ShredBench: Evaluating the Semantic Reasoning Capabilities of Multimodal LLMs in Document Reconstruction

Zichun Guo[†,1], **Yuling Shi**[†,2], **Wenhao Zeng**[2], **Chao Hu**[2],
Haotian Lin[2], **Terry Yue Zhuo**[3], **Jiawei Chen**[4], **Xiaodong Gu**[2], **Wenping Ma**[✉,1]
[1]Xidian University [2]Shanghai Jiao Tong University
[3]Alibaba Qwen [4]Old Dominion University
guozichun3@gmail.com, wp_ma@mail.xidian.edu.cn

Abstract

Multimodal Large Language Models (MLLMs) have achieved remarkable performance in Visually Rich Document Understanding (VRDU) tasks, but their capabilities are mainly evaluated on pristine, well-structured document images. We consider content restoration from shredded fragments, a challenging VRDU setting that requires integrating visual pattern recognition with semantic reasoning under significant content discontinuities. To facilitate systematic evaluation of complex VRDU tasks, we introduce SHREDBENCH, a benchmark supported by an automated generation pipeline that renders fragmented documents directly from Markdown. The proposed pipeline ensures evaluation validity by allowing the flexible integration of latest or unseen textual sources to prevent training data contamination. SHREDBENCH assesses four scenarios (English, Chinese, Code, Table) with three fragmentation granularities (8, 12, 16 pieces). Empirical evaluations on state-of-the-art MLLMs reveal a significant performance gap: The method is effective on intact documents; however, once the document is shredded, restoration becomes a significant challenge, with NED dropping sharply as fragmentation increases. Our findings highlight that current MLLMs lack the fine-grained cross-modal reasoning required to bridge visual discontinuities, identifying a critical gap in robust VRDU research[1].

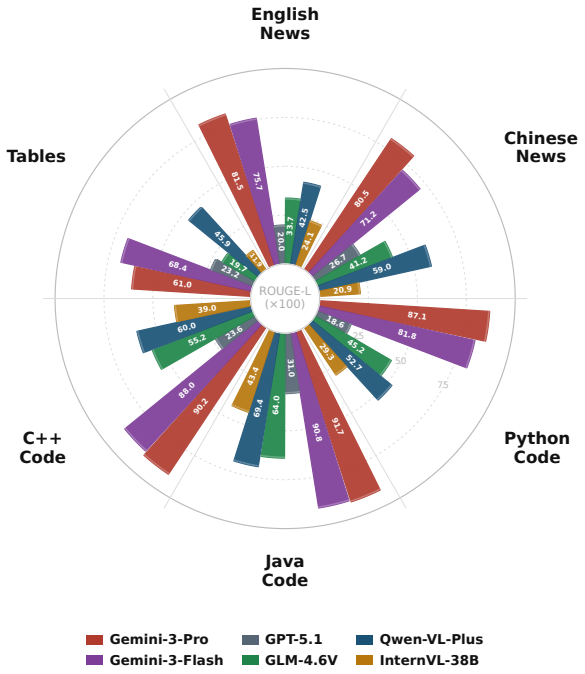

Figure 1: Evaluation results on SHREDBENCH across 6 dimensions (Metric: ROUGE-L). Our proposed benchmark reveals significant gaps in current MLLMs' capabilities on fragmented documents.

1 Introduction

The advance in Multimodal Large Language Models (MLLMs), such as GPT-5 (OpenAI, 2025) and Gemini 3 Pro (Google DeepMind, 2025), has revolutionized the field of Visually Rich Document Understanding (VRDU) (Yin et al., 2024; Wang et al., 2023b, 2025c,b). By projecting visual features into a shared semantic space with textual representations, these models have almost achieved human expert performance on tasks ranging from standard Optical Character Recognition (OCR) (Lee et al., 2023; Lv et al., 2023) to complex information extraction (CIE) from well-formatted documents (Kim et al., 2022; Yu et al., 2023; Tang et al., 2023). However, real-world document processing often encounters inputs that are far from ideal, where documents may be occluded, damaged, or physically torn. Although recent high-resolution MLLMs (Wang et al., 2023a; Li et al., 2024) attempt to mitigate visual noise and enhance fine-grained perception, the specific challenge of reconstructing physically fragmented information remains underexplored. While recent benchmarks have begun to address robustness against image corruptions (Qiu et al., 2025) or super-long con-

[†]Equal contribution. ✉Corresponding author.
[1]Code and dataset are available at https://github.com/ythere-y/ShredBench.

Findings of the Association for Computational Linguistics: ACL 2026, pages 22603–22615

text retrieval (Chia et al., 2024), the challenge of reconstructing physically fragmented information remains underexplored. While humans can rely on strong language priors and world knowledge (Wagemans et al., 2012; Schlichting and Preston, 2015) to mentally piece together fragmented information, the extent to which MLLMs possess this capability remains an open question.

In this paper, we explore *shredded content restoration* at the intersection of vision and NLP. Unlike traditional jigsaw puzzles based on edge matching, this task demands profound semantic reasoning (Zhang, 2024). For instance, connecting *"The algorithm optimiz-"* with *"-es the loss function"* relies less on ambiguous visual cuts than on syntactic expectation. Consequently, this task serves as a rigorous probe for evaluating whether MLLMs can leverage internal language priors to maintain coherence across visual discontinuities.

To systematically evaluate this, we propose SHREDBENCH, a benchmark characterized by three key dimensions: *(1) Multi-Granularity Complexity.* We partition images into 8, 12, and 16 fragments. This hierarchy enables the analysis of how visual entropy correlates with performance degradation. *(2) Diverse Scenarios.* Comprising 756 documents, our dataset spans English and Chinese text, source code (strict syntax), and tables (complex 2D structure). Tables and code are notably difficult, requiring models to restore rigid indentation and alignment—a challenge even for specialized models (Zhang et al., 2024). *(3) Extensive Experiments.* We evaluate state-of-the-art proprietary and open-source MLLMs. Using standard textual metrics, we establish the first quantitative baselines to facilitate future research.

We employ NED, TEDS, BLEU, and ROUGE-L as our primary evaluation metrics and conduct extensive experiments across 14 representative MLLMs, including both leading proprietary and open-source models. The results are sobering: While models exhibit high proficiency on intact documents, their performance collapses under fragmentation. In the hardest setting (16 fragments), the average NED reaches a high of 0.73, even the most advanced models failing to identify correct reading orders or hallucinating non-existent bridging text (Guan et al., 2024; Li et al., 2023b). Our study reveals that current MLLMs struggle to effectively align visual positional embeddings with semantic continuity, often treating fragments as independent entities rather than parts of a cohesive whole.

Our contributions are summarized as follows. First, we introduce SHREDBENCH, the first benchmark specifically designed to stress-test the semantic reasoning capabilities of MLLMs via shredded content restoration. Second, we design an automated pipeline for generating shredded document benchmarks with adjustable granularity. This enables the synthesis of diverse samples covering English and Chinese text, source code, and tables, thereby presenting a comprehensive range of semantic and structural challenges. Third, we conduct a comprehensive evaluation of various MLLMs, revealing significant limitations in their ability to handle visual structural noise and maintain coherence in both textual semantics and 2D spatial layouts.

2 Related Work

2.1 Benchmarking Multimodal Reasoning

Recent MLLM benchmarks have expanded beyond visual perception to evaluate complex reasoning. Representative works include MMBench (Liu et al., 2023b) and SEED-Bench (Li et al., 2023a) for general and generative comprehension, alongside domain-specific benchmarks like MathVista (Lu et al., 2024) and MME-Reasoning (Yuan, 2024) that target mathematical and logical deduction. However, these benchmarks largely focus on coherent and clean inputs, leaving models' ability to reason under structurally disordered or fragmented data underexplored. In contrast, SHREDBENCH is specifically designed to evaluate semantic reconstruction in the presence of structural disruption, providing a rigorous assessment of long-context coherence under disordered inputs.

2.2 Document Parsing and Understanding

The field has evolved from modular OCR to end-to-end MLLMs capable of holistic parsing and understanding. In *document parsing*, models like Nougat (Blecher et al., 2023) reconstruct papers into markup, while TextMonkey (Liu et al., 2024) and Vary (Wei et al., 2023) handle dense text and layout reconstruction. For *document understanding*, proprietary models such as GPT-5 (OpenAI, 2025) and Gemini 3 Pro (Google DeepMind, 2025) show strong zero-shot reasoning, while open-source models like LLaVA (Liu et al., 2023a), Qwen-VL (Bai et al., 2023), and InternVL (Chen et al., 2024) focus on high-resolution processing and reducing hallucinations.

Benchmark	Domain	Modality	Deformation	Reasoning Type	Granularity	Capabilities	
						OCR	Reconst.
Document Parsing Benchmarks							
OmniDocBench (Ouyang et al., 2025)	Document	Text, Table, Formula	/	Structural Parsing	/	✓	✗
WildDoc (Wang et al., 2025a)	Scene Doc	Text, Chart	Shadow, Blur, Warp	Robust Perception	/	✓	✗
DocPTBench (Du et al., 2025)	Photo Doc	Text	Geom. Warp	Parsing & Trans.	/	✓	✗
Visual Jigsaw & Reconstruction Benchmarks							
Jigsaw-Puzzles (Lyu et al., 2025)	Natural Img	Visual Pixels	Grid Crop (2D)	Spatial Arrangement	Grid (2x2 to 5x5)	✗	✓
RePAIR (Tsesmelis et al., 2024)	Artifacts	3D Geometry	Erosion, Fragments	Geometric Matching	/	✗	✓
Proposed Benchmark							
ShredBench (Ours)	Hybrid	Text, Table, Code	3D Shredding	Semantic Bridging	Voronoi (8, 12, 16 pcs)	✓	✓

Table 1: Comparison of ShredBench with representative benchmarks. Domain: target data domain. Modality: input data types. Deformation: visual or physical distortion applied to inputs. Reasoning Type: core cognitive ability evaluated. Granularity: fragment or subunit size/layout. Capabilities: evaluated capabilities, including OCR and implicit reconstruction reasoning.

Comprehensive benchmarks support these tasks: OmniDoc (Ouyang et al., 2025) and HierText (Long et al., 2022) target multi-task reconstruction and dense text perception, DocVQA (Mathew et al., 2021) and ChartQA (Masry et al., 2022) assess information extraction and logical reasoning, and WildDoc (Wang et al., 2025a) evaluates MLLMs on natural scene documents with lighting and physical distortions, revealing robustness limitations.

However, these approaches predominantly assume clear, intact inputs, ignoring scenarios where document structure is physically disrupted. Consequently, the ability of MLLMs to reason over fragmented or shredded documents remains underexplored. SHREDBENCH addresses this gap by evaluating semantic reconstruction under structural disruption, advancing research into physically impaired document understanding.

2.3 Visual Reconstruction

Visual reconstruction has traditionally been framed as the *Jigsaw Puzzle* problem in computer vision. In the image domain, traditional methods use edge detection or Deep Metric Learning (Noroozi and Favaro, 2016; Paixao et al., 2020), and neural approaches like PairingNet (Zhou et al., 2023) leverage graph networks and transformers for improved matching. Benchmarks such as Jigsaw-Puzzles (Lyu et al., 2025) and RePAIR (Tsesmelis et al., 2024) assess spatial reasoning on natural images and fragmented artifacts, but focus primarily on visual or geometric cues.

However, shredded content restoration adds challenges due to sparse text and uniform backgrounds, where visual cues are ambiguous. Semantic reasoning—completing truncated text or formulas—is essential. SHREDBENCH evaluates this capability, testing scenarios beyond the reach of purely visual methods.

3 ShredBench Dataset

In this section, we present the construction process of SHREDBENCH. Our pipeline consists of three stages: content acquisition across multiple domains, physics-based shredding simulation, and the formulation of the reconstruction task.

3.1 Data Collection

To ensure the model's robustness across different semantic contexts and layouts, we constructed a diverse corpus comprising bilingual news, programming code, and scientific tables.

News Articles. We collected high-quality journalism text to represent standard natural language prose. For English content, we scraped articles from *China Daily* via RSS feeds (covering World, Business, and Opinion sections). For Chinese content, we sourced articles from *People.com.cn* (People's Daily Online). To ensure content density, we filtered articles with lengths between 800 and 2,500 characters.

Source Code. To introduce structured syntax and indentation challenges, we utilized the GitHub API to crawl code snippets in three major programming languages: `Python`, `C++`, and `Java`. We specifically targeted files with sizes between 1KB and 4KB and extracted metadata (e.g., commit dates) to enrich the dataset context.

Scientific Tables. To introduce structured data challenges, we sourced tabular samples from the public SWHL table recognition dataset[2]. This

[2] https://huggingface.co/datasets/SWHL/table_rec_test_dataset

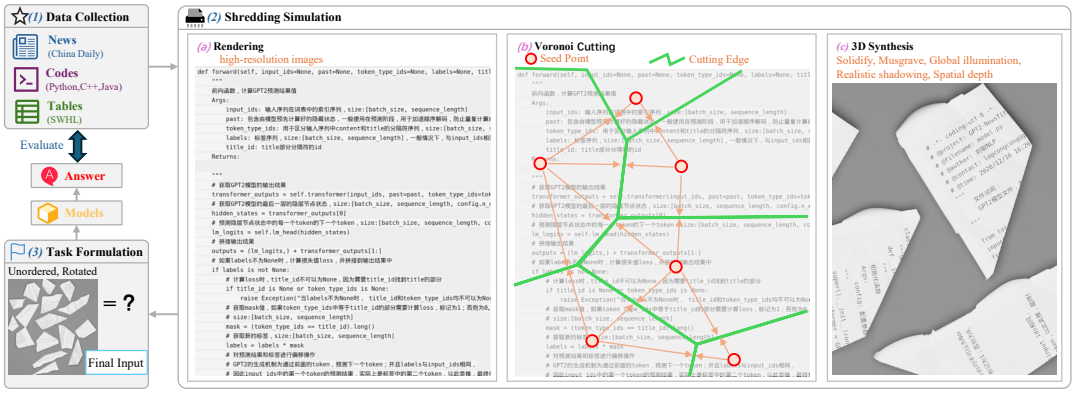

Figure 2: Schematic illustration of the SHREDBENCH data generation pipeline. The process consists of three stages: (1) Data Collection from diverse sources (News, Code, Tables), (2) Shredding Simulation including Voronoi tessellation and physics-based 3D rendering, and (3) Task Formulation where the unordered fragments serve as the final input.

dataset aggregates a diverse range of table layouts, including bordered and borderless styles, complex headers, and spanning cells. Incorporating these samples ensures that SHREDBENCH rigorously evaluates the model's capacity to reconstruct strict spatial dependencies and grid-like structures typical in academic and financial documents.

3.2 Shredding Simulation

Standard 2D cropping preserves pixel-perfect continuity, allowing models to bypass semantic reasoning by exploiting trivial edge matches. To rigorously benchmark document understanding, we developed a physics-based rendering pipeline that simulates real-world artifacts, including crumpling, shadows, and irregular edges. This approach suppresses visual shortcuts, ensuring that successful reconstruction depends on interpreting the semantic context.

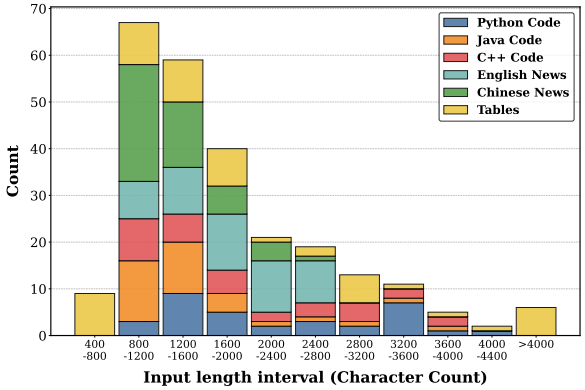

Figure 3: Distribution of dataset input lengths (in characters). The dataset is segmented into intervals of 400 characters, showing the count of files for each category (Code, News, Tables).

Document Rendering. First, raw text data is rendered into high-resolution images (1600px width) using a headless Chrome browser. We apply custom CSS styling (Times New Roman/SimSun fonts, 28px size) and inject random RGB noise to simulate paper texture.

Voronoi Cutting Algorithm. To generate realistic, irregular fragments, we employ a Voronoi tessellation approach. For a given document image, we randomly sample N seed points ($N \in \{8, 12, 16\}$) on the canvas. A k-d tree algorithm assigns each pixel to the nearest seed point, naturally forming jagged, non-rectilinear boundaries that mimic manual shredding.

3D Physical Synthesis. The 2D fragments are then imported into Blender for physical simulation. We apply a *Solidify* modifier (thickness 0.002) and distinct displacement maps: a *Marble* texture for large-scale waves and a *Musgrave* texture for sharp crumples. The fragments are scattered using a pixel-perfect packing algorithm to ensure no overlap. Finally, the scene is rendered using the Cycles engine at 4K resolution (4096×4096) with global illumination, creating realistic shadowing and spatial depth.

3.3 Quality Control

To ensure the rigorousness of SHREDBENCH, we implemented a verification process on a random sample of 50 documents. Two independent human annotators assessed whether the fragments contained sufficient semantic cues for unique reconstruction. The inspection yielded a Cohen's Kappa (κ)(Cohen, 1960) of 0.79, indicating sub-

stantial inter-annotator agreement and confirming the objective nature of the task. Crucially, final adjudication confirmed that 96% of the sampled fragments (48/50) were strictly solvable, while only a marginal fraction (4%) was deemed ambiguous and subsequently removed. Although a minor noise floor exists, it is statistically negligible compared to the drastic performance collapse observed in state-of-the-art MLLMs (avg. NED 0.73), confirming that the reported failure stems from model reasoning limitations rather than data defects.

3.4 Task Formulation

We formulate the shredded content restoration problem as a set-to-sequence task. Formally, let $\mathcal{I} = \{f_1, f_2, \ldots, f_N\}$ be a set of unordered, scattered image fragments derived from a single source document D. The input to the model is the visual set $\mathcal{I}$, where each fragment f_i contains partial visual information, potentially rotated and subjected to lighting distortions.

The objective is to generate a text string $\hat{T}$ that matches the ground-truth text content T of the original document D. Unlike geometric reconstruction tasks that require predicting the spatial coordinates (x, y, θ) of each piece, our task focuses purely on content restoration. The model must implicitly solve the jigsaw puzzle to recover the correct reading order and utilize OCR capabilities to transcribe the text.

4 Experimental Setup

4.1 Models Evaluated

To ensure a comprehensive evaluation across different architectures and capabilities, we selected a diverse set of MLLMs, ranging from proprietary state-of-the-art model APIs to leading open-source model weights.

Proprietary Models: We select GPT-5 Mini and GPT-5.1 (OpenAI, 2025) as representative baselines for efficiency and high-level reasoning, respectively. Similarly, we evaluate Google's Gemini 3 Flash for low-latency tasks and Gemini 3 Pro (Google DeepMind, 2025) for state-of-the-art multimodal logic.

Open-Source Models: InternVL (Chen et al., 2024) and Qwen-VL series (Plus/Flash) (Bai et al., 2023) serve as robust general-purpose baselines with strong visual understanding. For specialized capabilities, we include GLM-4.6v (GLM et al., 2024) for bilingual interactions, and Mistral3-Reasoning (Team, 2025a) for transparent multi-step logic. In the domain of document parsing, we evaluate DeepSeek-OCR (Wu et al., 2024), which utilizes an MoE visual encoder for high-resolution processing, and Hunyuan-OCR (Team, 2025b), optimized for end-to-end text spotting.

4.2 Evaluation Metrics

We employ three standard metrics to quantitatively evaluate the similarity between the generated text and the ground truth. Let Y denote the ground truth (reference) text and $\hat{Y}$ denote the generated text (hypothesis).

NED and TEDS: We employ Normalized Edit Distance (NED) (Levenshtein, 1965) for general text similarity. It normalizes the Levenshtein distance (Lev) between prediction $\hat{Y}$ and ground truth Y:

$$NED(Y, \hat{Y}) = \frac{Lev(Y, \hat{Y})}{\max(|Y|, |\hat{Y}|)} \tag{1}$$

A lower NED implies higher similarity. For tables, we use Tree-Edit-Distance-based Similarity (TEDS) (Zhong et al., 2020), which models content as trees (e.g., HTML DOM) to assess both structure and accuracy:

$$TEDS(T, \hat{T}) = 1 - \frac{TED(T, \hat{T})}{\max(|T|, |\hat{T}|)} \tag{2}$$

where $TED(\cdot)$ is the tree edit distance; higher scores indicate better reconstruction.

BLEU (Bilingual Evaluation Understudy): Proposed by Papineni et al. (2002), BLEU calculates the geometric mean of n-gram precision, penalized for brevity:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{3}$$

where p_n is n-gram precision and w_n are weights. The Brevity Penalty (BP) accounts for generation length bias:

$$BP = \begin{cases} 1 & \text{if } c > r, \\ e^{(1-r/c)} & \text{if } c \leq r, \end{cases} \tag{4}$$

with c and r denoting generated and reference lengths, respectively.

Model	8 Fragments			12 Fragments			16 Fragments		
	NED↓	BLEU↑	ROUGE↑	NED↓	BLEU↑	ROUGE↑	NED↓	BLEU↑	ROUGE↑
Open-source Models									
InternVL3.5-8B	0.78	0.07	0.24	0.79	0.05	0.21	0.78	0.05	0.21
InternVL3.5-14B	0.76	0.08	0.26	0.77	0.07	0.24	0.78	0.07	0.23
InternVL3.5-38B	0.74	0.10	0.28	0.75	0.08	0.26	0.76	0.08	0.24
Mistral3-Reas-8B	0.77	0.09	0.28	0.79	0.06	0.23	0.79	0.06	0.24
Mistral3-Reas-14B	0.76	0.10	0.30	0.77	0.09	0.28	0.77	0.08	0.27
DeepSeek-OCR	0.86	0.02	0.12	0.87	0.01	0.09	0.87	0.01	0.10
Hunyuan-OCR	0.88	0.01	0.15	0.88	0.01	0.14	0.89	0.00	0.12
GLM-4.6v	0.67	0.20	0.45	0.70	0.17	0.40	0.71	0.15	0.37
Qwen-VL-Flash	0.59	0.26	0.58	0.63	0.22	0.54	0.65	0.19	0.50
Qwen-VL-Plus	0.59	0.26	0.58	0.63	0.22	0.53	0.65	0.20	0.50
Proprietary Models									
GPT-5 Mini	0.86	0.06	0.27	0.84	0.04	0.26	0.84	0.05	0.25
GPT-5.1	0.77	0.07	0.28	0.81	0.05	0.22	0.82	0.04	0.21
Gemini 3 Flash	0.34	0.47	0.82	0.40	0.44	0.77	0.44	0.41	0.74
Gemini 3 Pro	**0.33**	**0.51**	**0.83**	**0.37**	**0.48**	**0.81**	**0.41**	**0.44**	**0.76**

Table 2: Overall Performance Summary. Aggregated results across all categories. The metrics are split into separate columns for clarity: NED (↓), BLEU (↑), and ROUGE (↑). Gemini 3 Pro shows consistent superiority across all settings.

ROUGE-L: We use ROUGE-L (Lin, 2004) to capture sentence-level structure via the Longest Common Subsequence (LCS). Precision (P_{lcs}) and recall (R_{lcs}) are defined as:

$$R_{lcs} = \frac{LCS(Y, \hat{Y})}{|Y|}, \quad P_{lcs} = \frac{LCS(Y, \hat{Y})}{|\hat{Y}|} \quad (5)$$

The final score is the weighted F-measure of these components:

$$ROUGE - L = \frac{(1 + \beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \quad (6)$$

where β controls the relative importance of precision versus recall.

4.3 Performance Analysis

In this section, we conduct a multi-dimensional analysis of reconstruction performance. Our evaluation is structured into four key aspects: (1) Natural Language, covering general prose; (2) Source Code, focusing on syntactic logic; (3) Structured Data, assessing tabular processing; and (4) Granularity Impact, analyzing performance degradation as fragment counts increase. Table 2 summarizes the overall performance across all categories. Gemini 3 Pro demonstrates the strongest resilience, achieving the lowest NED (0.33) and highest ROUGE (0.83) scores at the 8-fragment level, consistently outperforming other proprietary and open-source models.

Natural Language (Table 3). We observe a marked performance disparity between languages, with models consistently scoring lower on Chinese News compared to English. This divergence stems partially from the high information density of Chinese logograms: unlike Latin scripts where redundancy is distributed across multi-letter words, a physical tear through a single Chinese character often obliterates its semantic identity, creating a harder reconstruction task. Furthermore, this numerical gap is amplified by metric sensitivity. Since metrics like BLEU and ROUGE rely on exact n-gram matching, the lack of explicit delimiters in Chinese means that even minor reconstruction errors can disrupt word segmentation boundaries, disproportionately penalizing the scores compared to English.

Source Code (Table 4). Results reveal a performance hierarchy driven by syntax. Averaged across all models and fragment settings ($N \in \{8, 12, 16\}$), explicitly structured languages like Java (Avg. NED 0.59) and C++ (0.62) outperform Python (0.68). We attribute this to syntactic redundancy: explicit delimiters (curly braces '{ }', sem icolons) act as visual anchors for alignment. Conversely, Python's whitespace dependence proves challenging as shredding disrupts spatial layout. Lacking explicit closures, models struggle to infer indentation and maintain logical scope, resulting in higher structural error rates.

Structured Data (Table 5). Table reconstruction presents a unique anomaly. While Gemini 3 Pro leads in text and code, Gemini 3 Flash significantly outperforms it on tabular data (NED 0.49 vs. 0.59). We suspect Flash's architecture might be more optimized for preserving rigid 2D spatial structures,

Model	English News			Chinese News		
	$N = 8$	$N = 12$	$N = 16$	$N = 8$	$N = 12$	$N = 16$
Open-source Models						
InternVL3.5-8B	0.75 / 0.07 / 0.18	0.77 / 0.06 / 0.18	0.77 / 0.04 / 0.15	0.91 / 0.01 / 0.30	0.92 / 0.02 / 0.22	0.91 / 0.02 / 0.21
InternVL3.5-14B	0.73 / 0.11 / 0.22	0.73 / 0.11 / 0.25	0.75 / 0.06 / 0.19	0.92 / 0.01 / 0.26	0.92 / 0.01 / 0.20	0.93 / 0.01 / 0.22
InternVL3.5-38B	0.70 / 0.14 / 0.27	0.71 / 0.13 / 0.25	0.74 / 0.07 / 0.20	0.92 / 0.01 / 0.24	0.92 / 0.01 / 0.20	0.92 / 0.01 / 0.19
Mistral3-Reas-8B	0.70 / 0.14 / 0.30	0.71 / 0.11 / 0.25	0.72 / 0.08 / 0.22	0.94 / 0.04 / 0.23	0.95 / 0.02 / 0.16	0.96 / 0.02 / 0.17
Mistral3-Reas-14B	0.69 / 0.17 / 0.29	0.71 / 0.16 / 0.28	0.71 / 0.13 / 0.25	0.93 / 0.05 / 0.26	0.94 / 0.03 / 0.24	0.94 / 0.03 / 0.25
DeepSeek-OCR	0.80 / 0.03 / 0.13	0.82 / 0.02 / 0.12	0.83 / 0.01 / 0.10	0.95 / 0.00 / 0.13	0.95 / 0.01 / 0.10	0.95 / 0.01 / 0.13
Hunyuan-OCR	0.88 / 0.02 / 0.08	0.85 / 0.02 / 0.08	0.88 / 0.01 / 0.07	0.92 / 0.01 / 0.27	0.94 / 0.01 / 0.23	0.94 / 0.00 / 0.24
GLM-4.6v	0.66 / 0.31 / 0.38	0.70 / 0.27 / 0.34	0.70 / 0.21 / 0.30	0.86 / 0.03 / 0.47	0.86 / 0.03 / 0.41	0.88 / 0.03 / 0.35
Proprietary Models						
Qwen-VL-Flash	0.58 / 0.40 / 0.49	0.63 / 0.35 / 0.43	0.65 / 0.27 / 0.37	0.76 / 0.09 / 0.63	0.82 / 0.07 / 0.57	0.83 / 0.06 / 0.54
Qwen-VL-Plus	0.59 / 0.41 / 0.47	0.63 / 0.35 / 0.43	0.65 / 0.28 / 0.38	0.77 / 0.08 / 0.63	0.79 / 0.08 / 0.58	0.84 / 0.06 / 0.56
GPT-5 Mini	0.82 / 0.04 / 0.23	0.82 / 0.04 / 0.24	0.86 / 0.01 / 0.16	0.97 / 0.04 / 0.30	0.97 / 0.03 / 0.29	0.98 / 0.03 / 0.27
GPT-5.1	0.74 / 0.09 / 0.22	0.73 / 0.08 / 0.23	0.80 / 0.03 / 0.15	0.94 / 0.06 / 0.32	0.96 / 0.03 / 0.24	0.96 / 0.03 / 0.25
Gemini 3 Flash	0.20 / 0.81 / 0.85	0.31 / 0.75 / 0.76	0.41 / 0.67 / 0.67	0.59 / 0.11 / 0.75	0.68 / 0.10 / 0.70	0.74 / 0.09 / 0.68
Gemini 3 Pro	**0.16 / 0.87 / 0.90**	**0.25 / 0.79 / 0.82**	**0.35 / 0.70 / 0.73**	**0.47 / 0.14 / 0.84**	**0.57 / 0.12 / 0.81**	**0.60 / 0.10 / 0.76**

Table 3: Natural Language Reconstruction. Comparison on English and Chinese News. Format: NED (↓) / BLEU (↑) / ROUGE (↑). Models are grouped by availability (Open-source vs. Proprietary).

Model	C++			Java			Python		
	$N = 8$	$N = 12$	$N = 16$	$N = 8$	$N = 12$	$N = 16$	$N = 8$	$N = 12$	$N = 16$
Open-source Models									
InternVL3.5-8B	.67 / .15 / .34	.74 / .10 / .29	.70 / .11 / .27	.67 / .17 / .37	.67 / .13 / .35	.66 / .14 / .36	.76 / .07 / .24	.76 / .05 / .22	.73 / .07 / .26
InternVL3.5-14B	.63 / .18 / .40	.69 / .11 / .32	.68 / .13 / .32	.63 / .17 / .41	.65 / .15 / .39	.65 / .16 / .38	.73 / .07 / .26	.76 / .05 / .24	.73 / .09 / .30
InternVL3.5-38B	.61 / .20 / .43	.65 / .15 / .38	.65 / .18 / .36	.60 / .21 / .46	.61 / .18 / .41	.62 / .20 / .43	.72 / .11 / .29	.73 / .07 / .29	.73 / .08 / .30
Mistral3-Reas-8B	.71 / .12 / .31	.75 / .06 / .25	.75 / .07 / .24	.64 / .18 / .42	.66 / .14 / .36	.66 / .15 / .39	.75 / .08 / .31	.76 / .05 / .27	.74 / .07 / .28
Mistral3-Reas-14B	.69 / .14 / .35	.69 / .12 / .33	.71 / .10 / .28	.60 / .21 / .47	.64 / .17 / .41	.62 / .17 / .44	.71 / .09 / .32	.73 / .08 / .30	.72 / .09 / .33
DeepSeek-OCR	.82 / .03 / .14	.83 / .02 / .11	.84 / .02 / .10	.81 / .05 / .17	.85 / .02 / .10	.83 / .03 / .11	.86 / .01 / .09	.87 / .01 / .06	.86 / .01 / .09
Hunyuan-OCR	.87 / .03 / .06	.91 / .00 / .02	.91 / .00 / .02	.91 / .00 / .03	.90 / .00 / .02	.91 / .00 / .03	.89 / .01 / .07	.91 / .01 / .04	.91 / .00 / .04
GLM-4.6v	.51 / .37 / .61	.56 / .31 / .54	.58 / .26 / .50	.45 / .44 / .67	.51 / .37 / .64	.51 / .36 / .61	.63 / .20 / .48	.65 / .14 / .44	.65 / .16 / .43
Proprietary Models									
Qwen-VL-Flash	.47 / .43 / .66	.48 / .37 / .62	.54 / .31 / .54	.42 / .48 / .74	.48 / .45 / .68	.55 / .41 / .63	.57 / .32 / .55	.56 / .25 / .55	.60 / .22 / .51
Qwen-VL-Plus	.47 / .43 / .66	.53 / .35 / .59	.55 / .33 / .55	.42 / .49 / .73	.47 / .45 / .70	.53 / .44 / .65	.59 / .31 / .56	.58 / .25 / .54	.58 / .20 / .48
GPT-5 Mini	.74 / .13 / .30	.74 / .09 / .25	.78 / .08 / .21	.75 / .15 / .38	.79 / .08 / .29	.74 / .13 / .36	.98 / .04 / .19	.80 / .04 / .23	.75 / .08 / .28
GPT-5.1	.69 / .14 / .29	.76 / .05 / .21	.76 / .06 / .21	.63 / .14 / .37	.71 / .07 / .26	.70 / .10 / .30	.76 / .05 / .16	.77 / .05 / .18	.78 / .05 / .22
Gemini 3 Flash	.21 / .73 / .91	.23 / .68 / .89	.29 / .66 / .85	.20 / .78 / .92	.21 / .78 / .91	.23 / .71 / .89	.23 / .68 / .86	.32 / .61 / .79	.30 / .58 / .80
Gemini 3 Pro	**.20 / .78 / .92**	**.21 / .76 / .90**	**.25 / .74 / .88**	**.18 / .84 / .93**	**.19 / .81 / .92**	**.22 / .77 / .90**	**.20 / .72 / .88**	**.19 / .70 / .88**	**.25 / .64 / .85**

**Leading zeros (e.g., 0.74) are omitted in this table for space efficiency.*

Table 4: Source Code Reconstruction Breakdown. Detailed metrics for C++, Java, and Python. Format: NED (↓), BLEU (↑), and ROUGE (↑). Open-source and Proprietary models are separated.

Category: Structured Table Data (Not Text/Code)			
Model	$N = 8$	$N = 12$	$N = 16$
Open-source Models			
InternVL3.5-8B	0.85 / 0.06 / 0.10	0.83 / 0.07 / 0.09	0.84 / 0.03 / 0.09
InternVL3.5-14B	0.82 / 0.03 / 0.12	0.82 / 0.04 / 0.11	0.84 / 0.02 / 0.09
InternVL3.5-38B	0.80 / 0.06 / 0.12	0.79 / 0.05 / 0.12	0.79 / 0.05 / 0.11
Mistral3-Reas-8B	0.82 / 0.05 / 0.20	0.84 / 0.03 / 0.16	0.83 / 0.05 / 0.19
Mistral3-Reas-14B	0.83 / 0.03 / 0.19	0.84 / 0.05 / 0.18	0.84 / 0.03 / 0.16
DeepSeek-OCR	0.87 / 0.01 / 0.07	0.85 / 0.00 / 0.06	0.86 / 0.01 / 0.06
Hunyuan-OCR	0.80 / 0.09 / 0.30	0.81 / 0.04 / 0.30	0.83 / 0.03 / 0.22
GLM-4.6v	0.75 / 0.04 / 0.23	0.79 / 0.04 / 0.19	0.82 / 0.05 / 0.16
Proprietary Models			
Qwen-VL-Flash	0.62 / 0.15 / 0.49	0.66 / 0.12 / 0.44	0.63 / 0.12 / 0.48
Qwen-VL-Plus	0.61 / 0.17 / 0.51	0.68 / 0.12 / 0.44	0.67 / 0.10 / 0.43
GPT-5 Mini	0.84 / 0.06 / 0.25	0.85 / 0.06 / 0.24	0.85 / 0.05 / 0.24
GPT-5.1	0.80 / 0.10 / 0.30	0.84 / 0.06 / 0.22	0.87 / 0.04 / 0.18
Gemini 3 Flash	**0.49 / 0.23 / 0.69**	**0.49 / 0.22 / 0.68**	**0.49 / 0.22 / 0.68**
Gemini 3 Pro	0.59 / 0.20 / 0.63	0.58 / 0.22 / 0.63	0.61 / 0.19 / 0.57

Table 5: Structured Data Reconstruction. Evaluation on tabular data. Format: NED (↓) / TEDS (↑) / ROUGE (↑).

whereas Pro prioritizes semantic flow, which can sometimes be detrimental when "reading" a non-linear table.

4.4 Impact of Granularity

We analyze the rate of performance decay as fragmentation increases ($N = 8 \rightarrow 16$). As shown in Table 2, performance degrades linearly for most models. However, stronger models exhibit a "flatter" decay curve. For instance, while Qwen-VL-Plus sees a significant NED increase (+0.14) when moving from 8 to 16 fragments, Gemini 3 Pro is remarkably stable, with NED increasing by only 0.08. This suggests that advanced reasoning models can maintain global coherence even when the local visual context is severely partitioned.

5 Qualitative Analysis

To understand the cognitive processes underlying reconstruction, we examine specific success and failure modes visualized in our case studies.

5.1 Success Cases: Visual Semantic Bridging

Figure 4 illustrates a successful reconstruction of a news article by Gemini 3 Pro. The model demonstrates two key capabilities. First, regarding visual closure (green highlights), the model successfully recovers words that are physically bisected by cuts. For example, the word "school" was split across two separate shards. The model did not merely OCR the fragments as "sch" and "ool"; instead, it synthesized the disjointed visual cues to recover the complete token "school". This indicates the model

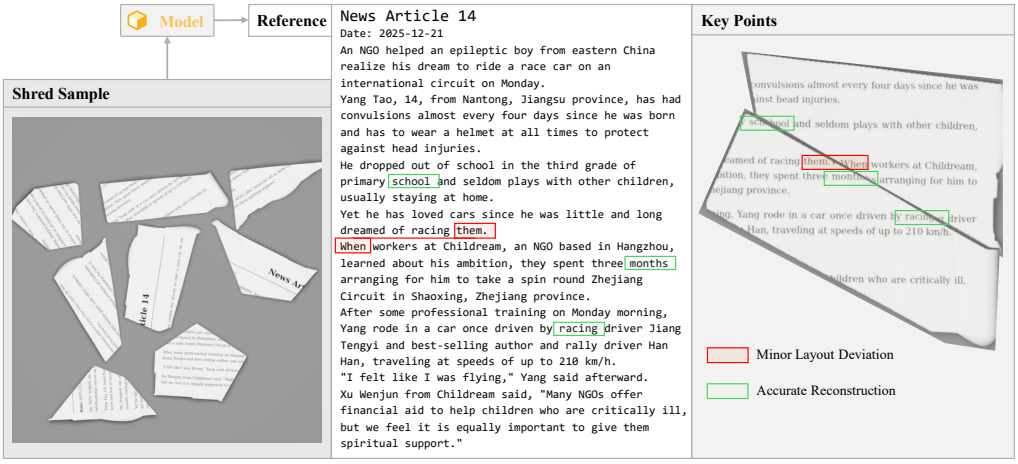

Figure 4: Good Case Study. The red rectangle highlights a minor layout inconsistency where the model interpreted a horizontal gap between fragments as a paragraph boundary (over-segmentation), despite the semantic continuity. The green rectangle demonstrates the model's robustness to physical fragmentation. Even though the characters are physically bisected, the model accurately synthesizes the disjointed visual cues to recover the complete word.

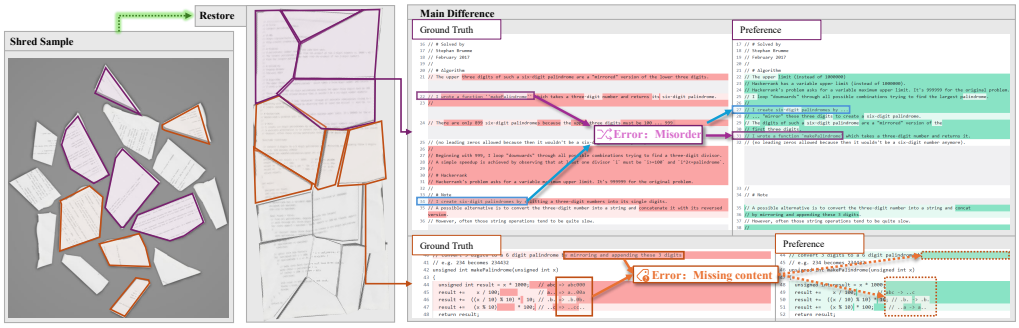

Figure 5: Bad Case Study. An example of code reconstruction failure. The pink arrow indicates an ordering error, where lines of code were structurally recognized but placed in the wrong logical sequence due to ambiguous visual cues. The orange box highlights content loss, where a narrow strip containing code (e.g., `unsigned int`) was completely omitted, likely treated as visual noise.

is performing *multimodal bridging*—using visual edge continuity to inform semantic prediction. Second, regarding layout sensitivity (red highlight), the model is highly sensitive to physical gaps. In one instance, a horizontal gap between fragments was misinterpreted as a paragraph break ("When workers..."), leading to a minor layout deviation (over-segmentation) despite the text being semantically continuous.

5.2 Failure Analysis: Where do MLLMs fail?

Despite high aggregate scores, models struggle with global logic in complex documents, as seen in the code reconstruction example in Figure 5.

Regarding ordering error (pink highlight), the most common error in code is *logical misalignment*. The model correctly identified the text of lines 22 and 34 but swapped their order. Unlike prose, where semantic flow dictates order, code often consists of independent statements whose order is determined solely by algorithm logic, which is harder for the model to infer from visual shards alone. As for content loss (orange highlight), we observe instances of "Hallucinated Deletion," where the model omits an entire line of code (e.g., line 43 'mirroring and appending this three digits'). This tends to happen with small, narrow strips of paper that contain only one line of text; the model may treat these isolated shards as visual noise or fail to integrate them into the larger context.

6 Conclusion

In this work, we introduced SHREDBENCH, a novel benchmark for evaluating the shredded content restoration capabilities of Multimodal LLMs. Our experiments across 756 documents and various modalities reveal that reconstruction is not merely a visual matching task but a complex reasoning challenge requiring the integration of visual cues (edge continuity) and semantic priors (language

modeling).

We find that Gemini 3 Pro establishes a new state-of-the-art, demonstrating superior resilience to fragmentation. However, significant challenges remain, particularly in strictly structured data (Tables), where even top models struggle to align disjointed cells.

Limitations

Our study operates under specific controlled constraints. First, regarding regular cuts, we employ rectilinear grid cuts in our dataset, whereas real-world document destruction often involves irregular tearing or cross-cut shredding mechanics. Second, regarding our 2D assumption, we assume all fragments are flat and fully visible, currently abstracting away 3D physical complexities such as crumpling, folding, or occlusion between overlapping pieces. Third, regarding digital synthesis, while our "ShredBench" pipeline mimics physical fragmentation, domain shifts introduced by real-world environmental factors—such as variable lighting conditions and paper textures—remain an area for future exploration.

Acknowledgements

We thank Haoran Gu for the helpful discussions. This paper is supported by the National Key Research and Development Program of China (Grant No. 2023YFB4503802) and the Natural Science Foundation of Shanghai (Grant No. 25ZR1401175).

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier of large multimodal models. *arXiv preprint arXiv:2308.12966*.

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie He, Tong Xu, and 1 others. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yew Ken Chia, Liying Cheng, Hou Pong Chan, Chaoqun Liu, Maojia Song, Sharifah Mahani Aljunied, Soujanya Poria, and Lidong Bing. 2024. M-longdoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework. *arXiv preprint arXiv:2411.06176*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Yongkun Du, Pinxuan Chen, Xuye Ying, and Zhineng Chen. 2025. Docptbench: Benchmarking end-to-end photographed document parsing and translation. *arXiv preprint arXiv:2511.18434*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, and 1 others. 2024. Glm-4: Towards intelligent chat agents. *arXiv preprint arXiv:2406.12793*.

Google DeepMind. 2025. Gemini: Most capable AI models. https://deepmind.google/models/gemini/pro/. Accessed: 2026-01-04.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lijie Chen, Furong Furrer, Yabo Dou, and 1 others. 2024. Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v(ision), llava-1.5, and gemini. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*, pages 498–517. Springer.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Ming-Wei Shaw, Peter andchang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning (ICML)*, pages 18888–18912.

Vladimir I Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. 2023a. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 292–305.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2024. Monkey: Image resolution and text label are important things for large multi-modal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. In *NeurIPS*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Chi Zhang, Wattanit Zhao, and 1 others. 2023b. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*.

Yuliang Liu and 1 others. 2024. Textmonkey: An ocr-free large multimodal model for understanding document. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. 2022. Towards end-to-end unified scene text detection and layout analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1049–1059.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Rachel Hannan, Gabriel Cheng, and Kai-Wei ands others Chang. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*.

Tengchao Lv, Yupan Huang, and Furu Wei. 2023. Kosmos-2.5: A multimodal literate model. *arXiv preprint arXiv:2309.11419*.

Zesen Lyu, Dandan Zhang, Wei Ye, Fangdi Li, Zhihang Jiang, and Yao Yang. 2025. Jigsaw-puzzles: From seeing to understanding to reasoning in vision-language models. *arXiv preprint arXiv:2505.20728*.

Ahmed Masry, Xuan Do, Joty Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 69–84. Springer.

OpenAI. 2025. Introducing GPT-5. https://openai.com/zh-Hans-CN/index/introducing-gpt-5/. Accessed: 2026-01-04.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. 2025. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thiago M Paixao, Rodrigo F Berriel, Maria C Boeres, Alessandro L Oliveira, Claudine Badue, and Alberto F De Souza. 2020. Fast(er) reconstruction of shredded text documents via self-supervised deep asymmetric metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Xinkuan Qiu, Meina Kan, Yongbin Zhou, and Shiguang Shan. 2025. Benchmarking multimodal large language models against image corruptions. *IEEE/CVF International Conference on Computer Vision (ICCV)*. Open Access.

Margaret L Schlichting and Alison R Preston. 2015. Memory integration: neural mechanisms and implications for behavior. *Current Opinion in Behavioral Sciences*, 1:1–8.

Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chengnian Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19254–19264.

Mistral AI Team. 2025a. Magistral: A multimodal reasoning framework for transparent logic. *arXiv preprint arXiv:2506.10910*.

Tencent Hunyuan Vision Team. 2025b. Hunyuanocr technical report. *arXiv preprint arXiv:2511.19575*.

Theodore Tsesmelis, Luca Palmieri, Marina Khoroshiltseva, Adeela Islam, Gur Elkin, Ofir Itzhak Shahar, Gianluca Scarpellini, and 1 others. 2024. Reassembling the past: The repair dataset and benchmark for real world 2d and 3d puzzle solving. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

Johan Wagemans, Jacob Feldman, Sergei Gepshtein, Ruth Kimchi, James R Pomerantz, Peter A van der Helm, and Cees van Leeuwen. 2012. A Century

of Gestalt Psychology in Visual Perception II. Conceptual and Theoretical Foundations. *Psychological Bulletin*, 138(6):1218–1252.

An-Lan Wang, Jingqun Tang, Liao Lei, Hao Feng, Qi Liu, Xiang Fei, Jinghui Lu, Han Wang, Weiwei Liu, Hao Liu, Yuliang Liu, Xiang Bai, and Can Huang. 2025a. Wilddoc: How far are we from achieving comprehensive and robust document understanding in the wild? *arXiv preprint arXiv:2505.11015*.

Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao. 2025b. Vidorag: Visual document retrieval-augmented generation via dynamic iterative reasoning agents. *arXiv preprint arXiv:2502.18017*.

Qiuchen Wang, Ruixue Ding, Yu Zeng, Zehui Chen, Lin Chen, Shihang Wang, Pengjun Xie, Fei Huang, and Feng Zhao. 2025c. Vrag-rl: Empower vision-perception-based rag for visually rich information understanding via iterative reasoning with reinforcement learning. *arXiv preprint arXiv:2505.22019*.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, and 1 others. 2023a. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*.

Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. 2023b. Vrdu: A benchmark for visually-rich document understanding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5184–5193.

Haoran Wei and 1 others. 2023. Vary: Scaling up the vision vocabulary for large vision-language models. *arXiv preprint arXiv:2312.06109*.

Zhiyu Wu and 1 others. 2024. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*. Updated version in 2024.

Yuechen Yu, Yulin Li, Chengquan Zhang, Xiaoqiang Zhang, Zengyuan Guo, Xiameng Qin, Kun Yao, Junyu Han, Errui Ding, and Jingdong Wang. 2023. Structextv2: Masked visual-textual prediction for document image pre-training. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Yilei Jiang Yiting Lu Renrui Zhang Kaituo Feng Chaoyou Fu Tao Chen Lei Bai Bo Zhang Xiangyu Yue Yuan, Tianshuo Peng. 2024. Mme-reasoning: A comprehensive benchmark for logical reasoning in mllms. *arXiv preprint arXiv:2505.21327*.

Jinxu Zhang. 2024. Read and Think: An Efficient Step-wise Multimodal Language Model for Document Understanding and Reasoning. *arXiv preprint arXiv:2403.00816*.

Tianshu Zhang, Xiang Yue, Yifei Li, Hunar Batra, Shangmin Guo, Shiyu Chen, Linbin Wang, Semih Yavuz, Richard Yan, Xinyu Zhang, and Tao Yu. 2024. Tablellama: Towards open large generalist models for tables. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Xu Zhong, Elaheh Shafieibavani, and Antonio Jimeno Yepes. 2020. Image-based table recognition: data, model, and evaluation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 564–580. Springer.

Rixin Zhou, Ding Xia, Yi Zhang, Honglin Pang, Xi Yang, and Chuntao Li. 2023. Pairingnet: A learning-based pair-searching and -matching network for image fragments. *arXiv preprint arXiv:2312.08704*.

A Additional Evaluation: Metric Suitability for Code Restoration

While standard string-matching metrics (such as NED, BLEU, and ROUGE) offer a robust general measure of text similarity, they may over-penalize benign formatting variations in strictly structured domains like source code. To provide a more structurally aware evaluation of model performance, we present additional experimental results utilizing CodeBLEU. Unlike standard n-gram metrics, CodeBLEU considers abstract syntax trees (AST) and semantic data flow, making it robust to whitespace and formatting differences that do not alter the underlying code logic.

Table 6 presents the CodeBLEU scores of representative open-source and proprietary models on our full code dataset across C++, Java, and Python at varying fragmentation contexts ($N = 8$, $N = 12$, and $N = 16$).

Discussion and Analysis:

As demonstrated in Table 6, transitioning to an AST-aware metric highlights significant disparities in structural code restoration capabilities. The proprietary models, specifically Gemini 3 Pro and Gemini 3 Flash, exhibit exceptional structural fidelity, consistently achieving the highest scores across all languages and context lengths. This validates their robustness in complex structural formatting tasks over standard string-matching methods.

Among the open-source candidates, the InternVL3.5 series maintains strong baseline perfor-

Table 6: Source Code Restoration evaluated using CodeBLEU (Higher is better).

Model	C++			Java			Python		
	$N=8$	$N=12$	$N=16$	$N=8$	$N=12$	$N=16$	$N=8$	$N=12$	$N=16$
Open-source Models									
InternVL3.5-8B	0.32	0.28	0.25	0.34	0.31	0.31	0.22	0.20	0.25
InternVL3.5-14B	0.35	0.28	0.28	0.38	0.33	0.34	0.24	0.22	0.26
InternVL3.5-38B	0.38	0.32	0.34	0.39	0.37	0.38	0.28	0.24	0.26
Mistral3-Reas-8B	0.26	0.21	0.21	0.33	0.31	0.31	0.22	0.22	0.22
Mistral3-Reas-14B	0.26	0.21	0.21	0.33	0.31	0.31	0.22	0.22	0.22
DeepSeek-OCR	0.23	0.18	0.21	0.24	0.18	0.19	0.16	0.15	0.16
Hunyuan-OCR	0.12	0.12	0.11	0.11	0.07	0.09	0.10	0.08	0.06
Proprietary Models									
GPT-5 Mini	0.26	0.22	0.19	0.34	0.27	0.31	0.17	0.19	0.22
GPT-5.1	0.28	0.20	0.21	0.33	0.26	0.29	0.23	0.20	0.25
Gemini 3 Flash	0.79	0.76	0.73	0.86	0.84	0.81	0.74	0.68	0.68
Gemini 3 Pro	0.83	0.79	0.79	0.87	0.85	0.83	0.77	0.77	0.71

mance (peaking at 0.39 on Java for the 38B model), effectively demonstrating positive

B Reproducibility and Evaluation Protocols

To ensure complete methodological transparency and facilitate future research, we detail the technical specifications of our evaluation pipeline and data generation process. We commit to open-sourcing our entire code repository—encompassing data generation, 3D rendering, and inference scripts—upon publication.

B.1 Model Inference Protocol

All evaluations were conducted using a consistent zero-shot system prompt. This prompt explicitly instructs the models to mentally "stitch" the fragments together and perform verbatim transcription, while strictly ignoring physical artifacts such as shadows, tears, and noise.

To ensure deterministic and reproducible outputs across all evaluated model APIs, the decoding temperature was set to zero (or the minimum supported value). Furthermore, a rigorous post-processing script was applied to the raw model outputs to strip non-content artifacts (e.g., markdown tags, extraneous whitespace). This guarantees that our evaluation metrics (ROUGE, NED, TEDS, and CodeBLEU) exclusively reflect the accuracy of the restored document content.

B.2 Data Generation and Physical Simulation

For the visual inputs, we adopted a "single composite image" approach. The unordered document fragments were rendered onto a 4096×4096 high-resolution canvas using the Blender Cycles engine with global illumination to simulate realistic scanning environments. The original text documents were initially rendered at a width of 1600px with a 28px font size. The final composite images were subsequently resized to a maximum dimension of 2048px for model inference, striking a balance between preserving fine-grained visual perception and adhering to the models' visual token limits.

The physical complexity of the shredded documents is governed by the following simulation parameters:

- **Spatial Arrangement:** Fragments were subjected to random Z-axis rotations ranging from 0° to 360°.
- **Irregular Boundaries:** Natural tearing edges were generated via Voronoi tessellation using $N \in \{8, 12, 16\}$ seed points.
- **Paper Deformation:** 3D physical artifacts were simulated using a Solidify modifier (thickness = 0.002) combined with a two-tier displacement strategy. Large-scale paper waves were generated using a Marble texture (noise scale = 1.5, strength = 0.15), while sharp micro-crumples were applied using a Musgrave texture (noise scale = 8.0, strength = 0.02).

C Ablation Study: Semantic Reasoning vs. Visual Matching

To determine whether models solve the fragmented document reconstruction task via semantic reasoning or by merely exploiting visual artifacts (e.g., edge matching), we conducted a controlled ablation experiment.

We generated a **Control Dataset** consisting of 50 documents using randomized "nonsense" text (e.g., "the circumstances eligendi..."). We strictly

Table 7: Comparison of Reconstruction Performance on Real vs. Nonsense Text ($N = 16$ Fragments). "Real" refers to the English News dataset, while "Nonsense" represents the randomized control text. Δ ROUGE denotes the absolute performance drop.

Model	**NED** (↓)		**BLEU** (↑)		**ROUGE** (↑)		**Δ ROUGE**
	Real	Nonsense	Real	Nonsense	Real	Nonsense	
Gemini 3 Pro	0.35	0.65	0.70	0.39	0.73	0.33	-0.40
Gemini 3 Flash	0.41	0.71	0.67	0.35	0.67	0.29	-0.38
Qwen-VL-Plus	0.65	0.75	0.28	0.06	0.38	0.13	-0.25
Qwen-VL-Flash	0.65	0.78	0.27	0.03	0.37	0.12	-0.25
GLM-4.6v	0.70	0.74	0.21	0.17	0.30	0.18	-0.12
GPT-5.1	0.80	0.81	0.03	0.00	0.15	0.08	-0.07
GPT-5 Mini	0.86	0.82	0.01	0.00	0.16	0.08	-0.08

preserved the exact layout, character length distribution, and font settings of the original English News dataset. The hardest fragmentation granularity ($N = 16$) was applied using our physics-based pipeline. Our hypothesis is straightforward: if models rely primarily on visual edge matching (jigsaw solving), their performance on "Nonsense" text should be comparable to "Real" text. Conversely, if they depend on semantic language priors, their performance on "Nonsense" text should collapse due to the absence of semantic context needed to bridge visual discontinuities.

We evaluated seven representative models under this setting. As shown in Table 7, performance dropped precipitously across all metrics when semantic meaning was removed. For instance, Gemini 3 Pro (the state-of-the-art model) achieves a high ROUGE score of 0.73 on real text, but this score collapses to 0.33 on nonsense text, accompanied by an NED degradation from 0.35 to 0.65. Gemini 3 Flash exhibits a similar decline (Δ ROUGE = -0.38).

Crucially, we observe a **Convergence of Failure**: on the nonsense dataset, all models degrade to a similarly low performance tier (NED ranging from 0.65 to 0.82). This indicates that without semantic cues, even the most capable models cannot effectively reconstruct the document based on visual features alone. The substantial performance gap confirms that visual artifacts are insufficient for reconstruction in SHREDBENCH, success necessitates strong semantic reasoning.