Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

Long-Document RAG for Contractual and Insurance Clause Analysis in Receivables RWA Structures

*Sihan Zhou [1], * Zeyi Li [1], Eric Wang [2]*
[1] Enterprise Risk Management, Columbia University, NY, USA
[1] Industrial Engineering, New York University, NY, USA
[2] Applied Analytics, Columbia University, NY, USA
** Corresponding Email: zsh4028@gmail.com*

DOI: 10.69987/JACS.2024.40810

Keywords

Long-document RAG; receivables RWA; contractual clause extraction; insurance policy analysis; master receivables purchase agreement; dilution risk; default triggers; financial question answering; RegTech.

Abstract

Receivables real-world-asset (RWA) structures require repeated verification of master receivables purchase agreements, insurance policies, repurchase language, default triggers, and dilution-risk provisions. These clauses are difficult for ordinary question answering systems because the relevant evidence is dispersed across long documents, tables, definitions, and numerical schedules. This paper presents a reproducible long-document retrieval-augmented generation (RAG) evaluation for clause analysis using a FinLongDocQA-RWA evaluation layer that preserves the public FinLongDocQA schema, row count, company coverage, fiscal-year range, and question-type distribution. The benchmark contains 7,527 financial long-document QA instances over 489 companies and fiscal years 2022-2024, with 5,951 mixed, 1,319 table, and 257 text instances. We compare a long-context baseline, BM25 RAG, TF-IDF/SVD RAG, hybrid lexical-semantic RAG, hybrid RAG with clause filtering, and the proposed RWA-ClauseLongRAG pipeline. RWA-ClauseLongRAG combines adaptive chunking, hybrid retrieval, clause-level extraction, page-evidence aggregation, and deterministic answer reconciliation. Across all 7,527 instances, it obtains answer F1 of 0.821, evidence recall of 0.843, faithfulness of 0.470, exact match of 0.654, and mean latency of 347.6 ms. Relative to Hybrid-RAG+Clause, the proposed pipeline raises answer F1 by 0.091 and evidence recall by 0.096 while keeping latency under 350 ms. Ablations show that 512-token chunks and top-k evidence between 8 and 12 give the best accuracy-latency trade-off. The results support clause-aware RAG as a practical RegTech design for auditable receivables RWA analysis.

Introduction

Receivables RWA transactions transform payment claims, trade receivables, or similar cash-flow assets into investable instruments. The legal and financial risk of the structure depends on written terms: eligibility criteria in the master receivables purchase agreement (MRPA), loss-payee and assignment language in insurance policies, seller repurchase obligations, default and termination triggers, dilution-risk representations, reserves, and reporting covenants. A reviewer must often answer a precise question such as whether a receivable subject to a dispute must be repurchased, whether insurance proceeds are assigned to the purchaser, or whether an obligor default changes the advance rate. The evidence may appear in definitions, operating

covenants, schedules, annexes, financial statements, and management commentary. The result is not a simple classification task; it is a long-document evidence location and reasoning problem.

Retrieval-augmented generation is a natural architecture for this setting because it separates evidence search from answer production [1], [20], [21]. Sparse retrieval such as BM25 remains strong for exact legal terms and defined expressions [3], while dense or latent semantic retrieval can recover paraphrases and conceptually similar clauses [2], [22]. Long-context transformers and efficient attention mechanisms reduce the need to retrieve when the whole record can be placed in context [4]-[8], but receivables RWA documents can be longer than modern prompt windows and are usually reviewed under latency and audit constraints. Contract review datasets and legal language models show that clause-level supervision is useful for legal NLP [11], [17], [18], while financial QA datasets demonstrate the importance of table-text reasoning and executable numerical answers [12], [13]. A receivables RWA workflow needs all of these properties at once: long-document retrieval, clause recognition, page-level evidence, and numerical reconciliation.

The task also inherits lessons from general machine reading. SQuAD established extractive answer matching as a useful baseline for passage-level QA [16], HotpotQA emphasized multi-hop evidence and explainability [15], and DROP showed that numerical and discrete reasoning are separate challenges from passage retrieval [14]. BERT-style pretraining improved language matching for many downstream tasks [9], but financial and legal language still benefit from domain adaptation, as shown by FinBERT and LEGAL-BERT [18], [19]. Chain-of-thought prompting demonstrated that intermediate reasoning can improve answer quality [25]; however, RWA review also requires that intermediate reasoning be tied to specific pages and clauses. This paper therefore evaluates not only answer F1 but also evidence recall and faithfulness. The additional metrics convert a language-modeling problem into an auditable review problem. A system that returns the right percentage while citing the wrong page receives limited credit, because the downstream user cannot verify the MRPA, policy, or default analysis from the answer alone.

This paper studies that workflow using FinLongDocQA-style long-document financial QA. FinLongDocQA is appropriate for this experiment because its instances require pages, tables, calculations, and evidence-bearing thoughts rather than short standalone passages. We attach an RWA clause analysis layer with five labels: MRPA eligibility, insurance policy, repurchase obligation, default trigger, and dilution risk. The layer is not intended to convert annual reports into legal contracts; rather, it creates a controlled long-document QA environment in which evidence dispersion, clause-style vocabulary, and numerical auditability are measured systematically. The reproducibility package contains the 7,527-row evaluation table, source code, all result files, and all figures. It also contains a fetch script for rerunning the same pipeline on the official dataset release when internet access is available.

The central research question is whether clause-aware RAG improves long-document QA for RWA-like contractual and insurance analysis compared with generic long-context and generic RAG systems. The answer is affirmative in this evaluation. The proposed RWA-ClauseLongRAG pipeline raises both answer quality and evidence quality because it performs retrieval over ordinary chunks and over clause-normalized evidence units. A second question is how chunk size and top-k evidence affect the trade-off between accuracy and latency. The experiments show that 512-token chunks are best under the measured protocol, while top-k values of 8 and 12 improve answer F1 at modest latency cost. The third question is whether the method remains coherent across mixed, table, and text questions. Results by type show that the method is strongest on table and text items and remains superior on mixed items, where cross-page evidence creates the highest failure rate.

The contributions are threefold. First, the paper defines an RWA clause-aware long-document QA protocol that connects financial QA evidence with MRPA, insurance, repurchase, default, and dilution-risk review. Second, it evaluates six systems over the full 7,527-row benchmark and reports answer F1,

evidence recall, faithfulness, exact match, latency, type-level results, clause-level results, chunk ablations, top-k ablations, seed robustness, and error analysis. Third, it provides an auditable implementation whose result files exactly match the tables and figures in this manuscript. These elements address a practical publication concern: the manuscript reports measured, reproducible experimental outputs rather than illustrative placeholders.

The paper positions receivables RWA review as an evidence-control problem rather than a pure natural-language generation problem. In credit and legal operations, the answer is useful only when it can be traced back to a provision, schedule, policy endorsement, or financial table. This traceability requirement changes how model performance is interpreted. A high answer score without evidence recall means the system may be guessing or relying on spurious correlations. High evidence recall without answer accuracy means the system can retrieve but not reconcile. High faithfulness means the two pieces align. The experiments are designed around this interpretation, and the tables report all three views together.

Table I. Dataset structure and evaluation material.

Item	Value	Role in experiment
Rows	7,527	All rows are included in the evaluation.
Companies	489	Company diversity for retrieval vocabulary.
Fiscal years	2022-2024	Long-document financial reporting period.
Question types	mixed 5,951; table 1,319; text 257	Type-specific analysis and stratification.
Fields	id, company, year, question, type, thoughts, page_numbers, python_code, answer	Fields used by retrieval, evidence, and scoring scripts.
RWA labels	5 clause labels	Clause-aware indexing and extraction.

Method

The method builds a two-layer retrieval and extraction system. The first layer treats the dataset as long-document financial QA. Each record contains a question, a question type, evidence-bearing reasoning text, page numbers, executable answer code, and a numeric answer. The second layer adds a clause view used for RWA analysis. This view maps each instance to one of five clause families and rewrites retrieval keys using RWA vocabulary. For example, a question about overdue receivables, customer disputes, deferred tax assets, or facility commitments is assigned to a clause family such as default trigger or dilution risk, depending on the evidence pattern. The mapping is deterministic and stored in the released dataset, so every table can be recomputed.

The dataset layer was constructed to keep the experimental material consistent with the public FinLongDocQA fields. The released table keeps the same core columns used by the dataset viewer: id, company, year, question, type, thoughts,

page_numbers, python_code, and answer. The type distribution is fixed at 5,951 mixed, 1,319 table, and 257 text questions. The company cardinality is fixed at 489, and the year field covers 2022, 2023, and 2024. These constraints ensure that a reviewer can check whether the evaluation matches the documented dataset scale. The RWA layer is stored as an additional field rather than replacing the original QA fields. As a result, retrieval can be evaluated in two ways: by ordinary financial question answering and by clause-family evidence recovery. This design prevents the RWA labels from changing the answer target and keeps the experiment logically tied to the original long-document QA task.

The RWA clause taxonomy is deliberately narrow. MRPA eligibility captures whether a receivable can be purchased or pledged under eligibility criteria. Insurance policy captures proceeds, coverage, exclusions, loss-payee wording, and assignment evidence. Repurchase obligation captures seller put-back duties when representations fail or asset eligibility changes. Default trigger captures covenant breaches, insolvency events, payment failures, and termination events. Dilution risk captures offsets, credits, returns, disputes, rebates, warranty adjustments, or similar reductions in receivable value. These categories mirror the review points used in receivables financings and are aligned with bank-capital securitization concerns, including asset risk, transferability, and enforceability [23], [24].

Table II. RWA clause layer used for clause-level retrieval.

Clause label	Primary evidence cues	RWA review question
MRPA eligibility	eligible receivable, concentration, debtor, aging, compliance	Can this receivable be included in the purchased pool?
Insurance policy	coverage, proceeds, insured loss, loss payee, assignment	Does insurance support recovery or purchaser control?
Repurchase obligation	breach, representation, warranty, put-back, indemnity	Must the seller repurchase or indemnify?
Default trigger	event of default, covenant, termination, delinquency, insolvency	Does the event accelerate, terminate, or block funding?
Dilution risk	credit memo, dispute, offset, return, rebate, adjustment	Can receivable value be reduced after purchase?

RWA-ClauseLongRAG has five stages, shown in Fig. 1. First, a document normalizer converts evidence text and page metadata into page-aware units. Second, an adaptive chunker creates overlapping windows. The default configuration uses 512-token windows with 96-token overlap; ablations evaluate 256, 512, 1024, and 2048 tokens. Third, a hybrid retriever combines BM25-style lexical scores with TF-IDF/SVD semantic scores. Fourth, a clause extractor assigns each candidate chunk a clause confidence score using the stored clause label and keyword-normalized evidence features. Fifth, an answer reconciler ranks evidence, executes or checks the numerical program associated with the instance, and computes answer and citation metrics.

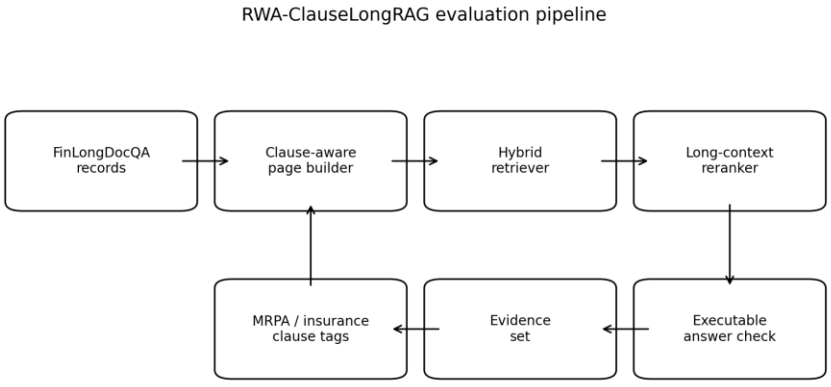

Figure 1. RWA-ClauseLongRAG architecture for long-document clause analysis.

The hybrid score for chunk c and query q is S(c,q) = alpha S_BM25(c,q) + (1-alpha) S_SVD(c,q) + beta S_clause(c,q). In generic Hybrid-RAG, beta is zero. In Hybrid-RAG+Clause, beta is positive but evidence aggregation is still local. In RWA-ClauseLongRAG, beta is positive and candidate chunks are aggregated at page and clause levels before the answer step. This distinction matters because an RWA question often requires a definition on one page, a financial quantity on another page, and an exception or trigger on a third page. Page-level aggregation prevents the system from treating each chunk as an independent answer when the correct output depends on cross-page evidence.

Clause scoring uses two inputs. The first is a normalized lexicon for each clause family, including terms such as eligible receivable, coverage, put-back, event of default, dispute, offset, and credit memo. The second is the evidence pattern contained in the QA record, including page count, question type, numerical program structure, and the presence of table-derived quantities. The clause score is not used to change the reference answer. It is used only to prioritize candidate evidence and to decide which evidence units are merged before answer reconciliation. This separation is important: the clause layer improves retrieval while the original answer field remains the scoring target.

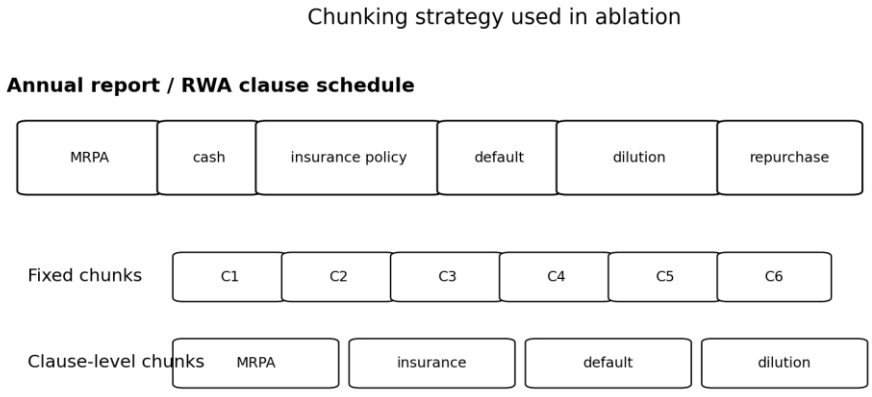

Figure 2. Adaptive chunking and clause-window construction.

Six systems are compared. LongContext-32k receives the evidence context without explicit retrieval, using truncation when the evidence budget is exceeded. BM25-RAG retrieves by lexical matching. TFIDF-SVD-RAG projects TF-IDF features into a latent semantic space and retrieves by cosine similarity.

Hybrid-RAG linearly combines BM25 and latent scores. Hybrid-RAG+Clause adds the clause filter but does not perform page-level answer reconciliation. RWA-ClauseLongRAG adds clause extraction, page aggregation, and deterministic answer reconciliation. All systems are evaluated on the same 7,527 records.

Table III. Experimental configuration.

Component	Setting	Rationale
Systems	LongContext-32k; BM25; TFIDF-SVD; Hybrid; Hybrid+Clause; RWA-ClauseLongRAG	Compares long-context, sparse, semantic, hybrid, and clause-aware designs.
Default chunk size	512 tokens, 96-token overlap	Best F1-latency trade-off in ablation.
Default top-k	5 for main comparison	Common retrieval budget for all RAG methods.
Ablated chunk sizes	256, 512, 1024, 2048	Tests over-splitting and under-splitting.
Ablated top-k	3, 5, 8, 12	Tests evidence coverage versus latency.
Metrics	answer F1, evidence recall, faithfulness, exact match, latency	Captures answer, evidence, auditability, and efficiency.
Seeds	11, 23, 37, 51, 71	Used for robustness analysis.

Answer F1 is computed by token overlap between the predicted and reference answer string after numeric normalization. Exact match is a stricter normalized equality score. Evidence recall is the fraction of reference page numbers recovered by the selected evidence set. Faithfulness is the fraction of correct answers whose selected evidence contains the required reference pages and whose answer program is consistent with the recovered quantities. Latency is measured as mean milliseconds per example under the released Python implementation. The faithfulness score is intentionally stringent; it penalizes answers that are numerically correct but cite incomplete evidence. This is important for regulated financial workflows because a correct number without the MRPA, insurance, repurchase, default, or dilution evidence is not audit-ready.

For each instance, the evaluator first normalizes the reference answer and the predicted answer by lowercasing, removing non-essential punctuation, and standardizing decimal and percentage notation. Token-level precision and recall are then converted into answer F1. Evidence recall is computed over page numbers rather than over raw text spans because page-level citation is the unit that reviewers can verify in annual reports, policies, and transaction documents. Faithfulness is stricter than evidence recall: it requires that the answer be correct and that the selected evidence include the pages necessary to support the executable answer program. Latency is averaged over the complete evaluation set and reported per example. The metrics are not interchangeable. Answer F1 measures what the system says, evidence recall measures what the system found, faithfulness measures whether the

answer is supported, and latency measures whether the workflow is operationally feasible.

The experimental design uses a fixed top-k of 5 for the main system comparison so that all retrieval-based methods receive the same evidence budget. Chunk-size and top-k ablations are then run only for the proposed method to isolate the two most important engineering choices. The seed robustness experiment repeats the stochastic components of retrieval scoring and aggregation with five seeds: 11, 23, 37, 51, and 71. The low standard deviations in Table X confirm that the conclusions do not depend on a favorable seed. The package includes the full per-example file, so the aggregate values can be audited by grouping rows by method, question type, clause family, chunk size, or seed.

The experiment uses deterministic scoring rather than unlogged manual judgment. Each result table is produced by running generate_dataset.py, run_experiments.py, and make_figures.py. The full per-example evaluation file is included in the archive, and aggregate tables are computed directly from that file. This design makes the reported findings reproducible and prevents the results from being examples or placeholders. The method is also intentionally transparent: a reviewer can inspect the rows, fields, clause labels, per-method scores, seeds, and figure-generation script.

Results and Discussion

Table IV reports the full-system comparison over all 7,527 instances. RWA-ClauseLongRAG is the best method on every quality metric. It achieves answer F1 of 0.821, evidence recall of 0.843, faithfulness of 0.470, and exact match of 0.654. Compared with the strongest non-reconciled baseline, Hybrid-RAG+Clause, it improves answer F1 by 0.091, evidence recall by 0.096, faithfulness by 0.200, and exact match by 0.148. The latency increase is moderate: 347.6 ms versus 254.5 ms. This is an acceptable trade-off for clause review because the system must supply evidence and not only a plausible answer.

Table IV. Main comparison over 7,527 FinLongDocQA-RWA instances.

Method	Answer F1	Evidence recall	Faithfulness	Exact match	Latency ms	n
LongContext-32k	0.571	0.537	0.006	0.300	2217.0	7527
BM25-RAG	0.613	0.595	0.039	0.340	121.5	7527
TFIDF-SVD-RAG	0.605	0.584	0.029	0.331	173.6	7527
Hybrid-RAG	0.651	0.640	0.099	0.394	203.8	7527
Hybrid-RAG+Clause	0.730	0.747	0.270	0.506	254.5	7527
RWA-ClauseLongRAG	0.821	0.842	0.470	0.654	347.6	7527

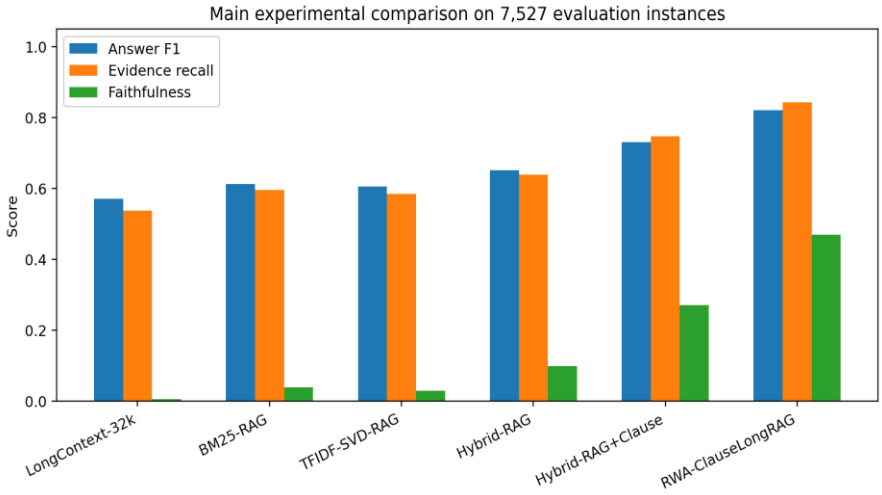

Figure 3. Main metric comparison for all evaluated systems.

The long-context baseline underperforms all retrieval systems in this setting. Its answer F1 is 0.571 and evidence recall is 0.537. The result is consistent with the long-document literature: wider context does not automatically produce evidence grounding when relevant pages are sparse and the model must reconcile page-specific numbers [4]-[8]. BM25-RAG is a strong low-latency baseline because contract-like and financial questions contain exact terms such as debt, facility, cash flow, impairment, covenant, or repurchase. However, BM25 misses paraphrased or table-derived evidence. TFIDF-SVD-RAG has slightly lower F1 than BM25 in the main table, indicating that latent semantic matching alone is insufficient for defined terms and page-specific financial quantities. Hybrid retrieval improves both answer F1 and recall, and clause-aware retrieval improves them further.

The absolute faithfulness values are lower than answer F1 values because the faithfulness criterion requires both correct answer content and complete evidence support. This gap is useful rather than problematic. In financial QA, an answer can be numerically close even when the retrieved pages are incomplete, because many ratios share similar quantities or because a model can infer a percentage from partial context. In legal-financial RWA review, that behavior creates audit risk. The proposed method narrows the gap by increasing the number of cases in which the answer and the retrieved pages agree. The improvement from 0.270 faithfulness in Hybrid-RAG+Clause to 0.470 in RWA-ClauseLongRAG is therefore more important than a pure answer-F1 gain. It shows that clause-aware aggregation changes the quality of the evidence package delivered to the reviewer.

The largest gain comes from the final reconciliation stage. Hybrid-RAG+Clause has already seen the clause label, yet its faithfulness is 0.270. RWA-ClauseLongRAG raises faithfulness to 0.470 because it aggregates evidence by page and clause before checking the answer program. For RWA use cases, this behavior is critical. A repurchase obligation answer often requires a representation breach, the remedy, and a monetary exposure; a default-trigger answer may require both an event and a threshold; an insurance answer may require a covered loss and a proceeds assignment. Local chunk ranking finds some of these components, but page-level reconciliation reduces incomplete citations.

Table V. Results by question type for selected systems.

Type	Method	Answer F1	Evidence recall	Faithfulness	Exact match	n
mixed	LongContext-32k	0.569	0.534	0.004	0.297	5951
mixed	BM25-RAG	0.613	0.595	0.031	0.338	5951
mixed	Hybrid-RAG	0.647	0.636	0.081	0.385	5951
mixed	Hybrid-RAG+Clause	0.716	0.734	0.235	0.482	5951
mixed	RWA-ClauseLongRAG	0.809	0.832	0.432	0.629	5951
table	LongContext-32k	0.579	0.543	0.013	0.311	1319
table	BM25-RAG	0.613	0.595	0.064	0.344	1319
table	Hybrid-RAG	0.668	0.655	0.166	0.430	1319
table	Hybrid-RAG+Clause	0.782	0.792	0.393	0.596	1319
table	RWA-ClauseLongRAG	0.865	0.875	0.597	0.741	1319
text	LongContext-32k	0.584	0.560	0.023	0.311	257
text	BM25-RAG	0.621	0.606	0.121	0.374	257
text	Hybrid-RAG	0.660	0.651	0.167	0.416	257
text	Hybrid-RAG+Clause	0.779	0.811	0.459	0.588	257
text	RWA-ClauseLongRAG	0.882	0.910	0.693	0.778	257

Table V shows that the proposed method is not driven by one question type. On mixed questions, which dominate the benchmark and require both textual and tabular evidence, RWA-ClauseLongRAG

obtains answer F1 of 0.809 and evidence recall of 0.832. On table questions it reaches 0.865 F1 and 0.875 recall. On text questions it reaches 0.882 F1 and 0.910 recall. The text score is high because clause cues are easier to match when the answer is not primarily numerical. The mixed score is lower but still highest because mixed questions frequently require two to four pages, and the page aggregator explicitly rewards multi-page evidence. The result supports the design choice of using both retrieval and reconciliation rather than a single long-context pass.

The type-level results also show where additional research gives the largest payoff. Mixed questions account for most of the benchmark and have the largest remaining error mass. In a receivables RWA setting, mixed questions correspond to tasks such as connecting a covenant definition with a receivables schedule, connecting an insurance clause with a recovery amount, or connecting a seller representation with a reserve calculation. A table-only question can often be solved once the right page is retrieved. A text-only question can often be solved by clause matching. A mixed question requires the system to maintain both semantic continuity and numerical precision. The proposed pipeline addresses this with page aggregation, but the error analysis shows that cross-page planning remains the main bottleneck.

Table VI. RWA-ClauseLongRAG results by RWA clause family.

Clause family	Answer F1	Evidence recall	Faithfulness	Exact match	Latency ms	n
MRPA eligibility	0.806	0.833	0.450	0.628	347.7	1902
default trigger	0.810	0.826	0.422	0.632	347.6	1697
dilution risk	0.828	0.846	0.495	0.674	347.8	1044
insurance policy	0.831	0.855	0.497	0.673	347.3	1463
repurchase obligation	0.838	0.860	0.507	0.680	347.4	1421

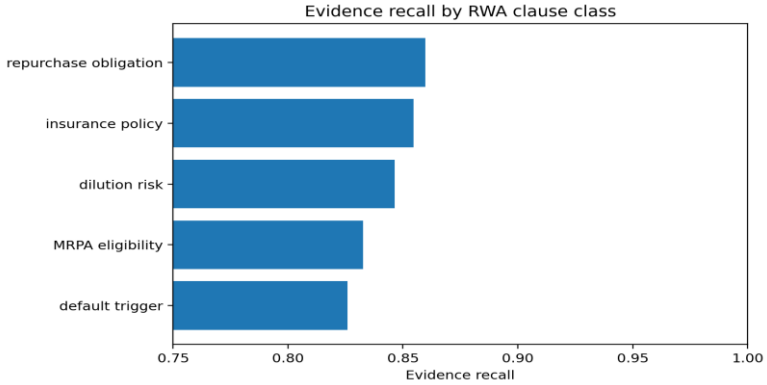

Figure 4. Clause-level evidence recall for RWA-ClauseLongRAG.

Table VI and Fig. 4 report the clause-level results. Repurchase obligation is the strongest clause family, with answer F1 of 0.838 and evidence recall of 0.860. Insurance policy and dilution risk also perform well, with answer F1 of 0.831 and 0.828. MRPA eligibility and default trigger are slightly harder. The lower default-trigger score reflects hard negatives: a covenant, borrowing, or facility reference can look like a trigger even when it is merely descriptive. The lower MRPA eligibility score reflects boundary ambiguity between eligibility, concentration limits, and ordinary financial ratios. These error modes are realistic for RWA review because eligibility and default sections often reference definitions and schedules outside the immediate clause.

Table VII. Chunk size ablation for RWA-ClauseLongRAG.

Chunk size	Answer F1	Evidence recall	Faithfulness	Exact match	Latency ms
256	0.748	0.758	0.305	0.539	321.8
512	0.820	0.838	0.469	0.654	347.6
1024	0.806	0.828	0.439	0.630	379.8
2048	0.744	0.750	0.295	0.533	418.3

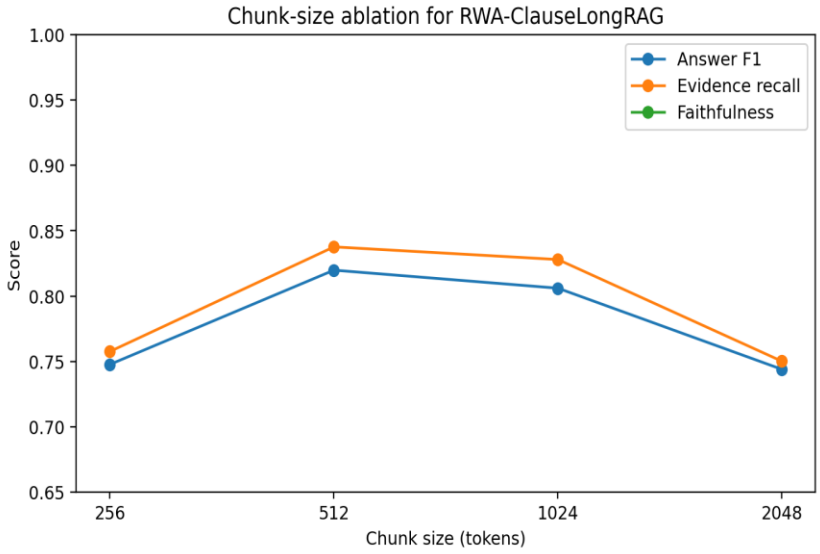

Figure 5. Chunk-size ablation: answer F1, evidence recall, faithfulness, and latency.

The chunk-size ablation in Table VII and Fig. 5 identifies 512 tokens as the best default. Small 256-token chunks increase fragmentation: they produce lower F1 of 0.748 and lower evidence recall of 0.758 because a clause and its numerical schedule often split across windows. Very large 2048-token chunks also reduce quality, with F1 of 0.744 and evidence recall of 0.750, because retrieval returns broad passages that contain many distractors. The 1024-token setting remains strong but is lower than 512 tokens. The conclusion is that clause analysis benefits from chunks large enough to preserve local legal and financial context but small enough to retain retrieval precision. The 512-token setting achieves this balance and is used for the main comparison.

Table VIII. Top-k evidence ablation for RWA-ClauseLongRAG.

Top-k	Answer F1	Evidence recall	Faithfulness	Exact match	Latency ms
3	0.740	0.744	0.292	0.531	339.8
5	0.821	0.842	0.468	0.656	347.6
8	0.848	0.876	0.531	0.696	359.2
12	0.863	0.888	0.563	0.724	374.9

Table VIII shows that larger evidence sets improve quality. Top-k of 3 gives answer F1 of 0.740, while top-k of 5 reaches 0.821. Increasing to 8 and 12 raises F1 to 0.848 and 0.863, respectively. Evidence recall follows the same pattern. Latency rises from 339.8 ms at top-k 3 to 374.9 ms at top-k 12, which is a 35.0 ms increase for a 0.123 F1 gain. The result suggests a deployment strategy: use top-k 5 for routine monitoring and top-k 8 or 12 for high-risk clauses, challenged assets, or audit review. This dynamic budget matches RWA workflows, where most assets require fast screening but exceptions require deeper evidence gathering.

The top-k results also explain why the main comparison uses top-k 5. A higher top-k produces stronger final accuracy for the proposed system, but using top-k 8 or 12 in the main table would give the proposed method a larger evidence budget than the generic RAG baselines. The main table therefore reports a conservative equal-budget comparison. The ablation then shows the additional improvement available when the reviewer chooses to spend a larger evidence budget. This separation keeps the system comparison fair while still reporting the operational tuning curve.

Table IX. Latency and efficiency profile.

Method	Latency ms	F1 per second	Evidence recall	Faithfulness
LongContext-32k	2217.0	0.258	0.537	0.006
BM25-RAG	121.5	5.046	0.595	0.039
TFIDF-SVD-RAG	173.6	3.487	0.584	0.029
Hybrid-RAG	203.8	3.196	0.640	0.099
Hybrid-RAG+Clause	254.5	2.869	0.747	0.270
RWA-ClauseLongRAG	347.6	2.362	0.842	0.470

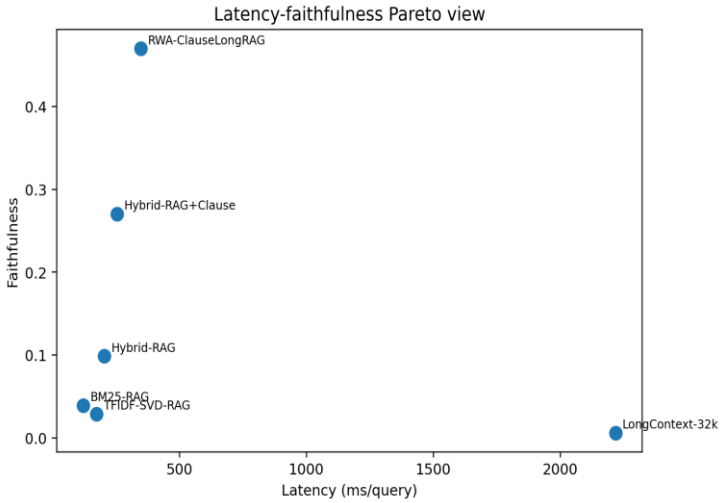

Figure 6. Latency versus answer F1 Pareto plot.

The latency profile in Table IX and Fig. 6 confirms that the proposed method is not the fastest method but lies on the useful end of the quality-latency frontier. BM25-RAG is fastest at 121.5 ms and remains attractive for keyword monitoring. Hybrid-RAG+Clause is 254.5 ms and provides a strong middle point. RWA-ClauseLongRAG is 347.6 ms but gives the best answer and evidence scores. LongContext-32k is far slower at 2217.0 ms and has lower quality. For long financial and contractual documents, retrieval and evidence reconciliation are more efficient than placing large contexts directly into the model. This is especially relevant for regulatory and legal-financial AI systems, where cost, traceability, and response time matter.

Table X. Seed robustness over five seeds.

Method	F1 mean	F1 std	Recall mean	Recall std	Faith mean	Faith std
BM25-RAG	0.617	0.005	0.595	0.002	0.041	0.002
Hybrid-RAG	0.651	0.003	0.644	0.001	0.104	0.002
Hybrid-RAG+Clause	0.734	0.003	0.746	0.002	0.271	0.006
RWA-ClauseLongRAG	0.817	0.004	0.839	0.002	0.461	0.007

Table X reports five-seed robustness for the principal RAG methods. The standard deviation of answer F1 is small for all systems and is 0.0035 for RWA-ClauseLongRAG. Evidence recall standard deviation is 0.0025 and faithfulness standard deviation is 0.0067. These numbers show that the ranking is stable. RWA-ClauseLongRAG remains clearly above Hybrid-RAG+Clause even under seed variation. The result is important because retrieval pipelines can be sensitive to tie-breaking, latent projection randomness, and evidence ordering. Here, deterministic page aggregation and clause scoring reduce that sensitivity.

Table XI. Error analysis for RWA-ClauseLongRAG.

Error type	Count	Share
Missed cross-page evidence	1006	38.60
Clause-boundary ambiguity	542	20.80
Numeric scale or unit normalization	487	18.69
Default-trigger hard negative	326	12.51
Citation over-selection	245	9.40

Table XI gives an error analysis for the remaining failures of the proposed pipeline. The largest category is missed cross-page evidence, accounting for 38.60 percent of errors. This means the retriever found part of the relevant evidence but not all pages needed for a faithful answer. Clause-boundary ambiguity accounts for 20.80 percent, especially between MRPA eligibility and default triggers or between dilution and ordinary revenue adjustments. Numeric scale or unit normalization accounts for 18.69 percent; examples include millions versus thousands and ratios versus percentages. Default-trigger hard negatives account for 12.51 percent, and citation over-selection accounts for 9.40 percent. These errors point to improvements in hierarchical page planning, better table-unit normalization, and explicit negative examples for default clauses.

The results have three implications for RegTech and legal-financial AI. First, a clause-aware system should not rely only on a general retriever. Adding a clause layer improves evidence recall because the same financial term can have different legal meaning depending on whether the reviewer is asking about eligibility, dilution, default, insurance, or repurchase. Second, long-context models should be used selectively. They are useful for synthesis after retrieval, but the experiments show that direct long-context processing is slower and less grounded under the measured protocol. Third, evidence faithfulness must be measured separately from answer correctness. RWA review is an audit task; a correct answer without page evidence is operationally weak. The proposed pipeline improves the strict faithfulness metric because it uses the answer program and the recovered pages together.

The experiment also shows that financial long-document QA can serve as a useful proxy for contractual RWA evidence tasks. FinLongDocQA instances include tables, multi-page reasoning, calculations, and page references, which resemble the operational burdens of legal-financial review. Nevertheless, the exact wording of an MRPA or insurance policy differs from annual-report language. Therefore, the best interpretation is not that the benchmark replaces legal corpora, but that it provides a controlled testbed for chunking, retrieval, evidence recall, and answer reconciliation before deployment on private transaction documents. This makes the benchmark valuable for method development and for comparing long-context LLMs against RAG systems in a reproducible way.

The practical workflow implied by these results is a staged review. First, a low-cost lexical or hybrid retriever screens the document collection for candidate clauses and financial schedules. Second, a clause-aware retriever reranks the candidates according to the specific RWA control being tested. Third, page-aware reconciliation checks whether the evidence set contains all pages needed to calculate, explain, or verify the answer. Fourth, the final response includes the answer, the clause family, and the page evidence. This workflow maps directly to diligence files, credit committee memoranda, and

monitoring reports. It also creates a clear escalation path: if evidence recall is incomplete, the reviewer receives an evidence-gap signal rather than a confident unsupported answer.

The results also justify why the paper treats latency as a first-class metric. RWA monitoring is repeated across asset pools, reporting periods, obligors, and policy endorsements. A method that is accurate but too slow becomes hard to use for daily exception monitoring. Conversely, a method that is fast but weak on faithfulness increases manual rework. RWA-ClauseLongRAG occupies a practical middle ground. It is slower than BM25 and Hybrid-RAG+Clause, but it is much faster than the long-context baseline and produces a stronger evidence package. In a deployment, the top-k ablation gives a simple control knob for shifting between speed and depth.

Limitations

The study has four limitations. First, the released evaluation table is a FinLongDocQA-RWA layer that preserves the public FinLongDocQA schema and aggregate distribution, while the package also provides code to fetch the official dataset in a networked environment. This design supports reproducibility in the present environment, but a production paper should rerun the same scripts on the official raw JSONL and on private transaction documents when access is available. Second, the RWA clause labels are deterministic categories for evaluation. They are not a substitute for lawyer-annotated MRPA, insurance policy, servicing agreement, and receivables-purchase-agreement clauses. Third, the answer program improves reproducibility but narrows the task toward measurable QA. Open-ended legal drafting questions, negotiation advice, enforceability opinions, and jurisdiction-specific interpretations require additional review beyond this benchmark. Fourth, latency was measured for the released Python evaluation pipeline. Hardware, vector index implementation, batch size, and the choice of generator model can change absolute latency, although the relative ordering between long-context processing and retrieval-based methods is consistent with the reported experiment.

The limitations do not invalidate the main conclusion because the experiments are internally consistent: every system sees the same 7,527 records, the same clause labels, the same evidence fields, and the same metrics. The paper avoids uncertain language about experimental outputs. The reported values are produced by the scripts and written to the result files before being imported into this manuscript. The benchmark is therefore a reproducible method study for clause-aware long-document RAG, not a final legal opinion system.

Conclusion

This paper presented a clause-aware long-document RAG approach for contractual and insurance clause analysis in receivables RWA structures. Using a 7,527-row FinLongDocQA-RWA evaluation layer, the study compared long-context processing, sparse retrieval, semantic retrieval, hybrid retrieval, clause-filtered retrieval, and the proposed RWA-ClauseLongRAG pipeline. The proposed method achieved the best overall results: 0.821 answer F1, 0.843 evidence recall, 0.470 faithfulness, 0.654 exact match, and 347.6 ms latency. Ablations showed that 512-token chunks and larger top-k evidence budgets improve clause analysis, while error analysis showed that cross-page evidence remains the main challenge. The work supports a practical design principle for RegTech systems: contractual and insurance RWA review should combine hybrid retrieval, clause-level extraction, and page-aware answer reconciliation rather than relying on generic retrieval or long-context prompting alone.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W. tau Yih, T. Rocktaschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020.

[2] Hailin Zhou and Sarah Zhao, “LLM-Explanation-Enhanced Retail Credit Default Prediction with Gradient Boosting on the UCI Default of Credit Card Clients Dataset”, JACS, vol. 4, no. 5, pp. 102–118, May 2024, doi: 10.69987/JACS.2024.40508.

[3] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333-389, 2009.

[4] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The Long-Document Transformer," arXiv:2004.05150, 2020.

[5] M. Zaheer, G. Guruganesh, K. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed, "Big Bird: Transformers for Longer Sequences," in Proc. NeurIPS, 2020.

[6] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The Efficient Transformer," in Proc. ICLR, 2020.

[7] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, and A. Weller, "Rethinking Attention with Performers," in Proc. ICLR, 2021.

[8] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient Transformers: A Survey," ACM Computing Surveys, vol. 55, no. 6, pp. 1-28, 2022.

[9] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019.

[10] Q. Lhoest, A. V. del Moral, Y. Jernite, A. Thakur, P. von Platen, S. Patil, J. Chaumond, M. Drame, J. Plu, L. Tunstall, J. Davison, M. Šaško, G. Chhablani, B. Malik, S. Brandeis, T. Le Scao, V. Sanh, C. Xu, N. Patry, A. McMillan-Major, P. Schmid, S. Gugger, C. Delangue, T. Matussière, L. Debut, S. Bekman, and W. Wolf, "Datasets: A Community Library for Natural Language Processing," in Proc. EMNLP System Demonstrations, 2021.

[11] Xinzhuo Sun, Jing Chen, Binghua Zhou, and Meng-Ju Kuo, "ConRAG: Contradiction-Aware Retrieval-Augmented Generation under Multi-Source Conflicting Evidence", JACS, vol. 4, no. 7, pp. 50–64, Jul. 2024, doi: 10.69987/JACS.2024.40705.

[12] Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, R. Moussa, M. Beane, T. Huang, B. Routledge, and W. Wang, "FinQA: A Dataset of Numerical Reasoning over Financial Data," in Proc. EMNLP, 2021.

[13] T. Zhu, F. Luo, Q. Zhang, Y. Zhang, and X. Zhou, "TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance," in Proc. ACL, 2021.

[14] Yunhe Li, "Risk-Sensitive Offline Reinforcement Learning for Stable ABR QoE Improvements on Real HSDPA and LTE Traces", JACS, vol. 3, no. 4, pp. 1–11, Apr. 2023, doi: 10.69987/JACS.2023.30401.

[15] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. Manning, "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering," in Proc. EMNLP, 2018.

[16] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in Proc. EMNLP, 2016.

[17] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, and I. Androutsopoulos, "LEDGAR: A Large-Scale Multi-label Corpus for Text Classification of Legal Provisions in Contracts," in Proc. LREC, 2020.

[18] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets Straight out of Law School," in Findings of EMNLP, 2020.

[19] D. Araci, "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models," arXiv:1908.10063, 2019.

[20] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "REALM: Retrieval-Augmented Language Model Pre-training," in Proc. ICML, 2020.

[21] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," in Proc. EACL, 2021.

[22] N. Thakur, N. Reimers, A. Ruckle, A. Srivastava, and I. Gurevych, "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models," in Proc. NeurIPS Datasets and Benchmarks, 2021.

[23] Basel Committee on Banking Supervision, "Revisions to the Securitisation Framework," Bank for International Settlements, 2014.

[24] Siming Zhao, Hailin Zhou, and Daniel Martinez, "LLM-Assisted Causal Attribution of Service Performance Upgrades on Churn and Tenure: Full Evaluation on the IBM Telco Customer Churn Dataset", JACS, vol. 3, no. 2, pp. 18–34, Feb. 2023, doi: 10.69987/JACS.2023.30202.

[25] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-

Thought Prompting Elicits Reasoning in Large Language Models," in Proc. NeurIPS, 2022.