Dynamic, Coherent, and Lossless: The Iterative Structured Content Filling Framework for Infinite Document Parsing

Anonymous ACL submission

Abstract

Parsing long, unstructured documents into hierarchical machine-readable formats is critically challenged by the tension between document length and limited LLM context. Prevailing one-shot or stateless chunk-based methods suffer from the "lost-in-the-middle" phenomenon and fragmented structures. We argue for reframing the task as an iterative, stateful editing process. This paper introduces the **Iterative Structured Content Filling (ISCF)** framework, which incrementally builds a global tree through LLM-driven steps under partial observability. ISCF integrates three pillars: (1) **Dynamic Boundary Re-alignment (DBR)** for unsupervised chunk optimization via semantic coherence and uncertainty; (2) **Adaptive Structural Masking & Hierarchical-Aware Path Embedding (HAPE)** for efficient global awareness within finite context; (3) **Bijective Path-Content Mapping** ensuring lossless reconstruction. Extensive experiments on financial and legal documents demonstrate ISCF's superior accuracy, robustness, and efficiency over strong baselines, offering a principled solution for unbounded document structuring.

1 Introduction

The digital transformation of industries such as finance, law, and academia has generated vast repositories of long-form unstructured documents. Automating the conversion of these documents into machine-readable hierarchical formats is crucial for downstream applications like intelligent review, knowledge graph construction, and semantic analysis. However, existing methods for document structure parsing face a fundamental tension between the unbounded length of real-world documents and the fixed context windows of Large Language Models (LLMs), as well as between localized text processing and the need for global structural consistency(Xu et al., 2020).

Prevailing approaches exhibit significant limitations. Full-text ingestion is immediately constrained by context ceilings and suffers from the well-documented "lost-in-the-middle" phenomenon, where model performance degrades sharply on information located in the middle of long inputs (Liu et al., 2023). Independent chunk parsing treats each segment in isolation, leading to catastrophic structural fractures at boundaries due to a lack of cross-chunk state. Overlapping sliding windows introduce redundant computation and often fail to resolve ambiguous content attribution across windows (Ding et al., 2024). Underlying these methods is a treatment of the task as a one-shot or stateless sequence-to-tree conversion (Sutskever et al., 2014; Dong and Lapata, 2016), which fundamentally ignores the inherent continuity and hierarchical dependencies within long documents.

We argue that parsing long documents should be modeled not as a single transduction, but as an iterative, stateful editing process. Analogous to how a human editor builds an outline, the system should maintain and refine a growing structural skeleton, integrating new content against it and making dynamic adjustments based on evolving context. This paradigm directly addresses the challenge of partial observability under limited context windows and ensures each local decision is grounded in a distilled, global structural view.

In this work, we formalize the task as **Structured Content Filling** and introduce the **Iterative Structured Content Filling (ISCF)** framework. ISCF iteratively refines an initially empty hierarchical tree into a complete structure through a series of LLM-driven editing steps. Each step is conditioned on a local text segment and a compressed, adaptive view of the current global structure. The framework is built upon three synergistic technical pillars:

1. **Dynamic Boundary Re-alignment (DBR)**:

An online, unsupervised mechanism that optimizes chunk boundaries based on semantic coherence and local parsing uncertainty, preventing logical units from being severed by fixed segmentation (Duarte et al., 2024).

2. **Adaptive Structural Masking & Hierarchical-Aware Path Embedding (HAPE)**: A context compression and prompting strategy that retains the complete hierarchical skeleton and focuses attention on a relevant anchor point, effectively reconciling unbounded document structure with finite LLM context.

3. **Bijective Path-Content Mapping**: A deterministic protocol that guarantees lossless, 100% accurate reconstruction of the original text from the final structured tree, eliminating content omission or duplication.

Our contributions are summarized as follows:

1. We propose ISCF, a novel, model-agnostic iterative paradigm for long-document structure parsing that ensures global coherence through stateful, incremental tree construction and editing.

2. We design DBR, an unsupervised chunking algorithm that dynamically realigns boundaries based on semantic cohesion and a prompting-based self-consistency measure, significantly improving local structural continuity.

3. We develop adaptive tree masking and HAPE prompting to efficiently maintain structural awareness within LLM context limits, mitigating information decay in long contexts.

4. Through comprehensive experiments on challenging financial and legal document datasets, we demonstrate ISCF's superior accuracy, robustness, and efficiency over strong baselines, providing a principled solution for transforming unbounded unstructured text into precise hierarchical representations.

2 Method

We formalize the structural parsing of long, unstructured documents as a **Structured Content Filling** task. Given a long document $D = [s_0, s_1, \ldots, s_{M-1}]$ consisting of a sequence of sentenceswhere each s_i is a maximal textual unit ending with terminal punctuation and no explicit structural markers (e.g., headings or section delimiters) are presentour goal is to iteratively refine an initially empty hierarchical tree $T^{(0)}$, containing only a root node, into a complete, machine-readable structure tree $T^{(K)} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ denotes the set of nodes (including both structural and content nodes) and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ represents the set of directed parent-child edges that define the hierarchical structure(Wang et al., 2025), and K denotes the total number of LLM inference steps.

Unlike conventional one-shot approaches (e.g., sequence-to-tree or sequence-to-sequence models) (Sutskever et al., 2014; Dong and Lapata, 2016), our proposed **Iterative Structured Content Filling** (ISCF) framework formulates this as an iterative, stateful editing process. Constrained by the limited context window of large language models (LLMs), this process operates under partial observability: at each step, the system observes only a local text segment and a compressed view of the current structure, then executes discrete edit actions (e.g., creating nodes, assigning content) to update its internal state. This paradigm directly addresses the well-documented lost-in-the-middle phenomenon in long-context LLMs (Liu et al., 2023)where performance degrades on middle segments of long inputsby ensuring that each decision is grounded in both recent content and a distilled global outline.

The proposed solution is illustrated in Figure 1.

2.1 Problem Formulation

The input document D is a sequence of sentences from an unstructured source lacking predefined headings or sections. The target structure tree T comprises two types of nodes, both inferred purely from semantic content:

- **Structural nodes** $n_s \in \mathcal{N}_s$: represent inferred hierarchical sections (e.g., chapters, subsections). Each is formalized as a tuple $(\texttt{title}, \ell)$, where the `title` is generated by the model based on semantic content, and $\ell \in \{1, 2, \ldots, L\}$ denotes the inferred depth level. The span covered by n_s is implicitly defined as the union of spans of all its descendant content nodes: $\text{span}(n_s) := \bigcup_{n_c \in \text{Desc}(n_s) \cap \mathcal{N}_c} \text{span}(n_c)$.

- **Content nodes** $n_c \in \mathcal{N}_c$: represent contiguous sentence spans belonging to a specific structural node, formalized as (π, span). Here, π is a unique path identifier from

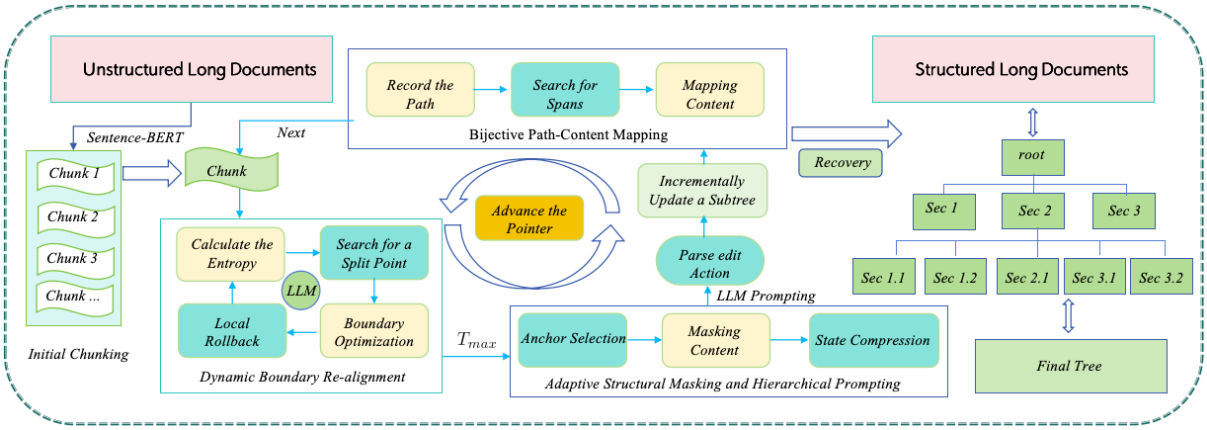

Figure 1: Architecture of the Iterative Structured Content Filling Framework.

the root to the parent structural node (e.g., `root.chapter1.section2`), and span $= (i, j)$ denotes the inclusive sentence indices in D corresponding to the original text. This ensures deterministic recovery and avoids ambiguity from repeated phrases.

Constraints: The final tree $T^{(K)}$ must satisfy:

1. **Content Coverage**: $\bigcup_{n_c \in \mathcal{N}_c} D[\text{span}(n_c)] = D$;
2. **Content Disjointness**: for any $n_c \neq n'_c$, $\text{span}(n_c) \cap \text{span}(n'_c) = \emptyset$;
3. **Structural Containment**: if structural node n_i is an ancestor of n_j, then $\text{span}(n_i) \supseteq \text{span}(n_j)$.

At global step t, the system maintains a state $S^{(t)} = (T^{(t)}, \Phi^{(t)}, \text{ptr}^{(t)})$, where:

- $T^{(t)}$: the partially constructed structure tree with generated structural nodes and associated content assignments.
- $\Phi^{(t)}$: a bijective mapping from finalized path identifiers π to their exact original sentence spans in D, used for lossless reconstruction.
- $\text{ptr}^{(t)}$: the index of the next unprocessed sentence in D (an absolute position in $[0, M]$).

The systems observation $O^{(t)}$the prompt fed to the LLMconsists of:

1. The **current text chunk** $C^{(t)}$: a contiguous segment starting at $\text{ptr}^{(t)}$, obtained via dynamic chunking (Section 2.2).
2. A **compressed structural context**: a masked version $T_M^{(t)}$ of $T^{(t)}$ via adaptive masking (Section 2.3).

Based on $O^{(t)}$, the LLM generates a sequence of edit actions $A^{(t)} = \{a_1, a_2, \dots\}$, which are parsed into an edit script $\Delta T^{(t)}$ and applied to $T^{(t)}$ to yield $T^{(t+1)}$. The process terminates when $\text{ptr}^{(t)} = M$, producing a final tree $T^{(K)}$ that satisfies all constraints.

2.2 Dynamic Boundary Re-alignment (DBR)

Fixed-length chunking often severs semantically coherent unitsa critical failure mode for unstructured documents (Duarte et al., 2024). We propose **Dynamic Boundary Re-alignment** (DBR), an online, unsupervised mechanism that adjusts chunk boundaries based on local parsing uncertainty.

2.2.1 Initial Chunking and Uncertainty Estimation

We first partition D into initial chunks represented as index intervals $\{(b_0, e_0), (b_1, e_1), \dots\}$ using a greedy merging algorithm based on Sentence-BERT embeddings (Reimers and Gurevych, 2019), respecting the token limit $L_{\max}$. Specifically, we iteratively merge consecutive sentences with the highest embedding similarity until adding the next sentence would exceed $L_{\max}$ tokens.

Let c denote the current chunk index being processed. At retry attempt r for chunk c, let $\Delta\mathcal{N}_c^{(r)}$ denote the set of newly generated content nodes. To estimate structural uncertainty without requiring model fine-tuning, we use a prompting-based self-consistency strategy: we run the LLM R times with different random seeds and extract the empirical distribution over semantic roles (e.g.,

topic, evidence, conclusion) assigned to each content node. We define:

$$\mathcal{H}_c^{(r)} = \frac{1}{|\Delta\mathcal{N}_c^{(r)}|} \sum_{n_c \in \Delta\mathcal{N}_c^{(r)}} \mathrm{H}\big(p_{\text{emp}}(\text{role} \mid n_c)\big), \tag{1}$$

where $\mathrm{H}(\cdot)$ is entropy and p_{emp} is the empirical role distribution. High $\mathcal{H}_c^{(r)}$ suggests the current boundary cuts through a coherent semantic unit.

2.2.2 Boundary Optimization via Semantic Coherence

When $\mathcal{H}_c^{(r)} > \tau$ and $c \geq 1$, DBR triggers a boundary adjustment between chunks $c-1$ and c. We search for a split point k (measured as the number of sentences from the start of chunk c) that maximizes intra-chunk coherence while encouraging smoothness across the boundary:

$$k^* = \arg \max_{1 \leq k \leq \min(w, |C_c|-1)} \Big[\underbrace{\mathrm{Coh}(C_c^{<k}) + \mathrm{Coh}(C_c^{\geq k})}_{\text{internal cohesion}} - \gamma \cdot \underbrace{\mathrm{Dist}(C_{c-1}, C_c^{<k})}_{\text{boundary discontinuity}} \Big], \tag{2}$$

where $w \in \mathbb{N}^+$ is the maximum number of initial sentences in chunk c eligible for reassignment, $\gamma > 0$ balances the trade-off, and $|C_c| = e_c - b_c$ denotes the number of sentences in chunk c. The coherence and distance functions are defined as:

$$\mathrm{Coh}(X) = \frac{1}{|X|^2} \sum_{s,s' \in X} \cos(\mathbf{e}_s, \mathbf{e}_{s'}), \quad |X| \geq 1, \tag{3}$$

$$\mathrm{Dist}(X, Y) = 1 - \cos(\bar{\mathbf{e}}_X, \bar{\mathbf{e}}_Y), \tag{4}$$

where $\bar{\mathbf{e}}_X$ is the mean embedding of all sentences in X. Here, $C_{c-1} = D[b_{c-1} : e_{c-1}]$ and $C_c = D[b_c : e_c]$ denote the current sentence sequences of chunks $c-1$ and c, respectively.

We then update the chunk boundaries by moving the first k^* sentences of C_c to the end of C_{c-1}:

$$e_{c-1} \leftarrow b_c + k^*, \tag{5}$$

$$b_c \leftarrow b_c + k^*. \tag{6}$$

A **local rollback** is performed: T and Φ are reverted to the checkpoint saved after processing chunk $c-2$, and chunks $c-1$ and c are reparsed jointly. Checkpoints for the last two processed chunks are maintained, ensuring $O(1)$ rollback cost. The retry loop runs at most $T_{\max}$ times per chunk.

2.3 Adaptive Structural Masking and Hierarchical Prompting

To reconcile unbounded document structure with finite LLM context, we introduce adaptive compression and structured prompting.

2.3.1 Anchor Selection and Adaptive Masking

Given state $S^{(t)}$, we identify the **anchor structural node** $n_{\text{anchor}}^{(t)}$ as the deepest structural node whose span ends closest tobut does not exceedptr$^{(t)}$. If no such node exists (e.g., at initialization), we use the root node.

The masking function $\mathcal{M}$ produces a compressed tree $T_M^{(t)}$ by:

- Retaining the full **skeleton**: all structural nodes (with titles and levels) and parent-child relations.
- For content nodes: retain full text only for those within the anchors subtree and overlapping with recent context; replace others with `[CONTENT_MASK]`, preserving path π and role (if available).
- Collapsing distant subtrees (depth $>$ d_{max} and path distance $>$ δ from anchor) into a single `[SUBTREE_MASK]` token.

This achieves >60% context reduction while preserving navigational cues.

2.3.2 Hierarchical-Aware Path Embedding (HAPE)

To ground the LLM in global structure, we design a hierarchical prompting strategy integrating:

1. **Path descriptors**: natural-language paths (e.g., under Section 3.1: Data Collection in Chapter 3: Methodology).
2. **Relative positioning**: explicit sequential context (e.g., this follows immediately after Experimental Setup).
3. **Structured delimiters**: XML-style tags for robust component separation.

The full prompt template is in Appendix A. This anchors working memory to recent content while preserving global awareness.

2.4 Bijective Path-Content Mapping and Deterministic Recovery

To guarantee 100% content recall:

- **Recording**: When a content node is finalized-defined as having low uncertainty or being outside the active editing window (i.e., not in the last two chunks)its (π, span) is stored in Φ.
- **Recovery**: At output time, replace each `[CONTENT_MASK]` by querying Φ using π and extracting $D[\text{span}]$.
- **Version control**: During DBR rollbacks, Φ is reverted to match the active structural state, ensuring bijectivity is preserved throughout execution.

2.5 Complete Algorithm

The dominant cost is LLM inference, invoked $O(N \cdot T_{\max})$ times with context length bounded by $L_{\max} + C_{\text{struct}}$ ($C_{\text{struct}} < 512$ tokens empirically). DBRs semantic computation adds negligible overhead. In practice, DBR activates sparsely (<15% of chunks in our experiments), yielding near-linear scaling.

2.6 Future Work: Towards Enhanced Robustness

To further improve the robustness of ISCF against noisy inputs and implicit hierarchies, a natural extension is to integrate **structure-aware adversarial training** during the policy fine-tuning phase. This involves generating perturbed documents via transformations that either preserve (e.g., mild sentence reordering) or disrupt (e.g., injecting ambiguous topic shifts) the underlying structure. Training with a combined loss that enforces parsing consistency between original and perturbed documents would enhance the model's stability against real-world variations, making the iterative process more reliable. This constitutes the immediate next step of our research. Furthermore, exploring how such robustness-enhancing techniques can enable remarkable performance even with smaller-parameter models presents a compelling long-term direction for efficient, deployable document structuring systems.

3 Experiments

We conduct a comprehensive evaluation of the proposed ISCF framework against state-of-the-art baselines on challenging, real-world long-document datasets[1].

Our experiments are designed to answer the following research questions:

- **RQ1 (Overall Performance)**: How does ISCF perform compared to existing methods in terms of holistic structural parsing accuracy and content fidelity on long, unstructured documents ?
- **RQ2 (Component Ablation)**: What is the contribution of each core component (DBR, Adaptive Masking, Bijective Mapping) to the overall performance of ISCF ?
- **RQ3 (Robustness to Length)**: How does the performance of ISCF scale with increasing document length compared to baseline methods ?
- **RQ4 (Efficiency)**: What is the computational overhead of ISCFs iterative process and DBR mechanism, and how does it compare to other methods in terms of inference cost ?

3.1 Datasets

We evaluate on two long-document domainsfinance and lawboth characterized by implicit, semantically driven hierarchical structure.

- **FinDoc-Struct**: A collection of 500 annual reports from Chinese A-share listed companies (2011-2023), sourced from publicly disclosed scanned PDFs. After OCR processing, documents average $12,500 \pm 2,800$ sentences (∼300K tokens) and contain no reliable structural markup (e.g., section headers or XML/HTML tags). We manually annotated 50 high-quality OCR documents with a 5-level hierarchy (Part → Section → Subsection → Sub-subsection → Paragraph); the remaining 450 are used for qualitative and efficiency analysis.
- **LegalCase-Struct**: 200 Chinese court judgments (2000-2023) from public sources such as China Judgments Online[2], including decisions from the Supreme Peoples Court and higher peoples courts. Documents average $8,400 \pm 3,100$ sentences and loosely follow a standard legal structure (e.g., "Statement of Claims", "Facts", "Court's Reasoning",

[1] We will publicly release our data and code.

[2] https://wenshu.court.gov.cn

"Judgment"), though with significant formatting variation. Expert annotators labeled discourse units (e.g., "Plaintiff Alleges", "This Court Holds", "Judgment Order") and their hierarchical relationships in 30 representative cases.

Both datasets lack consistent explicit structural markers, requiring models to infer document hierarchy solely from semantic and contextual cues.

3.2 Baselines and Implementation Details

We compare ISCF against three representative categories of baseline methods for long-document processing. For all LLM-based methods, we use `DeepSeek-V3.2` as the backbone model due to its strong performance and extensive context window (128K tokens) (DeepSeek, 2024), ensuring fair comparison of algorithmic paradigms rather than raw model capabilitys(Chalkidis et al., 2022).

- **Vanilla Full-Context (FC)**: The entire document is truncated/padded to fit the model's maximum context window (we use 100K tokens) and a single prompt instructs the model to output the complete hierarchical structure. This serves as an upper-bound for what a "one-shot" long-context model can achieve, and directly tests the "lost-in-the-middle" phenomenon.
- **Fixed Chunking + Naive Merge (FCNM)**: The document is split into non-overlapping chunks of 800 tokens (approximately 40 sentences) using Sentence-BERT embeddings for semantic cohesion (Reimers and Gurevych, 2019). Each chunk is parsed independently into a local tree. A simple, rule-based post-processing step attempts to merge consecutive chunks if their highest-level nodes share high TF-IDF cosine similarity (> 0.7). This mimics common production pipelines.
- **Sliding Window with State Cache (SW-SC)**: Inspired by recent streaming approaches (Zhang et al., 2024), we use a sliding window of 800 tokens with a 200-token stride. The model is prompted to continue expanding the structure tree from the previous state, which is summarized in a cache limited to 300 tokens. This baseline tests incremental, stateful processing without our proposed DBR and advanced masking.
- **ISCF**: We implement ISCF as described in Section 2. Initial chunking uses Sentence-BERT embeddings (Reimers and Gurevych, 2019) with a target of 40 sentences. DBR parameters: uncertainty threshold $\tau = 1.2$ (selected via grid search on a validation set), trade-off $\gamma = 0.5$, search window $w = 15$, max retries $T_{\max} = 2$. The entropy threshold $\tau = 1.2$ corresponds to moderate uncertainty in role assignment (approximately uniform distribution over 3-4 roles). For adaptive masking, we retain the anchor node's subtree and collapse subtrees beyond depth $d_{\max} = 4$ or path distance $\delta = 5$.

3.3 Evaluation Metrics

We employ a multi-faceted evaluation protocol covering both structural fidelity and content preservation.

- **Tree Edit Distance (TED)** (Zhang and Shasha, 1989): Measures the minimum number of node insertions, deletions, and relabelings to transform the predicted tree into the gold tree. We use normalized TED: N-TED $=$ TED$/$(|Nodes_Gold| $+$ |Nodes_Pred|).
- **Role Classification F1 (R-F1)**: For content nodes, we classify their semantic role based on the path context. We report the macro-averaged F1 score against gold labels.
- **Content Recall (C-R) & Overlap (C-O)**: C-R is the proportion of gold sentences correctly assigned to any node. C-O measures precision by penalizing boundary duplication: $C\text{-}O = 1 - \frac{\#\text{duplicated sentences}}{\#\text{sentences predicted}}$. ISCF guarantees C-R=1.0 by design.
- **Boundary Coherence Score (BCS)**: For each predicted boundary between content nodes, we compute the cosine similarity between the embeddings of the last and first sentences of adjacent nodes (Reimers and Gurevych, 2019). Lower average similarity indicates better boundary placement (coherent units are not severed).

4 Results discussion

4.1 Overall Performance (RQ1)

We conducted the experiment multiple times and took the average as the final result. Table 1 presents the main results.

Table 1: Overall performance comparison. Best results are in **bold**. ISCF significantly outperforms all baselines.

(a) FinDoc-Struct

Method	N-TED ↓	R-F1 ↑	C-R ↑	C-O ↑
Vanilla FC	0.412	0.538	0.923	0.881
FCNM	0.521	0.462	0.894	0.812
SW-SC	0.368	0.602	0.981	0.942
ISCF (Ours)	**0.287**	**0.724**	**1.000**	**0.988**

(b) LegalCase-Struct

Method	N-TED ↓	R-F1 ↑
Vanilla FC	0.385	0.621
FCNM	0.503	0.498
SW-SC	0.341	0.685
ISCF (Ours)	**0.258**	**0.763**

Table 2: Ablation study on FinDoc-Struct.

Variant	N-TED ↓	R-F1 ↑	C-O ↑	BCS ↓
ISCF	**0.287**	**0.724**	**0.988**	**0.15**
– DBR	0.355	0.661	0.962	0.31
– AM†	0.332	0.643	0.990	0.18
– BM†	0.301	0.698	0.941	0.16
– HAPE	0.318	0.672	0.985	0.17

†AM = Adaptive Masking; BM = Bijective Mapping.

Table 3: Efficiency comparison.

Method	Calls ↓	Tok/Call ↓	Lat. (s) ↓
Vanilla FC	**1**	98,750	42.1
FCNM	15.6	1,200	38.5
SW-SC	48.3	1,050	112.7
ISCF (Ours)	22.4	**950**	**65.8**

Calls = Number of LLM calls; Tok/Call = Average number of tokens per LLM call; Lat. (s) = Average latency per document in seconds.

- ISCF achieves the best structural accuracy (lowest N-TED, highest R-F1), demonstrating the effectiveness of iterative, stateful editing with global awareness.
- ISCF achieves perfect Content Recall and near-perfect Content Overlap (0.988), validating the bijective mapping and DBR. In contrast, FCNM suffers from severe boundary duplication, and Vanilla FC loses content, evidencing the "lost-in-the-middle" effect (Liu et al., 2023).
- SW-SC performs second best but is outperformed by ISCF, highlighting the benefits of DBR and adaptive masking over fixed-stride sliding windows.

4.2 Ablation Study (RQ2)

Table 2 shows the contribution of each component on FinDoc-Struct.

- **DBR** is critical for local coherence, causing the largest N-TED degradation and BCS increase when removed.
- **Adaptive Masking** is essential for balancing global structure and local content, as its removal hurts R-F1.
- **Bijective Mapping** ensures precision, with heuristic merge leading to content overlap drop.
- **HAPE** improves structural awareness, as flattened context reduces performance.

4.3 Scalability with Document Length (RQ3)

We analyze performance by binning FinDoc-Struct documents by sentence count. ISCF's N-TED remains stable across length bins, while SW-SC's degrades noticeably beyond 10,000 sentences. This demonstrates ISCF's robustness, stemming from adaptive masking (managing global skeleton size) and DBR (localizing boundary issues).

4.4 Efficiency Analysis (RQ4)

Table 3 reports computational cost on the full FinDoc-Struct dataset.

- ISCF requires fewer calls than SW-SC due to its non-overlapping, stateful iteration.
- ISCF uses the smallest context per call (**950 tokens**) thanks to adaptive masking, reducing per-step cost.
- Overall latency offers a favorable accuracy-efficiency trade-off. DBR retries added minimal overhead ($< 10\%$ of total time).

4.5 Conclusion of Experiments

The results comprehensively validate ISCF. It significantly outperforms strong baselines in structural accuracy and content fidelity on very long

documents, with each technical component contributing measurably. ISCF offers a practical and principled solution for transforming unbounded unstructured text into precise hierarchical representations.

5 Related Work

Our work on iterative structure parsing relates to several key areas in NLP.

5.1 Long-Context Modeling for LLMs.

The predominant challenge of processing lengthy documents stems from the fixed context window of Transformer-based LLMs. To address this, early research focused on sparse attention mechanisms, such as the sliding window pattern in Longformer (Beltagy et al., 2020) or the global+local attention in BigBird (Zaheer et al., 2020), which reduce computational complexity. Recent advancements like StreamingLLM (Xiao et al., 2024) stabilize attention for infinite-length inputs. However, these methods primarily alleviate computational constraints and do not inherently solve the *semantic* challenge of information integration across distant text segments, a phenomenon characterized as the lost-in-the-middle problem (Liu et al., 2023). Our ISCF framework directly tackles this by adopting an iterative, stateful paradigm, ensuring each parsing decision is informed by a compressed global structural memory, thus bypassing the need for unreliable full-context ingestion.

5.2 Document Structure and Hierarchy Parsing.

Parsing documents into a hierarchical format is a well-studied task. Traditional approaches relied on rule-based systems or sequence labeling models using layout features (Saijad et al., 2022). With the rise of LLMs, the task is often framed as a one-shot sequence-to-structure generation problem, using extensive prompting or fine-tuning (Dong and Lapata, 2016). A common production baseline involves processing fixed, independent chunks and merging results heuristically, which often leads to fractured structures at boundaries. Recent work like LumberChunker (Duarte et al., 2024) improves segmentation by considering semantic cohesion. ISCF fundamentally departs from these static or independent-chunk approaches by modeling parsing as a *stateful, iterative editing process*. This allows for dynamic correction and integration, ensuring global coherence akin to how an editor builds an outline incrementally.

5.3 Iterative and Stateful NLP Tasks.

The paradigm of iterative refinement is explored in related tasks. In long-form question answering, agents may iteratively gather evidence (Gao et al., 2023). In summarization, methods like those in (Zhang et al., 2024) process text in segments while maintaining a running summary. Our work adapts this iterative philosophy to the novel domain of hierarchical structure construction. Unlike simple sliding-window baselines that pass a limited text cache, ISCF actively maintains and refines an explicit, compressed structural skeleton, integrating novel techniques like uncertainty-driven boundary re-alignment (DBR) to manage state across chunk transitions effectively.

6 Conclusion and Outlook

This work addresses the critical challenge of parsing long, unstructured documents into precise hierarchical trees by proposing the **Iterative Structured Content Filling (ISCF)** framework. We reframe the task from a one-shot transduction to a stateful, iterative editing process, addressing fundamental tensions between document length and LLM context limits, and between local processing and global coherence. ISCF integrates three synergistic innovations: **Dynamic Boundary Realignment** optimizes chunk boundaries for semantic continuity; **Adaptive Structural Masking with HAPE** maintains global structural awareness within finite context; and **Bijective Path-Content Mapping** guarantees lossless text reconstruction. Comprehensive experiments on financial and legal documents demonstrate ISCF's superior structural accuracy and content fidelity over strong baselines, while maintaining favorable efficiency. This framework provides a principled, effective solution for transforming unbounded unstructured text into machine-readable hierarchical representations. Future work will focus on enhancing robustness to noisy inputs, extending to graph structures, optimizing efficiency, and testing broader domain generalization.

7 Limitations

ISCF provides a robust framework through the co-design of segmentation, context management, and state tracking. Limitations include: 1) Depen-

dence on initial sentence segmentation quality. 2) Assumption of tree structures; extending to graphs (DAGs) is future work. 3) Self-consistency check for DBR adds cost; a lightweight validator could improve efficiency.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2022. Lexglue: A benchmark dataset for legal language understanding in english. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*. Introduces a comprehensive benchmark for evaluating NLP systems on legal document understanding tasks, highlighting the need for hierarchical and long-context modeling in legal digitization.

DeepSeek. 2024. DeepSeek-V3 technical report. *Preprint*, arXiv:2412.19437.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longbench: A bilingual benchmark for long context understanding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. Provides empirical evidence on the inconsistent performance of LLMs on long-context tasks.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 33–43. Pioneering work on neural semantic parsing using sequence-to-tree generation via stack-based decoding.

André V. Duarte, João Marques, Miguel Graça, Manuel F. Freire, Lei Li, and Arlindo L. Oliveira. 2024. Lumberchunker: Long-form narrative document segmentation. In *arXiv preprint*. ArXiv:2406.17526; proposes a method for segmenting long narrative documents.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Haofen Wang, and Meng Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

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. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. The Lost in the Middle phenomenon is crucial for explaining the failure of Full-Text Ingestion.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Standard method for semantic sentence similarity, used in DBR.

Muhammad Saijad and 1 others. 2022. Parsing documents with guidance of layout and semantics. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*. Introduced the encoder-decoder RNN architecture for mapping sequences to sequences.

Shu Wang, Yingli Zhou, and Yixiang Fang. 2025. Bookrag: A hierarchical structure-aware index-based approach for retrieval-augmented generation on complex documents. *arXiv preprint arXiv:2512.03413*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. Presents a multimodal pre-trained model combining text and layout information, facilitating the conversion of scanned financial or legal documents into structured, machine-readable formats.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and 1 others. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262.

Yifan Zhang, Kento Sone, and Alexandre Alahi. 2024. Streaming llm: Efficient inference with fixed-size attention windows. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, volume 235 of *Proceedings of Machine Learning Research*. PMLR.

Appendix A Prompt Template

The complete structured prompt used in our Hierarchical-Aware Path Embedding (HAPE) framework is as follows:

```
<System Message>
You are the parsing engine for the Iterative Structured Content Filling (ISCF) framework.
You are at iteration t={t}, having processed {ptr} sentences so far (out of {total} total).
Your task is to extend the current partial structure by processing the next chunk of text.
</System Message>

<Global Structure Context>
Below is the current document tree structure, compressed for context efficiency:
- Nodes marked with [CONTENT_MASK] have their full text omitted but retain their hierarchical position.
- Subtrees marked with [SUBTREE_MASK] are collapsed as they are distant from the current focus.

{Indented representation of T_M^{(t)}}
  Example format:
  root (title: "Document Root")
    Chapter 1: Introduction (level: 1)
      Section 1.1: Background (level: 2)
        [CONTENT_MASK] (path: root.chapter1.section1.1, role: context-setting, span: 0-42)
      Section 1.2: Problem Statement (level: 2) [ANCHOR NODE]
        SentenceSpan_1 (role: definition, span: 43-58)
        [CONTENT_MASK] (path: root.chapter1.section1.2.s2, role: argument, span: 59-72)
    Chapter 2: Literature Review (level: 1)
      [SUBTREE_MASK] (depth: 3, distance from anchor: 3)
</Global Structure Context>

<Sequential Context>
The text you are about to process IMMEDIATELY FOLLOWS the content of the anchor node:
**Anchor Node**: {n_anchor.title} (level: {n_anchor.level})
**Anchor Path**: {n_anchor.path_descriptor}
**Last Processed Content**:
{If anchor has assigned content: [excerpt of last 2--3 sentences]; otherwise: (no content assigned yet)}
</Sequential Context>

<Current Processing Chunk>
You are now processing sentences {chunk_start} to {chunk_end} (absolute indices in document):
{C_i}
</Current Processing Chunk>

<Task Instructions>
Based on the global structure and sequential context above:
1. Identify if the new text continues existing structural nodes or begins new ones.
2. Create new structural nodes with appropriate titles and levels when needed.
3. Assign each contiguous sentence span in the new chunk to:
   - An existing structural node (specify by path), OR
   - A newly created structural node.
4. For each assigned content span, label its semantic role (topic, definition, assumption, method, evidence, result, conclusion, example).
5. Output a JSON edit script with the following schema:
   {
     "new_structural_nodes": [
       {"path": "root.chapter1.section1.3", "title": "Research Questions", "level": 2}
     ],
     "content_assignments": [
       {"span": [73, 85], "parent_path": "root.chapter1.section1.2", "role": "example"},
       {"span": [86, 94], "parent_path": "root.chapter1.section1.3", "role": "question"}
     ],
     "metadata": {
       "chunk_processed": {chunk_id},
       "estimated_coherence": null  # Will be computed externally via DBR
     }
   }
</Task Instructions>

<Constraints>
- Maintain bijective mapping: each sentence must be assigned to exactly one content node.
- Ensure hierarchical consistency: child node spans must be within parent node spans.
- Titles should be concise (38 words) and descriptive.
- DO NOT generate new content or modify existing node titles/spans.
- DO NOT output anything outside the specified JSON schema.
</Constraints>
```