M3DocDep: Multi-modal, Multi-page, Multi-document Dependency Chunking with Large Vision-Language Models

Supplementary Material

A. Datasets and Pre-processing Details

All datasets used in our experiments are publicly available research benchmarks. We rely exclusively on open corpora for both hierarchy parsing and RAG-based VQA, and do not use any proprietary or private documents.

Dataset	Type/Domain	#Documents	Avg. Pages	#QA pairs
DocHieNet	Mixed (Reports/Papers/Industrial)	1,673	5.3	–
HRDH	Academic Papers (arXiv, diverse layouts)	1,500	7.1	–
HRDS	Academic Papers (ACL Anthology, single template)	1,000	10.4	–
MP-DocVQA	General Documents (Multi-page)	17,000	3.4	48,000+
CUAD	Legal Documents (Contracts)	510	6.2	13,000+
DUDE	Mixed (Financial Reports/Manuals)	3,000+	4.9	7,000+
MOAMOB	Industrial Technical Documents	2	35.5	71

Table 1. Summary of key datasets used in our experiments. DocHieNet, HRDH, and HRDS are used for document hierarchy parsing, while MP-DocVQA, CUAD, DUDE, and MOAMOB are used for multi-page QA.

A.1. Document Hierarchical Parsing Corpora

We train and evaluate M3DocDep on three document hierarchical parsing benchmarks: DocHieNet, HRDH, and HRDS. These corpora collectively cover a wide range of industrial document types, including scanned reports, born-digital PDFs, and documents with complex multi-column layouts, which match the intended deployment scenarios for industrial RAG.

DocHieNet. DocHieNet [28] contains 1,673 documents from multiple domains (technical reports, scientific papers, industrial documents), with many multi-page scanned PDFs. Each document is annotated with block-level types (titles, section headers, paragraphs, tables, figures, captions, etc.) and parent–child relations between blocks, forming a ground-truth hierarchy tree. Most documents are English or Chinese.

HRDH and HRDS. HRDH and HRDS [16] are two subsets of HRDoc with different layout characteristics. HRDS contains 1,000 ACL Anthology papers with a nearly identical template, offering a clean and homogeneous setting. HRDH includes 1,500 arXiv papers with highly diverse layouts across many research domains, making it a harder and more realistic benchmark. Both provide block-level parent annotations for hierarchy supervision.

Table 2 summarizes the document- and page-level coverage for the three DHP corpus, including the proportion of blocks with parents, and the fraction of intra-page vs. cross-page edges. These statistics highlight that a large portion of relations are cross-page, particularly in HRDH/DocHieNet, making global tree reconstruction crucial for realistic DHP.

Dataset	Documents (multi-page)	GT page coverage (test)	Parent–child summary (test)
DocHieNet	161 (100%)	**Pages w/ GT** (avg / med / min / max): 14.205 / 9.000 / 3 / 50 **Pages total** (avg / med / min / max): 14.354 / 9.000 / 3 / 50 **Coverage** (avg / med / min / max): 0.994 / 1.000 / 0.727 / 1.000	**Valid blocks:** 29,401 **Parents:** 24,050 (81.80%) **Edges:** intra 59.36%, cross 40.64% **ROOT:** 18.20%
HRDH	500 (100%)	**Pages w/ GT** (avg / med / min / max): 14.080 / 12.000 / 2 / 35 **Pages total** (avg / med / min / max): 14.086 / 12.000 / 2 / 35 **Coverage** (avg / med / min / max): 0.999 / 1.000 / 0.800 / 1.000	**Valid blocks:** 77,954 **Parents:** 56,455 (72.42%) **Edges:** intra 50.58%, cross 49.42% **ROOT:** 27.58%
HRDS	100 (100%)	**Pages w/ GT** (avg / med / min / max): 10.380 / 10.500 / 5 / 19 **Pages total** (avg / med / min / max): 10.380 / 10.500 / 5 / 19 **Coverage** (avg / med / min / max): 1.000 / 1.000 / 1.000 / 1.000	**Valid blocks:** 15,584 **Parents:** 11,156 (71.59%) **Edges:** intra 67.27%, cross 32.73% **ROOT:** 28.41%

Table 2. Summary of page coverage and parent–child relations in DocHieNet, HRDH, and HRDS, based on the test set.

A.2. RAG-based VQA Corpora

For RAG-based evaluation we consider four multi-page VQA corpus: DUDE, MP-DocVQA, CUAD, and MOAMOB. They span financial reports, contracts, scanned forms, and complex industrial documentation.

DUDE. DUDE [13] includes more than 3,000 documents such as annual reports and technical manuals, with over 7,000 QA pairs. The official test-set answers are hidden on a server; thus we use the validation split for retrieval metrics and report ANLS on the official test server in the main paper.

MP-DocVQA. MP-DocVQA [24] contains roughly 17,000 multi-page documents with more than 48,000 questions (around 2.8 questions per document). Documents include various scanned and born-digital government and industrial reports with heterogeneous layouts.

CUAD. CUAD [10] consists of 510 legal contracts with over 13,000 QA pairs, targeting specific clauses and legal concepts. We follow prior work and use the official test split (about 50 documents, 1,200 QA) for evaluation, indexing all test documents jointly.

MOAMOB. MOAMOB [11] is a small-scale but challenging dataset with two long industrial documents in

Korean and 71 QA pairs about predictive maintenance. The questions often require cross-page reasoning and fine-grained reference to operational guidelines, making it a stress test for structure-aware chunking and retrieval.

A.3. Pre-processing Pipeline

All documents are processed through the SharedDet (DP+OCR) pipeline described in Sec.3.1 of the main paper. Here we detail design choices and hyperparameters.

Document Parsing (DP). We use detectors (DETR, DiT, VGT) trained on DocLayNet [18] to detect layout blocks (titles, headers, paragraphs, tables, figures, etc.) per page. For each detector we fix: (i) a confidence threshold τ_{det}, (ii) a NMS IoU threshold τ_{nms}, and (iii) a per-page upper bound $K_{\max}$ on block count. These are tuned once on a held-out validation subset and reused for all experiments, ensuring that all methods (M3DocDep and baselines) operate on the same block set.

OCR. Each detected block is independently passed to an OCR engine (Tesseract, EasyOCR, or TrOCR) depending on the experiment. We use the default English models, add Korean for MOAMOB, and include Chinese OCR models for DocHieNet documents containing Chinese text. All extracted text is normalized by lowercasing, removing control characters, and collapsing whitespace, and is then associated with each block's bounding box and page index.

Global Document Blocks. Bounding boxes are mapped into a global normalized coordinate frame $[0,1]^2 \times [0,1]^2$ using the original page sizes and page indices, yielding the Global Document Blocks

$$V = \{(\text{bbox}_i, \text{type}_i, \text{text}_i, t(i))\}_i.$$

These blocks form a stable, detector-and-OCR-agnostic canvas for both M3DocDep and all tree-aware/text-based baselines.

A.4. Additional Statistics for DHP Corpora

Table 2 provides a more detailed view of page coverage and parent–child relations in DocHieNet, HRDH, and HRDS, which is useful when analyzing cross-page dependencies and ROOT frequency.

B. Metric Definitions and Evaluation Protocol

B.1. Hierarchy Metrics

For DHP, we report (i) parent prediction F1 and (ii) STEDS [17]. Parent F1 is computed over all non-ROOT blocks, treating each predicted parent as a single-label classification target. STEDS follows the original definition and measures tree-level edit distance between predicted and ground-truth hierarchies.

B.2. Retrieval Metrics

For each question, we treat a chunk as relevant if it contains the gold answer span (or its annotated supporting block). Precision@k, Recall@k, and nDCG@k [12] are computed at the question level and macro-averaged over all questions.

B.3. QA Metrics

We compute ANLS [4], ROUGE-L [14], and METEOR [3] on normalized answers (lowercased, extra whitespace removed, punctuation stripped). For ANLS we follow the official MP-DocVQA evaluation, using character-level Levenshtein distance.

C. Baseline Implementations

Unless otherwise noted, all methods follow the common RAG setup in Sec. E; only DHP and chunking components differ. All baselines considered in this work are instantiated from publicly available implementations or public APIs; we do not rely on any internal or non-releasable systems. Whenever possible, we use the official code released by the original authors, and otherwise provide faithful re-implementations that follow their published descriptions.

C.1. Document Hierarchical Parsing (DHP) Baselines

For DHP baselines (DocParser, DSG, DSPS, DSHP-LLM, Qwen2.5-VL–DHP–SFT), we either use official implementations or faithful re-implementations following their papers.

DocParser. DocParser [20] is a pioneering DHP method that converts a flat list of layout elements into hierarchical relations using hand-crafted heuristics. It explicitly considers multi-column layouts and geometric cues such as indentation, relative position, and spacing, but largely ignores richer meta-information such as the actual text content of elements. As a result, DocParser is effective on clean, well-structured layouts but struggles on noisy scans and long documents where semantic signals are crucial.

DSG. DSG [21] replaces heuristic rules with an end-to-end neural relation predictor. It leverages a bidirectional LSTM to model relations between layout elements, using visual features extracted from an FPN backbone for image regions and GloVe word embeddings for layout element types. Compared to DocParser, DSG better captures local context among nearby blocks, but it is still limited by its reliance on relatively shallow sequence modeling and can degrade on highly irregular or domain-shifted layouts.

DSPS. DSPS [16] is the baseline introduced with the HRDoc dataset. It employs a multi-modal encoder and a GRU decoder for hierarchical organization. Textual embeddings of layout elements are extracted separately and fused with geometric and visual features inside the encoder. The decoder then predicts parent–child relations in an autoregressive manner. This design improves robustness by jointly modeling text and layout, but the GRU-based decoding and local decision process make it difficult to enforce a globally optimal tree, especially across multiple pages.

DSHP-LLM. DSHP-LLM [23] is an LLM-based DHP model that takes as input a textualized representation of each document, including block indices, layout types, and (optionally) truncated text. A fine-tuned instruction-following LLM (e.g., Mistral-8B) is prompted to output, for every block, the identifier of its parent or a special ROOT symbol. This approach benefits from strong long-context reasoning and flexible natural language prompting, but remains sensitive to prompt design, context-window limits, and instability in sequence generation, especially for cross-page links and complex layouts.

Qwen2.5-VL–DHP–SFT. Qwen2.5-VL–DHP–SFT adapts the DSHP-LLM idea to a multi-modal LVLM backbone. It retains the decoder-style sequence generation objective (predicting parent identifiers from textualized blocks) while allowing the model to access visual cues through image embeddings. In our experiments it serves as a strong SFT-only baseline: Qwen2.5-VL– DHP–SFT uses the same training data and prompts as DSHP-LLM but does *not* attach a biaffine head or perform MST-based decoding. This highlights the contribution of explicit dependency scoring and global tree constraints in M3DocDep.

Training and evaluation protocol. For all DHP baselines, we follow the official train/test splits of DocHieNet, HRDH, and HRDS and adopt the hyperparameters recommended in the original papers. In particular, DocParser, DSG, and DSPS are run with their official public implementations and default settings, and DSHP-LLM and Qwen2.5-VL–DHP–SFT are trained with the same learning rates, batch sizes, and optimization schedules reported in their respective works. As in DocHieNet [29] and MultiDocFusion [23], supervision and evaluation are defined at the block level: each annotated block is treated as a node with a single parent (or ROOT), and models are trained and evaluated by predicting the parent of every non-ROOT block. In Table 1 of the main paper, the tree-aware baselines and M3DocDep are evaluated on the same SharedDet blocks, where they consume the same Global Document Blocks produced by SharedDet. The general-purpose LVLM rows are included as reference baselines rather than part of this shared-block comparison. This isolates hierarchy recovery from differences in document parsing and OCR for the tree-aware methods.

Method	**Key hyperparameters**	**Notes**
Length	window=550 token	only text, no structure
Semantic	base encoder=E5	sentence-level clustering
LumberChunker	backbone=Mistral-8B	topic-shift prompts
Perplexity	backbone=Mistral-8B, perplexity window tuned	Meta-Chunking
Structure-based	uses DP types	Layout based chunks
MultiDocFusion	uses DHP tree, max_len=550 tok	tree-based chunks

Table 3. Typical configuration of chunking baselines. Exact values per dataset are provided in the released configs.

C.2. Chunking Baselines

This subsection provides comprehensive descriptions of the chunking methodologies compared against our proposed tree-based structure-aware dependency chunking in **M3DocDep**. Each chunking method is illustrated with examples in Table 12.

Length chunking [9] This method divides documents into chunks based on a fixed token length limit. Each chunk is created uniformly, without considering semantic or structural boundaries. While simple and computationally efficient, it risks splitting important contexts, leading to potential information loss and degraded performance in retrieval and QA tasks.

Semantic chunking [19] Semantic chunking leverages encoder-based language models to maintain semantic consistency. Chunks are formed by grouping sentences based on semantic similarity scores derived from language models (e.g., E5 embeddings). Although effective in maintaining semantic coherence, it tends to produce shorter, numerous chunks, potentially impacting retrieval efficiency. Following prior work [11], we employed the E5 model for consistency in our experiments.

LumberChunker [7] LumberChunker employs Large Language Models (LLMs) to dynamically partition documents by identifying topical shifts between sentences or paragraphs. It effectively captures the semantic independence of textual segments, resulting in chunks of variable sizes optimized for dense retrieval tasks. For experimental consistency across LLM-based methods, we employed the Mistral-8B model as the base model.

	MultiDocFusion	M3DocDep
Hierarchy recovery	LLM-based hierarchical parsing	LVLM embeddings + MST global constraint
Visual handling	Absorbed into OCR text	Preserves table/figure crops + captions
Chunking signal	Structure → chunk	Multimodal tree recovery → indexing signal

Table 4. Conceptual comparison between MultiDocFusion and M3DocDep. The main difference is not merely stronger supervision, but explicit multimodal dependency recovery with globally constrained tree decoding.

Perplexity chunking [31] Based on the concept of Meta-Chunking, Perplexity chunking identifies optimal chunk boundaries by analyzing the perplexity distribution of sentences and paragraphs. It dynamically merges or splits textual segments at a fine-grained level, effectively balancing granularity and computational efficiency. To ensure fairness among LLM-based methods, we also used the Mistral-8B model for these experiments.

Structure-based Chunking This approach partitions documents solely based on their structural layouts, such as section headers, tables, and figures. Similar methodologies have been explored in recent works [25, 30]. In our experiments, Structure-based Chunking served as a baseline to clearly isolate and demonstrate the impact of stronger hierarchical parsers. Specifically, chunks were created by ordering structural elements obtained via DP (Document Parsing), without explicitly considering hierarchical parent–child relationships identified by DSHP-LLM or M3DocDep. Segment types were included in the resulting chunks.

MultiDocFusion MultiDocFusion [23] is a prior hybrid multi-modal chunking pipeline that integrates hierarchical document structure into the chunking process. It utilizes a DSHP-LLM model (fine-tuned Mistral-8B) identified in previous work to explicitly reconstruct section hierarchies and then performs rule-based fusion of hierarchy-aware segments into chunks. This significantly enhances the semantic and structural coherence of chunks compared to purely text- or layout-based baselines and serves as a strong structure-aware chunking baseline in our experiments. M3DocDep further improves upon this line of work by replacing LLM-only hierarchy prediction with LVLM-based dependency scoring and MST-based global tree decoding, yielding more stable trees and more boundary-faithful chunks.

C.3. Chunking Configuration

Table 3 summarizes the hyperparameters and design choices for all chunking baselines used in our experiments. Each method follows its original formulation, but all chunkers operate on the same Global Document Blocks produced by SharedDet, use the same maximum chunk length (550 tokens), and are evaluated under the same corpus-level retrieval setting. This shared protocol is intended to isolate the effect of chunk construction rather than differences in parsing, chunk budget, or retrieval setup.

D. Implementation Details of M3DocDep

This section expands on Sec. 3 of the main paper by detailing each component of M3DocDep and its training configuration. Figures 1 and 2 also include corresponding real-world examples to illustrate each step of the pipeline.

D.1. SharedDet (DP+OCR)

We reuse the SharedDet pipeline defined in Sec. A.3 to produce *Global Document Blocks*. In the rest of this section we focus on how these blocks are consumed by M3DocDep (embedding, parsing, and chunking).

D.2. LVLM-based Multi-modal Block Embedding

M3DocDep uses frozen LVLM encoders to extract page-level visual tokens and aggregates them into block embeddings via SoftROI (see the list of backbones in Sec. E).

Page Multi-modal Tokens. Each page image is fed into a frozen LVLM. From the last decoder layer we extract hidden states at positions corresponding to visual tokens. Using token-grid metadata, we map each token to 2D coordinates in the global document frame $[0,1]^2$, yielding a set of tokens $\{z_p\}$ per page with coordinates (u_p, v_p).

SoftROI pooling. For each block i with normalized box bbox_i, the SoftROI Embedder collects tokens whose coordinates lie inside the box and assigns them boundary-aware weights:

$$w_p \propto [u_p(1-u_p)]^\alpha [v_p(1-v_p)]^\alpha, \quad \tilde{w}_p = \frac{w_p}{\sum_{q\in \mathrm{ROI}_i} w_q},$$

where α controls boundary sharpness. The SoftROI Multi-modal Block Embedding is

$$e_i = \sum_{p\in \mathrm{ROI}_i} \tilde{w}_p z_p.$$

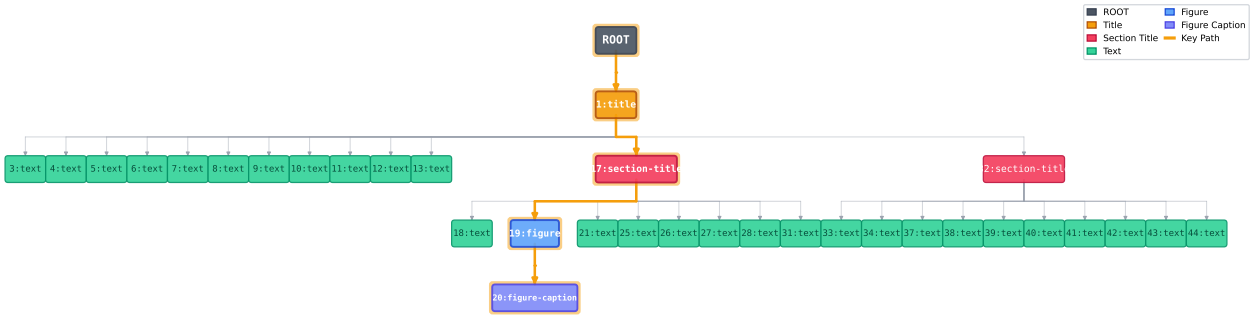
Compared to uniform pooling, this applies a spatially-aware weighting that downweights tokens near box edges and corners, making the embedding more robust to box jitter and imperfect detections.

Type-aware embeddings. Block types (title, section header, paragraph, table, figure, caption, other) are mapped to a small embedding table τ_i. We concatenate SoftROI embeddings and type embeddings, $x_i = [e_i; \tau_i]$, and pass them through a two-layer MLP to obtain hidden representations h_i used for dependency scoring.

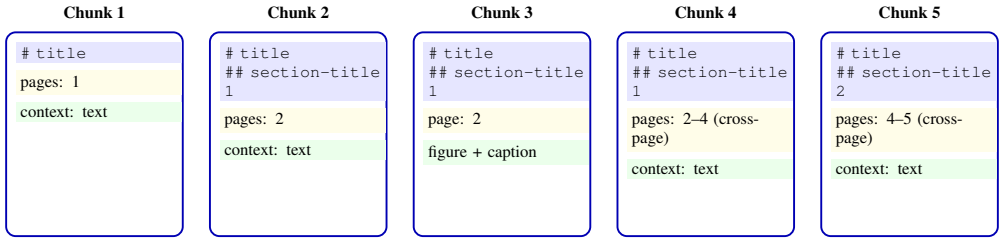
(a) **Multi-page document** — the original 5-page industrial guideline before any processing.

(b) **SharedDet(DP + OCR)** — page-level layout regions detected by DP and linked with OCR text to form Global Document Blocks.

Figure 1. Structure-aware pipeline (part 1): (a) raw 5-page document and (b) DP + OCR outputs used to construct Global Document Blocks.



(c) **Global Document Dependency Tree** — section headers, paragraphs, tables, and figures organized into a hierarchical tree with parent–child links.



(d) **Tree-based Structure-Aware Dependency Chunking** — five example chunks. Blue: section path, yellow: metadata, green: context type.

Figure 2. Structure-aware pipeline (part 2): (c) Global Document Dependency Tree reconstruction and (d) Tree-based Structure-Aware Dependency Chunking.

D.3. Global Document Dependency Parsing

We adopt a biaffine dependency scoring head to score parent–child relations between blocks and recover a global document tree.

Candidate parent selection. For each child block v, we construct a small candidate parent set $P(v)$ by:

- prioritizing title and section-header blocks;
- allowing "upward" links on the same page within a verti-

cal tolerance;
- allowing cross-page parents only within the most recent M pages;
- discarding implausible parents based on type constraints (e.g., paragraphs rarely parent top-level titles).

We keep at most K candidates per child based on a header- and distance-based heuristic; (M, K) are tuned on a validation subset and fixed across datasets.

Biaffine scoring. Given hidden states h_i, we score edges $u \to v$ using a biaffine function with geometric features:

$$s(u \to v) = [h_u; 1]^\top U [h_v; 1] + w_{\text{geo}}^\top \delta_g(u, v),$$

where $\delta_g(u, v)$ includes relative offsets, size ratios, page-distance, and overlap indicators. The virtual root score is defined as $s(\text{ROOT} \to v) = r^\top h_v + b_r$.

For each child v we normalize scores over $P(v) \cup \{\text{ROOT}\}$ with a $(K{+}1)$ child-softmax and minimize cross-entropy against the ground-truth parent.

MST-based global tree decoder. At inference time, we treat edge scores as weights and feed them into the Chu–Liu/Edmonds algorithm [6, 8] to obtain the maximum spanning arborescence, enforcing single-root, single-parent, and acyclicity constraints. We also measure a local argmax baseline that chooses the best-scoring parent per child without global constraints.

D.4. Tree-based Structure-Aware Dependency Chunking

Given the Global Document Dependency Tree, M3DocDep first applies a DFS-based dependency chunking procedure over the tree, which is summarized in Algorithm 1. Concrete examples of the resulting chunks are illustrated in Figure 2.

Section subtree grouping. We mark title and section-header blocks as section roots and perform DFS from each root to collect descendant blocks into section subtrees. Blocks in the same subtree are merged across page boundaries, so a chunk can span multiple pages when a section continues.

Figure/table–caption binding. For blocks labeled as figures or tables, we exploit tree structure: if a figure/table and a caption form a parent–child (or closely related) relation, they are forced into the same chunk. When no explicit edge exists, we fall back to spatial proximity and reading order heuristics on the page.

Algorithm 1 Tree-based Structure-Aware Dependency Chunking

Require: Global dependency tree $G = (V, E)$, max chunk length `max_len`

```
function BUILDCHUNKS(root)
    chunks ← [ ]
    DFSCHUNK(root, [], "", chunks)
    return chunks
end function
function DFSCHUNK(v, path, buffer, chunks)
    path ← path + [v]
    buffer ← buffer + text(v)
    if length(buffer) > max_len then
        split buffer into one or more chunks and append to chunks
        reset buffer to last partial chunk
    end if
    for child ∈ children(v) in doc order do
        DFSCHUNK(child, path, buffer, chunks)
    end for
end function
```

Section path and metadata. Each chunk is annotated with: (i) section path from root to governing header, (ii) page range, and (iii) constituent block IDs and layout types. This metadata is stored in the retrieval index and used during RAG to present structure-aware context to the LVLM.

Logical consistency. Because the dependency parser decodes a maximum spanning tree, every non-root block receives exactly one parent and the recovered structure is globally acyclic. The chunker therefore operates on a valid tree rather than on a set of independently chosen local links, which makes figure–caption binding and cross-page section grouping more stable.

Granularity control. Chunk granularity is adjusted deterministically on the recovered tree by changing the maximum chunk length and the cut policy. In practice, this allows section-level, paragraph-level, or finer chunking without retraining the dependency parser: coarse settings keep larger subtrees intact, while finer settings cut earlier along long paths or large sibling groups.

D.5. Training Hyperparameters

We train only the dependency head while keeping the LVLM encoders frozen, as described in Secs. D.1–D.4 above. The main hyperparameters are summarized in Table 5.

D.6. Training and Inference Environment

All experiments are conducted on a GPU cluster equipped with 8 NVIDIA A100-SXM4-80GB GPUs (80 GB VRAM each), 64-core CPUs, and 1 TB of system RAM. We implement M3DocDep in PyTorch (v2.2) with CUDA (v12.9), and use FAISS (v1.8) to build dense retrieval indices. Training the dependency head on HRDH for 3 epochs takes about

Component	Hyperparameters (default / range)
Dependency head	Learning rate: 1×10^{-5}–5×10^{-5} Optimizer: Adam / AdamW Batch size: 8–16 docs/GPU Epochs: 3–5 (early stop) Dropout: 0.0–0.1 (MLP layers) Weight decay: 0–10^{-2} Parent window M: recent pages (tuned on val) Candidate top-K: $\{8, 16, 32\}$ (ablations)

Table 5. Summary of key hyperparameters used to train the M3DocDep dependency head. Ranges denote grid/line searches; defaults follow the settings used for the main tables.

Setting	Configuration
Corpus	DUDE, MP-DocVQA, CUAD, MOAMOB
Index granularity	Chunk-level, corpus-level index
Retriever (dense)	BGE, E5, MM-Embed
Retriever (sparse)	BM25
k_{ret}	$\{1, 2, 3, 4\}$
Reader LVLMs	LLaVA-OneVision-1.5, InternVL-3.5, Qwen2.5-VL
Metrics (retrieval)	Recall, Precision, nDCG
Metrics (QA)	ANLS, ROUGE-L, METEOR
Prompting	Instruction + query + serialized chunk list
Decoding	Fixed temperature/top-p per LVLM

Table 6. Summary of RAG configuration used in our experiments.

6 hours on a single A100 GPU, including data loading and evaluation. For the per-page runtime numbers in Table 11, we use a single A100 (80GB) and observe 27 GB peak GPU memory during end-to-end indexing. Corpus-level indexing for DUDE, MP-DocVQA, CUAD, and MOAMOB requires approximately 1–3 hours per corpus, while full QA evaluation takes an additional 2–4 hours per corpus.

E. LVLM and RAG Setup

This section details the LVLM backbones, retrieval pipeline, and prompting strategy used across all experiments. Table 6 summarizes the full RAG configuration.

E.1. LVLM Backbone Configurations

We use three open-source LVLM backbones: LLaVA-OneVision-1.5 [1], InternVL-3.5 [27], and Qwen2.5-VL [2]. For all of them:

- Input pages are rendered at a shared resolution and fed either one page per call or in small page batches, depending on the model's context window.
- We fix the maximum number of images per call and slice long documents across multiple calls if needed.
- Decoding parameters (temperature, top-p, `max_new_tokens`) are kept identical across all chunking methods within each experiment.

For GPT-5 and other closed LVLM baselines, we record and report the provider, model identifier, API mode/endpoint, access date, prompt template, and decoding parameters (temperature, top-p, and output-token limit) used in each experiment. These baselines are used only for comparison; M3DocDep itself does not rely on closed models.

E.2. Retrieval Pipeline

All chunking methods share the same retrieval backbone.

Dense and sparse retrieval. We use dense embedding models such as BGE [5], E5 [26], and MM-Embed [15] to obtain chunk-level representations, storing them in a FAISS-based ANN index. For sparse retrieval, we use BM25 [22] over chunk texts.

Corpus-level top-k_{ret}. For each query we retrieve the top $k_{\text{ret}} \in \{1, 2, 3, 4\}$ chunks from the corpus-level index. Unless otherwise specified, $k_{\text{ret}} = 4$ is used in the main tables, while extended experiments sweep over k_{ret} to analyze sensitivity.

Chunk serialization and reader input. Each chunk is serialized with a shared schema consisting of section path, page range, block-type markers, and OCR/caption text; fields unavailable to a given chunker are left blank or omitted. For figure/table chunks, the associated caption is kept in the same serialized unit so that retrieval preserves the figure–caption relation. Text-only retrievers (BGE, E5, BM25) operate on the shared serialized text, while MM-Embed additionally receives the associated figure/table crops when present. For multimodal readers, the corresponding figure/table crops are likewise passed alongside the serialized text when available. This shared serialization is used across chunking methods so that comparisons reflect chunk quality rather than reader-side formatting differences.

E.3. DHP and QA Prompt

We design two instruction-style prompts for our LVLM-based components: (i) a VLM-only Document Hierarchical Parsing (DHP) prompt that predicts a parent for every layout block, and (ii) a RAG-style QA prompt that answers questions from retrieved chunks. Both prompts are intentionally lightweight and model-agnostic so that the same templates can be reused across different LVLM backbones and datasets.

DHP Prompt

You are a document hierarchy parser. Using the provided page images and the full list of layout blocks, predict the parent for *every* block in the document. Your goal is to recover the global dependency tree.

Use only the information given in the input. Do not invent new block ids or additional keys. Avoid self-parenting, cycles, and parents that come *after* the child in reading order (future parents).

`[Input]` A single JSON object with fields `"instruction"`, `"pages"`, and `"blocks"`. The `"pages"` field stores page sizes, and each entry in `"blocks"` contains the block id, page index, label, bounding box, and truncated text.

`[Output]` A JSON array of objects of the form `{"id": "<BLOCK_ID>", "parent": <PARENT_ID or null>}`, one for each input block, in the same order as `"blocks"`. For any block where no valid parent can be assigned under the above constraints, set `"parent"` to `null`. Do not output anything other than this JSON array.

Figure 3. Prompt used for VLM-only Document Hierarchical Parsing (DHP).

LVLM-based DHP. For the LVLM based DHP setting, the LVLM receives page images together with the full list of detected layout blocks and is asked to recover the global dependency tree by predicting a single parent for each block. The prompt encodes page sizes and block attributes (id, page, label, bounding box, truncated text) and constrains the model to output a JSON-only list of (`id`, `parent`) pairs that forms an acyclic tree consistent with reading order. The exact DHP prompt template is shown in Fig. 3.

RAG QA (LVLM read and generate). For downstream QA, the LVLM is given a natural-language question together with a small set of retrieved chunks, each tagged with its section path and page range and serialized under the shared chunk schema described above. The prompt asks the model to answer strictly based on these chunks and to explicitly abstain when the answer is not supported by the context, preventing hallucination and making RAG behavior easier to analyze. The QA prompt template is shown in Fig. 4.

QA Prompt

You are given a question about one or more technical documents. Use only the provided context chunks to answer the question. If the answer cannot be inferred from the context, say "I don't know".

`[Question] <QUESTION_TEXT>`
`[Context chunks]`
`<CHUNK_1>`
`<CHUNK_2>`
`<CHUNK_3>`
`<CHUNK_4>`

Figure 4. Prompt used for LVLM-based RAG QA over retrieved chunks.

F. Extended Quantitative Analyses

F.1. Robustness Across Upstream Modules, Retrieval Backbones, and LVLMs

Table 7 extends the robustness analyses with stronger SharedDet backbones, Table 8 adds retriever swaps, and Table 9 summarizes LVLM substitution results. The overall pattern is consistent: stronger upstream modules improve absolute performance for all structure-aware chunkers, while M3DocDep remains the best-performing method.

F.2. Fairness Control for Structural Metadata

To disentangle the effect of improved chunk boundaries from that of added metadata, we perform a pairwise fairness control in which section-path and page-range fields are removed from both MultiDocFusion and M3DocDep during indexing and reader input. Under this no-metadata control, M3DocDep still retains a 2.3% nDCG advantage over MultiDocFusion. This confirms that the gain is not explained solely by metadata injection: better dependency recovery and boundary formation remain the primary source of improvement.

F.3. Full Ablation on Dependency Recovery

Removing MST-based global decoding or cross-page edges yields the largest degradations, confirming that globally valid tree decoding and long-range document links are the two most important ingredients for stable hierarchy recovery. SoftROI, header-centric parent priors, and candidate pruning provide smaller but still consistent gains.

F.4. Per-corpus and Per-type Breakdowns

The main paper reports macro-averaged DHP, retrieval, and QA metrics across datasets. Here we additionally break down DHP performance by edge type and analyze how different methods behave on structurally difficult subsets.

Method	DP backbone (nDCG)					OCR backbone (nDCG)				
	DETR	DiT	VGT	MinerU2.5	DocLayout	EasyOCR	Tesseract	TrOCR	PaddleOCR	DotsOCR
Structure-based	0.4396	0.4171	0.4269	0.4516	0.4345	0.5194	0.4650	0.2993	0.5248	0.5384
MultiDocFusion	0.5014	0.4976	0.5061	0.5119	0.5047	0.5681	0.5068	0.4097	0.5742	0.5896
M3DocDep	**0.5239**	**0.5127**	**0.5382**	**0.5532**	**0.5325**	**0.5914**	**0.5279**	**0.4235**	**0.5978**	**0.6136**

Table 7. Robustness across stronger SharedDet backbones. Values are macro-averaged nDCG over DUDE, MP-DocVQA, CUAD, and MOAMOB with top-$k \in \{1, 2, 3, 4\}$. Stronger DP/OCR modules raise absolute performance, while M3DocDep remains best in every setting.

Chunking Method	BGE	E5	BM25	MM-Embed	Avg
Length chunking	0.4834	0.4715	0.4764	0.4864	0.4793
Semantic chunking	0.3114	0.3378	0.1825	0.2906	0.2804
LumberChunker	0.4708	0.4319	0.4539	0.4632	0.4549
Perplexity chunking	0.4715	0.4318	0.4495	0.4647	0.4542
Structure-based chunking	0.4679	0.4040	0.4118	0.4591	0.4357
MultiDocFusion	0.5213	0.4884	0.5085	0.5283	0.5116
M3DocDep	**0.5523**	**0.5014**	**0.5321**	**0.5654**	**0.5378**

Table 8. Retrieval-backbone robustness. Values are macro-averaged nDCG over DUDE, MP-DocVQA, CUAD, and MOAMOB with top-$k \in \{1, 2, 3, 4\}$. The benefit of M3DocDep is consistent across sparse, dense, and multimodal retrievers.

M3DocDep LVLM backbone	Qwen2.5-VL	InternVL-3.5	LLaVA-OneVision-1.5
DocHieNet parent F1	76.01	75.71	74.07

Table 9. LVLM swap for multimodal block embeddings under the shared DocHieNet DHP protocol. Performance remains stable across three open LVLM backbones.

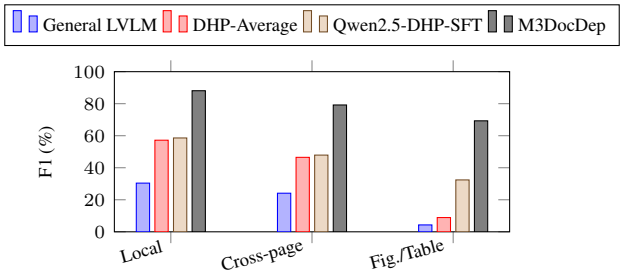


Figure 5. Per-type DHP performance over Local, Cross-page, and Fig./Table edge subsets for General LVLM, DHP-Average, Qwen2.5-DHP-SFT, and M3DocDep, macro-averaged over DocHieNet, HRDH, and HRDS.

Per-type DHP analysis. Figure 5 reports parent-prediction F1(%) on three edge subsets: *Local* (child and parent on the same page), *Cross-page* (child and parent on different pages), and *Fig./Table* (child block type is `figure` or `table`), macro-averaged over DocHieNet, HRDH, and HRDS. Across all methods the Local subset is always easier than Cross-page and Fig./Table, and General LVLMs as well as classical DHP parsers show only moderate accuracy on Local edges while almost completely failing to recover figure/table relations. Qwen2.5-DHP-SFT partially closes this gap, especially for Fig./Table edges, but still lags behind on cross-page structure. In contrast, M3DocDep clearly dominates on all three subsets in the plot: it maintains strong accuracy not only on Local edges but also on cross-page links, and is the only setting that achieves high, stable performance on Fig./Table edges. This suggests that M3DocDep is particularly effective at capturing long-range structure and image/table-centric regions that other approaches largely miss.

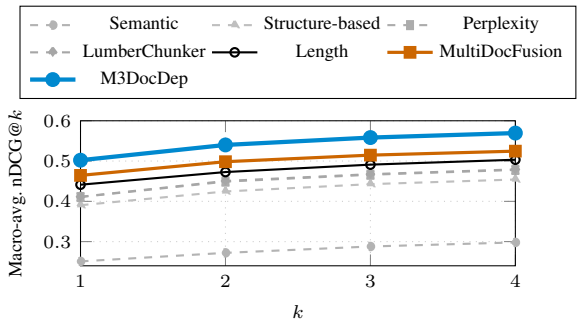


Figure 6. Macro-averaged nDCG@k over DUDE, MP-DocVQA, CUAD, and MOAMOB for $k \in \{1, 2, 3, 4\}$. **M3DocDep** dominates across all k, with the largest margins at small k, and remains consistently ahead as k grows.

F.5. Sensitivity to the Number of Retrieved Chunks

We assess robustness to the retrieval budget by plotting the *macro-averaged* nDCG@k over DUDE, MP-DocVQA, CUAD, and MOAMOB for $k \in \{1, 2, 3, 4\}$ (Figure 6). Across all k, **M3DocDep** consistently yields the best nDCG and preserves a clear margin over every baseline; the advantage is most pronounced under tight budgets (small k) and remains visible as k grows. These results indicate that tree-guided, structure-aware chunking provides high-quality evidence with few retrieved chunks and scales gracefully to larger retrieval budgets.

F.6. Chunk Length and Distribution

By design, our tree-guided chunking keeps chunk lengths within a moderate target range while respecting section boundaries, avoiding both tiny fragments and excessively large chunks. In contrast, purely length-based or seman-

Variant (Ablation)	HRDS		HRDH		DocHieNet	
	F1	STEDS	F1	STEDS	F1	STEDS
Full (SharedDet)	**82.87**	**76.52**	**77.75**	**71.65**	**76.01**	**70.83**
SoftROI $\rightarrow$ *uniform ROI pooling*	81.64 (-1.23)	74.85 (-1.67)	76.81 (-0.94)	70.13 (-1.52)	74.43 (-1.58)	68.92 (-1.91)
MST (global tree) $\rightarrow$ *local argmax*	78.31 (-4.56)	70.59 (-5.93)	71.92 (-5.83)	64.19 (-7.46)	70.82 (-5.19)	64.12 (-6.71)
no header-centric parent prior	81.25 (-1.62)	74.49 (-2.03)	75.26 (-2.49)	68.48 (-3.17)	74.13 (-1.88)	68.37 (-2.46)
candidate top-k pruning: k=8	81.61 (-1.26)	74.79 (-1.73)	76.12 (-1.63)	69.47 (-2.18)	74.55 (-1.46)	68.89 (-1.94)
candidate top-k pruning: k=16	82.68 (-0.19)	76.25 (-0.27)	77.42 (-0.33)	71.24 (-0.41)	75.79 (-0.22)	70.48 (-0.35)
candidate top-k pruning: k=32	82.51 (-0.36)	76.04 (-0.48)	77.28 (-0.47)	71.06 (-0.59)	75.63 (-0.38)	70.21 (-0.62)
disallow cross-page edges	77.53 (-5.34)	69.65 (-6.87)	68.83 (-8.92)	60.39 (-11.26)	68.83 (-7.18)	61.19 (-9.64)

Table 10. Full per-dataset ablation on hierarchy and dependency reconstruction. Each cell is in the form score (Δ), where Δ denotes the change relative to Full (SharedDet).

tic chunkers often produce a mix of very short and very long chunks. This design helps dense retrievers by aligning chunks more closely with underlying semantic units.

G. Qualitative Case Studies and Error Analysis

G.1. Chunking Comparisons

Using representative documents from VQA Datasets, we overlay chunk boundaries from different methods (Length, Semantic, LumberChunker, Perplexity, Structure-based, MultiDocFusion) directly on page images. Color-coding shows that text-based methods often split sections mid-paragraph or separate figures from captions, whereas M3DocDep aligns chunk boundaries with section subtrees and keeps visual content with its description. More details of the chunking method examples are shown in Tab. 12. Figures 2 and 7 make the MST-decoded tree, the induced chunks, and their multimodal bindings concrete; together they visualize the exact tree-to-chunk pathway used in the main method.

Handling of figure/image regions. Figure 7 zooms in on a single figure region and compares how different chunking strategies represent the same content. *Text-based chunking* operates purely on flattened OCR text, so the figure text is mixed with surrounding paragraphs and no explicit notion of a figure region is preserved. *Structure-based chunking* removes this entanglement by isolating the figure and its caption, but still treats them as plain text only. *MultiDocFusion* augments the structure-based chunk with an LLM-predicted `section_path`, yet this path can be misaligned with the true document hierarchy because it is inferred from text alone. In contrast, *M3DocDep* attaches the correct `section_path` from the global dependency tree and keeps the figure region aligned with its textual caption inside the same chunk representation, so figure–caption context is preserved rather than being split across unrelated text. Consequently, M3DocDep is the only compared chunking method that consistently preserves figure/image regions as first-class multimodal units, rather than collapsing them into text-only representations.

G.2. Failure Cases

We qualitatively observe the following typical failure modes:

- **parser misses and block fragmentation:** upstream detectors may split a logical region into several small blocks or merge nearby regions, which destabilizes the candidate-parent set before tree decoding;
- **OCR corruption in degraded scans:** missing or heavily garbled text weakens both the block embedding and the serialized chunk, especially for densely scanned manuals and contracts;
- **ambiguous or implicit headings:** some documents signal section transitions only through typography or whitespace, making header-centric parent selection harder;
- **repeated headers/footers and template artifacts:** boilerplate repeated across pages can attract spurious parents if the visual hierarchy is weak;
- **caption drift and multi-page figures/tables:** long visual regions that span pages or sit far from their captions can still be attached incorrectly when neither tree evidence nor spatial fallback is strong enough.

In such cases the dependency head may attach blocks to suboptimal parents or fail to link the correct captions, leading to imperfect trees and suboptimal chunks. These failures are nevertheless informative: they show that the remaining bottlenecks are concentrated in upstream block quality, OCR corruption, and highly ambiguous layouts rather than in ordinary section-level documents. We consider joint training with layout-normalized variants, stronger weak supervision, and query-aware reranking as promising directions.

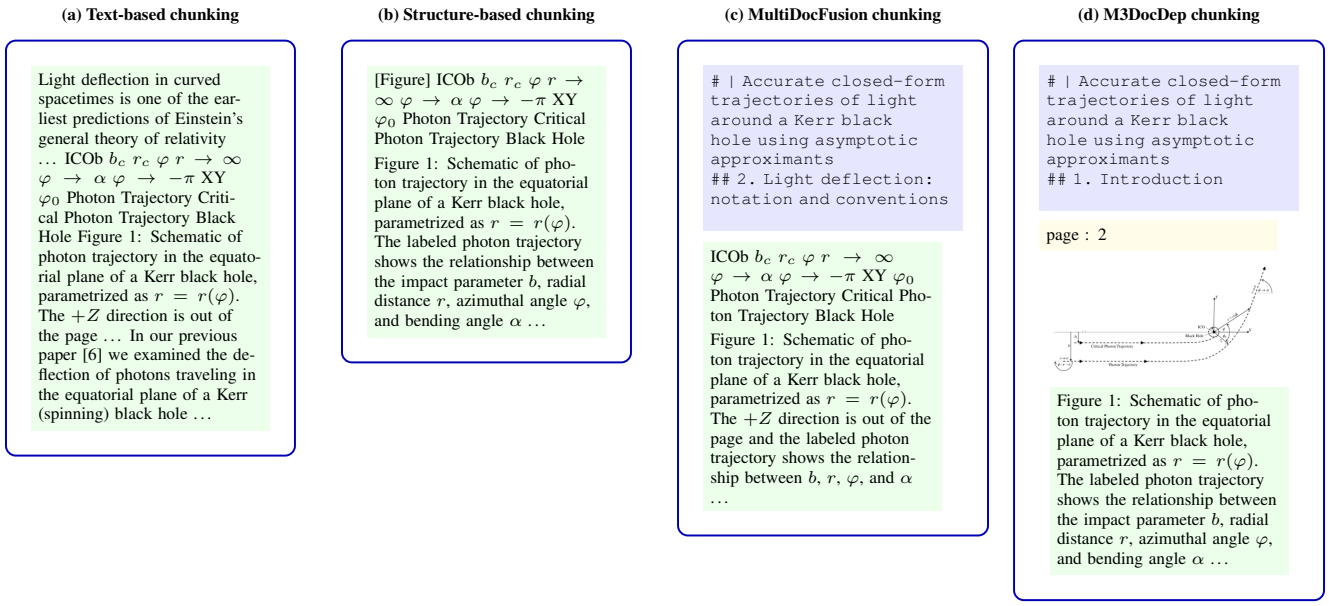


Figure 7. Comparison of four chunking paradigms on the same figure region: text-based chunking mixes surrounding text and OCR'ed figure text; structure-based chunking isolates figure and caption as text only; MultiDocFusion adds a section_path over text; M3DocDep attaches the correct section path, preserves the shared metadata fields when available, and keeps the figure region aligned with its caption in the same chunk representation.

Component	sec/page or memory
SharedDet (DP+OCR)	3.2 sec/page
Core (LVLM + SoftROI + scoring + MST)	0.4 sec/page
Total end-to-end	3.6 sec/page
Core throughput	2.5 pages/s
Peak GPU memory	27 GB
LVLM-only autoregressive hierarchy generation	20 sec/page

Table 11. Runtime summary on a single A100 (80GB). The lightweight M3DocDep core is substantially faster than LVLM-only autoregressive hierarchy generation.

H. Runtime and Scalability Analysis

H.1. Runtime Breakdown

We measure wall-clock runtime on a single A100 (80GB) and separate the end-to-end indexing cost into SharedDet and the M3DocDep core. The dominant cost is upstream DP+OCR (3.2 sec/page), while the core module—LVLM forward, SoftROI pooling, edge scoring, and MST decoding—requires only 0.4 sec/page and reaches 2.5 pages/s at 27 GB peak GPU memory. The runtime breakdown is shown in Table 11.

H.2. Scaling with Document Length

Because DP and OCR operate per page and dependency scoring is restricted to a small parent candidate set ($K \ll N$), the effective complexity of M3DocDep is near-linear in the number of blocks/pages for typical industrial documents. Empirically, the time per document grows roughly linearly with page count, and larger docs can be batched or parallelized. In practice, the core M3DocDep module contributes only a small fraction of the total latency, so larger deployments can parallelize SharedDet aggressively while keeping the dependency and chunking stage lightweight.

I. Limitations and Broader Impact

Limitations. M3DocDep relies on a frozen upstream DP detector and OCR engine; when these components produce fragmented or merged blocks on heavily degraded scans, the downstream dependency head inherits these errors, as discussed in the failure cases (Sec. G). The biaffine scoring head is trained on three DHP corpora that, while diverse, do not cover all industrial document types (e.g., handwritten forms, non-Latin scripts beyond Korean and Chinese). The pipeline currently constructs dependency trees per document; extending it to model inter-document relations (e.g., cross-references between contract annexes) remains future work. Finally, inference speed is dominated by the frozen LVLM forward pass and SharedDet, which may limit deployment on very large corpora without further engineering (e.g., distillation, quantization).

Broader impact. By improving the accuracy of document structure recovery and chunk construction, M3DocDep can help users retrieve more reliable answers

from long, complex documents, which has positive implications for domains such as legal review, technical maintenance, and financial auditing. We do not foresee direct negative societal impacts specific to our method; however, as with any RAG system, downstream answer quality depends on the accuracy of the source documents, and users should verify critical information independently.

Method	Chunk	Example Content
Length chunking	Chunk 1	Accurate closed-form trajectories of light around a Kerr black hole using asymptotic approximants — Ryne J. Beachley1, Morgan Mistysyn2, Joshua A. Faber1,4, Steven J. Weinstein3,4, Nathaniel S. Barlow1,4. 1 School of Mathematical Sciences, Rochester Institute of Technology, Rochester, NY 14623 … Abstract. Highly accurate closed-form expressions that describe the full trajectory of photons propagating in the equatorial plane of a Kerr black hole are obtained using asymptotic approximants …
	Chunk 2	This work extends a prior study of the overall bending angle for photons (Barlow et al. 2017, Class. Quantum Grav., 34, 135017). The expressions obtained provide accurate trajectory predictions for arbitrary spin and impact parameters, and provide significant time advantages compared with numerical evaluation of the elliptic integrals that describe photon trajectories … Keywords: Geodesics, Light deflection, Kerr black holes, Asymptotic approximants. Submitted to: Class. Quantum Grav. …
	Chunk 3	1. Introduction Light deflection in curved spacetimes is one of the earliest predictions of Einstein's general theory of relativity, and one of the best understood aspects of the theory. The null geodesics describing photon trajectories have been investigated for a wide variety of physical configurations in a number of limits … After the initial construction of the Kerr metric describing spinning black holes, many of the early results on null geodesics in these spacetimes were derived by Carter …
Semantic chunking	Chunk 1	Accurate closed-form trajectories of light around a Kerr black hole using asymptotic approximants — Ryne J. Beachley1, Morgan Mistysyn2, Joshua A. Faber1,4, Steven J. Weinstein3,4, Nathaniel S. Barlow1,4. 1 School of Mathematical Sciences, Rochester Institute of Technology, Rochester, NY 14623; 2 Department of Industrial and Systems Engineering; 3 Department of Chemical Engineering; 4 Center for Computational Relativity and Gravitation, Rochester Institute of Technology … E-mail: nsbsma@rit.edu …
	Chunk 2	Abstract. Highly accurate closed-form expressions that describe the full trajectory of photons propagating in the equatorial plane of a Kerr black hole are obtained using asymptotic approximants. This work extends a prior study of the overall bending angle for photons … To construct approximants, asymptotic expansions for photon deflection are required in various limits … new coefficients are reported for the bending angle in the weak-field limit (large impact parameter) …
	Chunk 3	1. Introduction Light deflection in curved spacetimes is one of the earliest predictions of Einstein's general theory of relativity … The limit where photons approach the innermost circular orbit (ICO), referred to as the strong-field limit, has also been explored for decades … We refer readers to Chandrasekhar's work on the subject for a thorough review on geodesics in black hole spacetimes …
LumberChunker	Chunk 1	Accurate closed-form trajectories of light around a Kerr black hole using asymptotic approximants — [title and front matter] Ryne J. Beachley1, Morgan Mistysyn2, Joshua A. Faber1,4, Steven J. Weinstein3,4, Nathaniel S. Barlow1,4 … Abstract. Highly accurate closed-form expressions that describe the full trajectory of photons propagating in the equatorial plane of a Kerr black hole are obtained using asymptotic approximants …
	Chunk 2	Abstract. Highly accurate closed-form expressions that describe the full trajectory of photons … The expressions obtained provide accurate trajectory predictions for arbitrary spin and impact parameters … Keywords: Geodesics, Light deflection, Kerr black holes, Asymptotic approximants. Submitted to: Class. Quantum Grav. 1. Introduction Light deflection in curved spacetimes is one of the earliest predictions of Einstein's general theory of relativity …
	Chunk 3	1. Introduction Light deflection in curved spacetimes is one of the earliest predictions of Einstein's general theory of relativity … The limit where photons approach the innermost circular orbit (ICO) has also been explored for decades … Figure 1: Schematic of photon trajectory in the equatorial plane of a Kerr black hole, parametrized as $r = r(\varphi)$, showing the relationship between impact parameter b, radial distance r, azimuthal angle φ, and bending angle α …
Perplexity chunking	Chunk 1	Accurate closed-form trajectories of light around a Kerr black hole using asymptotic approximants — Ryne J. Beachley1, Morgan Mistysyn2, Joshua A. Faber1,4, Steven J. Weinstein3,4, Nathaniel S. Barlow1,4 … 1 School of Mathematical Sciences, Rochester Institute of Technology, Rochester, NY 14623 … Keywords: Geodesics, Light deflection, Kerr black holes, Asymptotic approximants …
	Chunk 2	Abstract. Highly accurate closed-form expressions that describe the full trajectory of photons propagating in the equatorial plane of a Kerr black hole are obtained using asymptotic approximants. This work extends a prior study of the overall bending angle for photons (Barlow et al. 2017, Class. Quantum Grav., 34, 135017). The expressions obtained provide accurate trajectory predictions for arbitrary spin and impact parameters …
	Chunk 3	Light deflection in curved spacetimes is one of the earliest predictions of Einstein's general theory of relativity, and one of the best understood aspects of the theory … The limit where photons approach the innermost circular orbit (ICO) has also been explored for decades … Figure 1: Schematic of photon trajectory in the equatorial plane of a Kerr black hole, parametrized as $r = r(\varphi)$ …

Table 12. Qualitative comparison of chunking methods applied to the document in Fig. 1 (part 1/2).

Method	Chunk	Example Content
Structure-based	Chunk 1	[Title block] Accurate closed-form trajectories of light around a Kerr black hole using asymptotic approximants — Ryne J. Beachley1, Morgan Mistysyn2, Joshua A. Faber1,4 … Abstract. Highly accurate closed-form expressions that describe the full trajectory of photons … Keywords: Geodesics, Light deflection, Kerr black holes. Submitted to: Class. Quantum Grav.
	Chunk 2	[Section: 1. Introduction] Light deflection in curved spacetimes is one of the earliest predictions of Einstein's general theory of relativity … The limit where photons approach the innermost circular orbit (ICO) has also been explored for decades …
	Chunk 3	[Figure + Caption] Figure 1: Schematic of photon trajectory in the equatorial plane of a Kerr black hole, parametrized as $r = r(\varphi)$. The labeled photon trajectory shows the relationship between the impact parameter b, radial distance r, azimuthal angle φ, and bending angle α …
MultiDocFusion	Chunk 1	`section_path:` # Accurate closed-form trajectories … [Title and front matter] Ryne J. Beachley1 … Abstract. Highly accurate closed-form expressions … Keywords: Geodesics, Light deflection, Kerr black holes …
	Chunk 2	`section_path:` # Accurate closed-form trajectories > ## 1. Introduction Light deflection in curved spacetimes is one of the earliest predictions of Einstein's general theory of relativity … The null geodesics describing photon trajectories have been investigated for a wide variety of physical configurations …
	Chunk 3	`section_path:` # Accurate closed-form trajectories > ## 2. Light deflection Figure 1: Schematic of photon trajectory in the equatorial plane of a Kerr black hole, parametrized as $r = r(\varphi)$ …
M3DocDep	Chunk 1	`section_path:` # Accurate closed-form trajectories … `pages:` 1 [Title and front matter] Ryne J. Beachley1 … Abstract. Highly accurate closed-form expressions … Keywords: Geodesics, Light deflection, Kerr black holes. Submitted to: Class. Quantum Grav.
	Chunk 2	`section_path:` # Accurate closed-form trajectories > ## 1. Introduction `pages:` 1–2 (cross-page) Light deflection in curved spacetimes is one of the earliest predictions of Einstein's general theory of relativity … The null geodesics describing photon trajectories have been investigated for a wide variety of physical configurations in a number of limits …
	Chunk 3	`section_path:` # Accurate closed-form trajectories > ## 1. Introduction `pages:` 2 [Figure region + Caption] Figure 1: Schematic of photon trajectory in the equatorial plane of a Kerr black hole, parametrized as $r = r(\varphi)$. The labeled photon trajectory shows the relationship between the impact parameter b, radial distance r, azimuthal angle φ, and bending angle α …

Table 13. Qualitative comparison of chunking methods applied to the document in Fig. 1 (part 2/2). Structure-based chunking isolates visual regions; MultiDocFusion adds section paths; **M3DocDep** additionally preserves page metadata and keeps figure regions with captions.

References

[1] Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Chunsheng Wu, Huajie Tan, Chunyuan Li, Jing Yang, Jie Yu, Xiyao Wang, Bin Qin, Yumeng Wang, Zizhen Yan, Ziyong Feng, Ziwei Liu, Bo Li, and Jiankang Deng. Llava-onevision-1.5: Fully open framework for democratized multimodal training, 2025. 7

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. 7

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 2

[4] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marcal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4291–4301, 2019. 2

[5] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024. 7

[6] Y. J. Chu and T. H. Liu. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14(10):1396–1400, 1965. 6

[7] André V Duarte, João Marques, Miguel Graça, Miguel Freire, Lei Li, and Arlindo L Oliveira. Lumberchunker: Long-form narrative document segmentation. *arXiv preprint arXiv:2406.17526*, 2024. 3

[8] Jack Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards, Section B*, 71B(4):233–240, 1967. 6

[9] Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. Recurrent chunking mechanisms for long-text machine reading comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6751–6761, 2020. 3

[10] Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. Cuad: An expert-annotated nlp dataset for legal contract review. *NeurIPS*, 2021. 1

[11] Seongtae Hong, Joong Min Shin, Jaehyung Seo, Taemin Lee, Jeongbae Park, Cho Man Young, Byeongho Choi, and Heuiseok Lim. Intelligent predictive maintenance RAG framework for power plants: Enhancing QA with StyleDFS and domain specific instruction tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 805–820, Miami, Florida, US, 2024. Association for Computational Linguistics. 1, 3

[12] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002. 2

[13] Jordy Van Landeghem, Rafał Powalski, Rubèn Tito, Dawid Jurkiewicz, Matthew Blaschko, Łukasz Borchmann, Mickaël Coustaty, Sien Moens, Michał Pietruszka, Bertrand Ackaert, Tomasz Stanisławek, Paweł Józiak, and Ernest Valveny. Document understanding dataset and evaluation (dude). In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19471–19483, 2023. 1

[14] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 2

[15] Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. Mm-embed: Universal multimodal retrieval with multimodal llms, 2025. 7

[16] Jiefeng Ma, Jun Du, Pengfei Hu, Zhenrong Zhang, Jianshu Zhang, Huihui Zhu, and Cong Liu. Hrdoc: dataset and baseline method toward hierarchical reconstruction of document structures. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 2023. 1, 3

[17] Benjamin Paaßen. Revisiting the tree edit distance and its backtracing: A tutorial. *CoRR*, abs/1805.06869, 2018. 2

[18] Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. Doclaynet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 3743–3751, New York, NY, USA, 2022. Association for Computing Machinery. 2

[19] Renyi Qu, Ruixuan Tu, and Forrest Sheng Bao. Is semantic chunking worth the computational cost? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2155–2177, Albu-

querque, New Mexico, 2025. Association for Computational Linguistics. 3

[20] Johannes Rausch, Octavio Martinez, Fabian Bissig, Ce Zhang, and Stefan Feuerriegel. Docparser: Hierarchical document structure parsing from renderings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:4328–4338, 2021. 2

[21] Johannes Rausch, Gentiana Rashiti, Maxim Gusev, Ce Zhang, and Stefan Feuerriegel. Dsg: An end-to-end document structure generator. *arXiv preprint arXiv:2310.09118*, 2023. 2

[22] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009. 7

[23] Joongmin Shin, Chanjun Park, Jeongbae Park, Jaehyung Seo, and Heuiseok Lim. MultiDocFusion : Hierarchical and multimodal chunking pipeline for enhanced RAG on long industrial documents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20996–21015, Suzhou, China, 2025. Association for Computational Linguistics. 3, 4

[24] Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multipage docvqa, 2023. 1

[25] Prashant Verma. S2 chunking: A hybrid framework for document segmentation through integrated spatial and semantic analysis, 2025. 4

[26] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report, 2024. 7

[27] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025. 7

[28] Hangdi Xing, Changxu Cheng, Feiyu Gao, Zirui Shao, Zhi Yu, Jiajun Bu, Qi Zheng, and Cong Yao. Dochienet: A large and diverse dataset for document hierarchy parsing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. 1

[29] Hangdi Xing, Changxu Cheng, et al. Dochienet: A large and diverse dataset for document hierarchy parsing. In *EMNLP*, 2024. 3

[30] Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Renyu Li. Financial report chunking for effective retrieval augmented generation, 2024. 4

[31] Jihao Zhao, Zhiyuan Ji, Pengnian Qi, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu li. Meta-chunking: Learning efficient text segmentation via logical perception, 2024. 4