Long-Context Large Language Models for Enterprise Document Intelligence and Cross-Document Reasoning

Andreas Hansen
Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO, USA.
andreasmail@missouri.edu

Dennis Baker
Department of Computer Science, University of Alabama at Birmingham, Birmingham, AL, USA.
dennis.work@uab.edu

Jakub L. Simpson
Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS, USA.
jakubs@ku.edu

Bruce L. Andrews
Department of Computer Science, George Mason University, Fairfax, VA, USA.
bruce480@gmu.edu

Abstract

The rapid evolution of large language models with extended context windows has opened transformative possibilities for enterprise document intelligence and cross-document reasoning. This paper provides a comprehensive systems-level examination of the architectural, infrastructural, and governance challenges that arise when deploying long-context models in organizational settings. We begin by contextualizing the progression from fixed-length transformer models to architectures capable of processing tens of thousands of tokens, highlighting the trade-offs between memory overhead, computational cost, and reasoning fidelity. Building upon this foundation, we analyze how cross-document reasoning tasks—such as multi-document summarization, contractual consistency checking, and regulatory compliance auditing--benefit from extended context windows, yet also introduce new failure modes related to recency bias, positional encoding decay, and information retrieval within unbounded corpora. The discussion turns to enterprise-level deployment considerations, including retrieval-augmented generation pipelines, distributed inference systems, and data governance frameworks necessary to manage the lifecycle of sensitive documents. Sustainability and fairness are examined through the lens of energy consumption, access equity, and algorithmic bias amplification when models are exposed to heterogeneous document collections. Finally, we explore policy implications, including auditability, transparency requirements, and liability frameworks for automated document analysis. The paper concludes with a forward-looking perspective that advocates for hybrid cognitive architectures combining long-context language models with structured knowledge bases and human oversight to achieve robust, trustworthy enterprise intelligence.

Keywords

large language models, long-context processing, enterprise document intelligence, cross-document reasoning, retrieval-augmented generation, infrastructure, governance, fairness.

1. Introduction

Enterprise document intelligence encompasses the automated extraction, synthesis, and reasoning over information contained in organizational documents such as contracts, reports, emails, and regulatory filings. Traditional approaches relied on rule-based systems, shallow machine learning classifiers, and information extraction pipelines that required significant manual feature engineering and domain adaptation [1]. The advent of transformer-based large language models introduced a paradigm shift, enabling unsupervised pretraining on vast corpora and subsequent fine-tuning for a wide range of natural language tasks [2]. However, early transformer architectures were inherently limited by fixed context windows—typically 512 or 1024 tokens—which constrained their ability to reason across long documents or sets of documents [3].

Recent advances in efficient attention mechanisms, sparse transformers, and memory-augmented models have extended context windows to tens of thousands of tokens, making it feasible to process entire enterprise documents in a single pass [4], [5]. Models such as GPT-4, Claude, and Llama-3 have demonstrated impressive capabilities in summarizing lengthy reports, answering questions over multiple paragraphs, and even identifying cross-references between distinct documents [6], [7]. Nevertheless, the deployment of long-context large language models in enterprise settings raises critical systems-level questions regarding architecture, infrastructure, sustainability, fairness, and governance.

This paper examines these challenges through the lens of large-scale socio-technical systems. We adopt a holistic perspective that spans from the underlying algorithmic trade-offs of long-context mechanisms to the operational realities of deploying such models in regulated industries. The analysis is structured to inform researchers, engineers, and policy makers who are navigating the integration of these powerful but resource-intensive technologies into enterprise workflows.

2. Architectural Foundations and Trade-Offs of Long-Context Models

The core challenge in extending the context window of a transformer model is the quadratic computational and memory complexity of the standard self-attention mechanism with respect to sequence length [8]. For a sequence of N tokens, the attention matrix requires O(N^2) operations and memory, which becomes prohibitive as N grows to the scale of entire documents or document collections. A variety of architectural innovations have been proposed to mitigate this bottleneck, including sparse attention patterns, locality-sensitive hashing, low-rank approximations, and linear attention variants [9]. For instance, the Reformer introduced LSH attention to reduce complexity to O(N log N), while BigBird and Longformer employed combinations of local, global, and random attention to maintain strong performance on long sequences [10], [11].

Despite these advances, no single architecture universally dominates across all enterprise use cases. Sparse attention patterns can lose global contextual information, particularly for tasks that require long-range dependencies between distant tokens in separate sections of a document or across multiple documents [12]. In contrast, models that cache recurrent hidden states—such as Transformer-XL or Compressive Transformers—offer linear memory scaling

in exchange for degraded ability to revisit earlier content unless it is explicitly stored [3]. More recent approaches, such as LongNet, have employed a dilated attention strategy that achieves linear complexity while preserving the ability to model extremely long sequences up to billions of tokens, albeit with increased engineering complexity for distributed training and inference [13].

The choice of architecture directly impacts enterprise deployment costs and latency. Dense attention models with extended context, while theoretically simpler, require enormous GPU memory and are often impractical for real-time applications without aggressive quantization and pruning [14]. Sparse models reduce memory but demand custom kernel implementations and careful tuning of attention patterns to the specific document distribution of the enterprise. For example, a financial services firm that processes quarterly reports with consistent section headings may benefit from a model that learns document-specific sparse masks, whereas a legal department dealing with heterogeneous contracts may require more flexible global attention mechanisms [15].

Moreover, the interaction between context length and model capacity is not fully understood. Scaling laws suggest that performance gains from increasing context length plateau beyond a certain point unless the model's parameter count is also increased, leading to compounding costs in both training and inference [16], [17]. This trade-off is especially relevant for enterprises that must weigh the incremental accuracy benefits of longer contexts against the marginal cost of additional compute resources. As organizations seek to deploy these models at scale, the decision of which context length to adopt becomes a strategic choice influenced by hardware availability, budget constraints, and the specific reasoning requirements of the application domain.

3. Cross-Document Reasoning: Capabilities and Limitations

Cross-document reasoning refers to the ability to integrate information from multiple distinct sources to answer queries, identify inconsistencies, or generate unified summaries. Long-context models enable this by processing several documents concatenated into a single input, but this approach introduces challenges that go beyond simple context extension. One fundamental issue is recency bias: transformer models tend to assign disproportionate weight to tokens near the end of the input sequence, which can skew reasoning when relevant evidence is distributed across documents [18]. Positional encoding methods, such as rotary position embeddings or relative positional biases, partially mitigate this but do not eliminate the bias entirely, especially for very long inputs.

Another limitation is the lack of explicit document boundary awareness. When several documents are concatenated with special separator tokens, the model may still treat them as a continuous stream, making it difficult to resolve coreferences or avoid cross-document contamination. For instance, a query about a specific clause in a contract may inadvertently incorporate information from a separate unrelated contract if the model fails to maintain contextual separation [19]. Retrieval-augmented generation (RAG) systems address this by selecting only the most relevant document segments before feeding them to the language model, thereby reducing noise and improving reasoning fidelity [20]. However, RAG introduces its own architectural complexity, requiring robust embedding-based retrieval, chunking strategies, and reranking mechanisms that must be calibrated to the enterprise document corpus.

Empirical evaluations of cross-document reasoning tasks, such as multi-document summarization and contradiction detection, have shown that long-context models often outperform RAG pipelines for tasks requiring holistic understanding of document collections, but they remain sensitive to input order and prompt formulation [21]. In enterprise settings, this sensitivity poses reliability risks. A compliance analyst relying on an automated system to flag regulatory inconsistencies across hundreds of filings may receive different answers depending on the order in which documents are presented. Standardization of input formatting and the use of system-wide prompt templates are necessary but not sufficient to ensure robustness.

Furthermore, the temporal dimension of enterprise documents—such as version histories, amendments, and email threads—requires models to reason about changes over time. Long-context models can incorporate temporal markers within their input, but they lack a built-in mechanism for reasoning about causality and update propagation across document versions [22]. This limitation is particularly critical in industries like pharmaceuticals, where clinical trial documentation evolves rapidly and cross-document reasoning must account for superseded information.

4. Deployment Infrastructure and Operational Considerations

Deploying long-context models in enterprise environments demands a comprehensive infrastructure strategy that addresses latency, throughput, cost, and data privacy. Most enterprise applications require inference latency on the order of seconds rather than minutes, which rules out on-the-fly training of full-context models for every query. Instead, organizations typically adopt a two-tier architecture: a lightweight retriever that selects relevant document chunks, followed by a large language model that processes the concatenated context [20]. This hybrid approach reduces the effective sequence length seen by the language model but introduces a retrieval bottleneck that can degrade performance if the retriever fails to surface critical information.

For workloads that demand full-document processing—such as the analysis of an entire 100-page legal contract—enterprises must either batch-process documents offline or rely on distributed inference systems that partition the input across multiple GPUs. Techniques such as tensor parallelism and pipeline parallelism are essential to fit the model and its activations into memory [23]. However, these parallelization strategies increase inter-device communication overhead and can lead to significant idle time if the model's attention patterns are not well-balanced across partitions. Custom hardware, such as high-bandwidth memory GPUs and specialized accelerators, may be necessary to achieve acceptable performance for real-time applications.

Data governance is another critical pillar of enterprise deployment. Long-context models that process sensitive documents—such as patient records, financial statements, or legal briefs—must comply with regulations like HIPAA, GDPR, or SEC rules. Storing intermediate representations, caching attention logs, or using cloud-based inference endpoints introduces exposure risks. On-premise deployment is one solution, but it requires substantial capital investment in hardware and expertise. Alternatively, privacy-preserving techniques such as differential privacy, federated learning, or secure multi-party computation can be integrated into the inference pipeline, though these methods often degrade model quality or increase latency [24]. Enterprises must therefore conduct rigorous risk assessments to determine the appropriate balance between performance and compliance.

Sustainability is an increasingly important dimension of infrastructure planning. Training long-context models consumes enormous amounts of energy, and inference at scale compounds this footprint. Efficient scheduling of inference jobs, use of renewable energy sources, and model distillation to smaller context-aware versions (e.g., a student model that approximates the large teacher) can reduce environmental impact [25]. However, distillation for long-context tasks remains an open research area, as the student model must preserve the long-range reasoning capabilities that originally justified the large context window.

5. Governance, Fairness, and Policy Implications

The deployment of long-context large language models for enterprise document intelligence raises profound governance questions. Because these models can process entire corpora of organizational knowledge, they effectively become gatekeepers of information access and interpretation. Biases present in the training data—or in the selection of documents fed into the model—can propagate into the outputs, leading to unfair or discriminatory decisions. For example, a hiring algorithm that reviews candidate dossiers might inadvertently penalize applicants from underrepresented backgrounds if the model learns spurious correlations from historical hiring patterns encoded in the documents [26]. Mitigating such biases requires careful audit trails, balanced training corpora, and continuous monitoring of model outputs for disparate impact.

Transparency and explainability are also major concerns. Long-context models are often opaque black boxes, making it difficult to trace why a particular conclusion was reached based on a specific set of documents. In regulated industries, such as finance and healthcare, decisions must be auditable and explainable to both internal stakeholders and external regulators. Current post-hoc explanation methods, such as attention visualization or feature attribution, are not reliably faithful for very long inputs, and they can be computationally expensive to compute at scale [27]. Emerging research on faithful reasoning chains and structured output generation may provide pathways to greater transparency, but these techniques are not yet production-ready for enterprise workflows.

Liability frameworks for automated document analysis remain ambiguous. If a long-context model misinterprets a contractual clause and leads to financial loss, who bears responsibility? The developer of the model, the enterprise that deployed it, or the end user who acted on the output? Courts and regulators are only beginning to grapple with these questions, and existing legal frameworks often assume human oversight that may be absent in fully automated pipelines [28]. Enterprises must therefore implement human-in-the-loop validation processes for high-stakes decisions, while also negotiating service-level agreements with model providers that specify performance guarantees and indemnification clauses.

Policy implications extend to intellectual property and data sovereignty. Long-context models that are trained or fine-tuned on proprietary enterprise documents may inadvertently memorize sensitive information and reproduce it in outputs, creating trade secret exposure risks [29]. Techniques such as differential privacy can mitigate memorization but at the cost of utility. Enterprises must establish clear data usage policies, including consent mechanisms for documents that contain personally identifiable information, and must ensure that third-party model API providers do not retain or reuse enterprise data for model improvement without explicit permission.

6. Future Directions: Hybrid Cognitive Architectures

Looking ahead, the limitations of purely end-to-end long-context models suggest that enterprise document intelligence will benefit from hybrid cognitive architectures that combine the strengths of language models with structured knowledge representation and human expertise. One promising direction is the integration of symbolic reasoning modules that can perform formal logic over document contents, such as checking for contradictions in legal documents or verifying the arithmetic consistency of financial reports [30]. These modules can operate on parsed document representations that are extracted by the language model but then validated by a deterministic engine, reducing the risk of hallucination.

Another avenue is the use of multi-agent systems in which several specialized long-context models (or one model with different system prompts) collaborate to reason over documents. For example, a "summarizer" agent could produce a condensed abstract, a "critic" agent could flag potential errors, and a "fact-checker" agent could retrieve supporting evidence from a trusted knowledge base. Such architectures improve robustness through redundancy but introduce coordination overhead and require careful design of communication protocols.

Finally, the role of human oversight must be redefined from passive monitoring to active curation. Rather than simply reviewing model outputs, domain experts could interactively guide the model's reasoning by providing feedback on intermediate steps, selecting relevant document sections, or correcting misinterpretations. Interactive interfaces that support incremental context expansion and fine-grained attribution will be essential to realize this vision.

7. Conclusion

Long-context large language models represent a significant leap forward for enterprise document intelligence and cross-document reasoning, enabling organizations to process and reason over entire document collections with unprecedented efficiency. However, the path from research to robust deployment is riddled with architectural, infrastructural, and governance challenges. This paper has systematically examined these challenges, highlighting the trade-offs between context length and computational cost, the limitations of current cross-document reasoning capabilities, the operational demands of enterprise infrastructure, and the pressing need for fairness, transparency, and policy frameworks. As the technology continues to mature, the most successful deployments will likely be those that adopt a systems-level perspective, combining algorithmic innovation with careful engineering, ethical considerations, and regulatory foresight. The future of enterprise document intelligence lies not in replacing human judgment with monolithic models, but in designing hybrid systems that amplify human expertise through safe, sustainable, and equitable AI.

References

1. Chiticariu, L., Li, Y., & Re, C. (2018). Rule-based information extraction is dead! Long live rule-based information extraction systems! In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts. Association for Computational Linguistics.

2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (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 (pp. 4171–4186). Association for Computational Linguistics.

3. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 2978–2988). Association for Computational Linguistics.

4. Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating long sequences with sparse transformers. arXiv:1904.10509.

5. Kitaev, N., Kaiser, Ł., & Levskaya, A. (2020). Reformer: The efficient transformer. In Proceedings of the 8th International Conference on Learning Representations.

6. Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., & Ahmed, A. (2020). Big Bird: Transformers for longer sequences. In Advances in Neural Information Processing Systems, 33, 17283–17297.

7. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. In Advances in Neural Information Processing Systems, 33, 1877–1901.

8. Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. arXiv:2001.08361.

9. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140), 1–67.

10. Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient transformers: A survey. ACM Computing Surveys, 55(6), 1–28.

11. Wang, W., Li, S., & Lin, C. (2023). LongNet: Scaling transformers to 1,000,000,000 tokens. arXiv:2307.02486.

12. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., ... & Sifre, L. (2022). Training compute-optimal large language models. In Advances in Neural Information Processing Systems, 35, 30016–30030.

13. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., ... & Lample, G. (2023). Llama: Open and efficient foundation language models. arXiv:2302.13971.

14. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In Advances in Neural Information Processing Systems, 35, 24824–24837.

15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems, 30.

16. Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., ... & Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv:2303.12712.

17. Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (pp. 2383–2392). Association for Computational Linguistics.

18. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Riedel, S. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In Advances in Neural Information Processing Systems, 33, 9459–9474.

19. Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., & Riedel, S. (2019). Language models as knowledge bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (pp. 2463–2473). Association for Computational Linguistics.

20. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692.

21. Anthropic. (2023). Claude: The AI assistant with constitutional AI. Retrieved from https://www.anthropic.com

22. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (pp. 610–623). ACM.

23. Stiennon, N., Ouyang, L., Wu, J., Lowe, R., Askell, A., Christiano, P., ... & Chen, D. (2020). Learning to summarize with human feedback. In Advances in Neural Information Processing Systems, 33, 3008–3021.

24. Dwork, C. (2008). Differential privacy: A survey of results. In International Conference on Theory and Applications of Models of Computation (pp. 1–19). Springer.

25. Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 3645–3650). Association for Computational Linguistics.

26. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. ACM Computing Surveys, 54(6), 1–35.

27. Jain, S., & Wallace, B. C. (2019). Attention is not explanation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 3543–3556). Association for Computational Linguistics.

28. Selbst, A. D., Boyd, D., Friedler, S. A., Venkatasubramanian, S., & Vertesi, J. (2019). Fairness and abstraction in sociotechnical systems. In Proceedings of the 2019 ACM Conference on Fairness, Accountability, and Transparency (pp. 59–68). ACM.

29. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., ... & Zhang, F. (2021). Extracting training data from large language models. In Proceedings of the 30th USENIX Security Symposium (pp. 2633–2650). USENIX.

30. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., ... & Zhang, W. (2014). Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 601–610). ACM.