bases provide a length-agnostic prior that captures the narrative flow across variable scales.

Classification Tasks As shown in Table 21, learned bases often overfit to absolute token positions, resulting in lower performance than the spectral method. This pattern holds across both the 20NewsGroups and EURLEX datasets, regardless of whether a short-context (BERT) or long-context (Longformer) backbone is used.

Retrieval Tasks We further evaluate these methods on the Natural Questions (NQ) dataset under two settings: Standard and Permuted (shuffling document chunks to destroy narrative structure).

Analysis. Table 22 shows that learned bases degrade most after paragraph-order permutation, consistent with greater reliance on absolute positions. PCA is less sensitive to permutation but remains weaker overall. Fixed spectral bases provide the

Method	SATPool	Learned Bases
BERT-20News	0.637	0.628
Longformer-20News	0.645	0.625
BERT-EURLEX	0.712	0.704
Longformer-EURLEX	0.723	0.714

Table 21: Comparison of fixed spectral bases versus learned bases on classification tasks (F1 Score). Fixed bases consistently prevent positional overfitting and yield superior performance.

strongest balance of standard-set performance and permutation robustness, supporting their role as a length-agnostic structural prior.

H Computational Efficiency Analysis

H.1 Detailed Runtime and FLOPs Analysis

We report the full computational benchmarks for SATPool against baselines in Table 23. The experiment involved processing a document of length 19,502 tokens (approx. 91.5k characters) using a BERT backbone.

Result Analysis.

- **Near-mean-pooling runtime:** SATPool adds little end-to-end latency because the chunk encoder dominates computation.
- **Faster than Late Chunking:** SATPool avoids the additional contextual chunk-construction cost incurred by Late Chunking.
- **Faster than the long-context baseline:** BERT with SATPool is more efficient than the evaluated Longformer configuration, supporting the use of short-context encoders with lightweight global pooling.

H.2 Hyperparameter Impact

To assess the cost of our hyperparameters, we measure performance across a grid of K (spectral components) and r (low-rank dimension) as shown in Table 24.

- **K has negligible FLOP impact:** At fixed r , the measured FLOPs remain effectively unchanged as the number of spectral components increases.
- **r controls the trade-off:** Computational cost grows with the low-rank dimension, making r the primary efficiency knob.

H.3 Memory Benchmark

We measure the **Peak CUDA Memory** allocated during the forward pass of a single long document (19,502 tokens) on an NVIDIA RTX A5000 (24GB). We report the “Total System Memory,” which includes both the Retriever (Encoder + Pooling) and the Generator (Qwen2.5-3B), to demonstrate the real-world impact as shown in Table 25.

Table 25 shows that SATPool adds only modest memory overhead relative to the full retrieval-and-generation system. The generator remains the dominant memory component.

I Theoretical Analysis

I.1 Comparison with Classical Spectral Analysis.

STC draws inspiration from classical spectral analysis but uses a small fixed set of sine and cosine bases as positional features. The coefficient vectors are deterministic projections of the learned, content-dependent sequence $\mathbf{H}^{\text{attn}}$; they are not independent learned parameters. Because the layer is differentiable, the upstream attention projections and downstream output projection can learn task-specific representations and combinations of these fixed-basis features.

I.2 Dense Local vs. Sparse Global Attention

A fundamental question arises regarding the theoretical upper bound of pooling methods compared to dedicated long-context models: Can a model that encodes chunks in isolation (like SATPool) ever rival a model that accesses global context during encoding (like Longformer)?

Our results suggest that the answer lies in the trade-off between attention density and context length.

- **Sparse Global Attention:** Architectures like Longformer (Beltagy et al., 2020) rely on sparse attention patterns (e.g., sliding windows) to manage quadratic complexity. While they access global context, they sacrifice dense token-to-token connectivity, potentially missing fine-grained dependencies.
- **Dense Local + Global Pooling:** SATPool utilizes standard encoders (e.g., BERT) that preserve dense attention within each chunk. Stage 1 then models interactions among the

Standard Test Set	R@1	R@5	Ans. Recall
SATPool w/ Learned	0.2356±0.0085	0.4629±0.0101	0.2274±0.0076
SATPool w/ PCA	0.2386±0.0082	0.5009±0.0101	0.2358±0.0082
SATPool (Spectral)	0.2371±0.0083	0.5036±0.0096	0.2393±0.0081
Permuted Test Set	R@1	R@5	Ans. Recall
SATPool w/ Learned	0.1775±0.0072	0.3970±0.0098	0.2006±0.0074
SATPool w/ PCA	0.2120±0.0079	0.4758±0.0096	0.2203±0.0079
SATPool (Spectral)	0.2204±0.0078	0.4821±0.0096	0.2274±0.0079

Table 22: RAG Performance on NQ (Standard vs. Permuted Test Set). Fixed spectral bases demonstrate superior robustness compared to Learned or PCA bases.

Method	Time (ms)	FLOPs	Relative Speed
<i>Baselines</i>			
Truncation (CLS)	447.37	6.05e+08	3.0x
MaxP	989.76	4.66e+10	1.3x
Mean Pooling	1333.84	4.65e+10	1.0x (Baseline)
<i>Advanced Aggregators</i>			
PARADE	1024.73	4.67e+10	1.3x
TrLDC	1020.18	4.67e+10	1.3x
Late Chunking	1657.78	4.65e+10	0.8x
Longformer (CLS)	2356.94	4.59e+10	0.57x
SATPool (Ours)	1347.72	5.26e+10	~1.0x

Table 23: Wall-clock speed and computational cost with a BERT backbone. SATPool operates near the mean-pooling throughput in this benchmark.

Method	Time (ms)	FLOPs
<i>r = 64</i>		
K = 2	1344.32	4.95e+10
K = 4	1315.75	4.95e+10
K = 8	1328.66	4.95e+10
<i>r = 128</i>		
K = 2	1322.37	5.26e+10
K = 4	1405.63	5.26e+10
K = 8	1449.94	5.26e+10
<i>r = 256</i>		
K = 2	1460.34	5.93e+10
K = 4	1377.50	5.93e+10
K = 8	1446.98	5.93e+10

Table 24: The impact of r and k on computational cost

assembled local token representations with linear attention, but cannot recover information absent from those representations.

Empirically, our results on EURLEX (Table 2) demonstrate that preserving high-fidelity local representations combined with global rela-

tional aggregation (SATPool) often outperforms the sparse-attention approximations used by native long-context architectures. These results suggest that, in the evaluated settings, preserving dense local representations and adding lightweight global aggregation can be competitive with native sparse-attention long-context encoders. Because the compared models differ in architecture and pretraining, the performance gap cannot be attributed solely to their attention patterns.

I.3 Inductive Bias and Semantic Trajectories

I.3.1 The Necessity of Inductive Bias.

Standard mean pooling is a permutation-invariant operation; it creates a many-to-one mapping where structurally distinct sequences (e.g., swapping the introduction and conclusion) produce identical vectors. Consequently, structural signals are lost before reaching the classifier, regardless of downstream training. SATPool addresses this by introducing a necessary inductive bias: it modulates

Method	Total Sys Mem (MB)	Overhead vs. CLS (MB)	% of CLS Baseline
Truncation (CLS)	11,775.56	-	100.00%
MaxP	11,800.31	+24.75	100.21%
PARADE	11,805.28	+29.72	100.25%
TrLDC	11,805.43	+29.87	100.25%
Mean Pooling	11,856.97	+81.41	100.69%
Late Chunking	11,858.99	+83.43	100.71%
SATPool (Ours)	11,925.24	+149.68	101.27%

Table 25: Peak Memory Usage (Single Document Inference). SATPool introduces negligible overhead relative to the full system memory.

embeddings with spectral bases *before* aggregation. This renders the operation permutation-sensitive, explicitly encoding global positions into the feature magnitude and preserving the document’s narrative flow.

I.3.2 Modeling Semantic Trajectories.

We view the ordered token-representation sequence as a multidimensional discrete signal rather than a bag of meanings. Lower-frequency bases summarize broad positional variation, whereas higher-frequency bases represent more rapidly varying patterns. The output projection learns task-useful combinations of these deterministic coefficients; we do not assume that a particular frequency always corresponds to a fixed discourse role.

I.4 Pool-then-Attend vs. Attend-then-Pool.

The underperformance of hierarchical methods like TrLDC and PARADE in our frozen-encoder setup highlights a critical architectural distinction. These methods follow a “*Pool-then-Attend*” paradigm: they first compress each chunk into a single [CLS] vector before aggregation. When using frozen encoders not fine-tuned for the specific downstream task, the generic [CLS] token often captures suboptimal representations of local chunks, creating an early information bottleneck.

In contrast, SATPool follows an “*Attend-then-Pool*” paradigm. Stage 1 operates on the assembled sequence of locally encoded tokens before compression occurs in Stage 2. This delays the document-level information bottleneck relative to chunk-first pooling, while retaining the limitations of the initial local encoding stage.

J Full Fine-Tuning

The “Attend-then-Pool” paradigm is a fundamental architectural property of SATPool that preserves

high-fidelity token information before compression. As discussed in Appendix I.4, this contrasts with hierarchical methods that create early information bottlenecks by compressing chunks into single vectors before aggregation.

To determine whether the pattern is specific to frozen encoders, we evaluate full fine-tuning on 20NewsGroups with a BERT backbone and allow all parameters to update during training.

Table 26 shows that SATPool outperforms the compared hierarchical methods under full fine-tuning. The attend-then-pool design therefore remains useful beyond the frozen-encoder setting.

Method (Full Fine-Tune BERT)	F1 Score
TrLDC	0.709
PARADE	0.715
SATPool (Ours)	0.719

Table 26: Full-fine-tuning results on 20NewsGroups with BERT (F1).

K Results on Native Long-Context Models

To further evaluate the versatility of our approach, we demonstrate how SATPool integrates with native long-context embedding models, specifically NV-Embed-v2 (Lee et al., 2025) and GTE-Qwen2-7B (Li et al., 2023b). While these models provide inherently strong long-context baselines, SATPool functions as a plug-and-play aggregation module intended to enhance rather than compete with their representations. For this experiment, we apply SATPool directly on top of these models while keeping their respective encoders frozen.

As detailed in Table 27, adding SATPool improves classification performance for both tested

long-context encoders and datasets. Many documents in Table 1 still exceed the input capacity of extended-context models. SATPool can aggregate their chunk-encoded representations without modifying the frozen encoder, although the SATPool module and task head still require task-specific training.

Method	20NewsGroups	EURLEX
GTE-Qwen2	0.7424	0.7289
+ SATPool	0.7550	0.7430
NV-Embed-v2	0.7528	0.7128
+ SATPool	0.7645	0.7141

Table 27: Performance (F1 Score) of native long-context models with and without SATPool integration. SATPool consistently enhances the representational quality of frozen long-context encoders.

L Statistical Robustness

We use bootstrap resampling to examine whether the observed comparative results are stable under changes in the evaluated sample.

For each of the retrieval-augmented generation (RAG) and multimodal RAG (MM-RAG) experiments, we performed 1,000 bootstrap resamples of the full evaluation set with replacement. The results reported in Table 28 and Table 29 represent the mean and the standard deviation ($\pm$) across these 1,000 resamples.

The small bootstrap standard deviations indicate stable resampled performance. Across the text and multimodal datasets, SATPool preserves the same overall comparative pattern observed in the primary evaluation.

M AI Assistant Acknowledgement

We use LLMs for the purposes of code debugging and polishing the manuscript’s writing. We reviewed and edited all content as needed and take full responsibility for the final content of this publication.

Dataset: Natural Questions (NQ)				
Method	BERT		GTR	
	R@5	Answer Recall	R@5	Answer Recall
CLS	0.2038±0.0075	0.1984±0.0074	0.8099±0.0075	0.3511±0.0086
Mean	0.4595±0.0094	0.2601±0.0081	0.7874±0.0083	0.3528±0.0088
MaxP	0.3145±0.0092	0.2266±0.0079	0.6359±0.0093	0.3165±0.0088
LC	0.4688±0.0097	0.2605±0.0081	0.6418±0.0092	0.3014±0.0085
PARADE	0.3425±0.0092	0.2281±0.0079	0.7110±0.0084	0.3275±0.0085
TrLDC	0.3060±0.0094	0.2323±0.0078	0.3380±0.0088	0.2330±0.0078
SATPool	0.5622±0.0095	0.2940±0.0085	0.8175±0.0073	0.3676±0.0088

Dataset: HotpotQA				
Method	BERT		GTR	
	R@5	Answer Recall	R@5	Answer Recall
CLS	0.0628±0.0033	0.2231±0.0055	0.5926±0.0066	0.3309±0.0063
Mean	0.2148±0.0053	0.2611±0.0059	0.5135±0.0067	0.3139±0.0061
MaxP	0.1340±0.0045	0.2455±0.0056	0.1340±0.0045	0.2455±0.0056
LC	0.2345±0.0059	0.2620±0.0058	0.4228±0.0066	0.2971±0.0063
PARADE	0.1019±0.0041	0.2261±0.0056	0.2790±0.0060	0.2582±0.0058
TrLDC	0.0914±0.0039	0.2239±0.0056	0.0271±0.0022	0.2143±0.0054
SATPool	0.2466±0.0056	0.2693±0.0059	0.5744±0.0066	0.3221±0.0064

Table 28: Statistical robustness results for RAG on the NQ and HotpotQA datasets (Mean ± Std across 1,000 bootstrap resamples). **Bold** indicates the best result in each setting, while underline denotes the second best.

Dataset: OKVQA				
Method	CLIP-DPR		RAVQA	
	PRR@5	Answer Recall	PRR@5	Answer Recall
CLS	0.1371±0.0050	0.4065±0.0075	0.0551±0.0032	0.3983±0.0070
Mean	0.1424±0.0050	0.4048±0.0069	0.0817±0.0039	0.4045±0.0071
MaxP	0.0745±0.0036	0.4101±0.0074	0.0676±0.0033	0.4054±0.0070
LC	0.1321±0.0048	0.4022±0.0069	0.0727±0.0035	0.4009±0.0070
PARADE	0.1249±0.0045	0.3965±0.0070	0.0855±0.0039	0.4166±0.0069
TrLDC	0.1421±0.0049	0.4100±0.0069	0.0864±0.0040	0.4045±0.0070
SATPool	0.1564±0.0051	0.4148±0.0071	0.0964±0.0041	0.4115±0.0071

Dataset: EVQA				
Method	CLIP-DPR		RAVQA	
	R@5	Answer Recall	R@5	Answer Recall
CLS	0.2588±0.0105	0.2496±0.0105	0.0535±0.0054	0.1117±0.0072
Mean	0.2338±0.0101	0.2367±0.0103	0.0843±0.0067	0.1378±0.0084
MaxP	0.1730±0.0089	0.2005±0.0097	0.0445±0.0049	0.1168±0.0076
LC	0.2385±0.0107	0.2350±0.0104	0.0767±0.0066	0.1231±0.0077
PARADE	0.2407±0.0105	0.2368±0.0099	0.0908±0.0072	0.1383±0.0080
TrLDC	0.2200±0.0101	0.2293±0.0100	0.0797±0.0063	0.1244±0.0073
SATPool	0.2585±0.0107	0.2592±0.0106	0.0900±0.0071	0.1444±0.0082

Table 29: Statistical robustness results for MultiModal-RAG on the OKVQA and EVQA datasets (Mean ± Std across 1,000 bootstrap resamples). **Bold** indicates the best result in each setting, while underline denotes the second best.