SYNTHFAILMD: FAILURE BY DESIGN FOR SYNTHETIC MULTI-DOCUMENT GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Enterprise knowledge agents increasingly operate over collections of documents with diverse modalities and artifacts, such as text, tables, charts, forms, images, and annotations. When these agents fail, the failures are often systematic, recurring under similar document and artifact conditions and providing a useful signal for evaluation. However, the enterprise documents underlying these failures are often private and cannot be reproduced as open benchmarks. We introduce **SynthFailMD**, a synthetic multi-document benchmark generator that constructs targeted document failure conditions without exposing private enterprise documents. SynthFailMD generates two connected documents at a time, along three dimensions: (1) **document realism**, using information graphs extracted from UniDoc-Bench across five domains to ground cross-document topics and relationships; (2) **visual realism**, using document recipes derived from real document distributions of tables, charts, text, and other artifacts in enterprise PDFs; and (3) **behavioral realism**, using 27 target failure types spanning seven artifact categories, including text, tables, charts, forms, images, annotations, and OCR. A document critique-and-repair (DCR) agent further reduces answer leakage, contradictions, and inconsistencies in generated documents. Using SynthFailMD, we construct 410 PDFs and approximately 3,798 question-answer pairs. Across matched control and failure-conditioned documents, accuracy drops from 70.70% to 57.06%, showing that targeted failure conditions produce measurable degradation. SynthFailMD provides a scalable, reproducible, and privacy-preserving approach for evaluating enterprise document intelligence systems.

1 INTRODUCTION

Document agents are becoming a fundamental interface for enterprise knowledge work. Modern systems must answer questions, generate summaries, and support decision making across collections of reports, presentations, contracts, forms, spreadsheets, and other business artifacts. Unlike traditional document understanding tasks, these workflows can require reasoning across multiple documents, where relevant information is distributed across interconnected sources (Cho et al., 2025; Opsahl-Ong et al.). Despite substantial progress in multimodal foundation models, evaluating document agents in enterprise settings remains an open challenge. Failures can depend on specific properties of the documents, yet the underlying enterprise documents are often confidential and cannot be redistributed as public benchmarks. As a result, organizations are often forced to rely on anecdotal examples and manual testing when validating document failures. Moreover, aggregate benchmark performance can reveal that an agent answered a question incorrectly without isolating which document property contributed to the failure, motivating controlled evaluations that vary document properties or perturbations (Chen et al., 2024; Verma et al., 2026).

Existing document benchmarks address only part of this problem. Prior work evaluates multi-page document understanding (Tito et al., 2023; Ma et al., 2024), multi-document reasoning (Cho et al., 2025; Suri et al., 2025; Opsahl-Ong et al.), and document parsing and robustness (Zhang et al., 2026; Zhou et al., 2026; Chen et al., 2024). While valuable, these benchmarks primarily evaluate models over existing document collections rather than constructing new document environments around specific failure conditions. In particular, they do not provide a mechanism for asking a different question: *given a specific document condition under which a model may fail, can we systematically construct a realistic multi-document environment that contains that condition and measure its effect?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Synthetic document generation offers a natural path toward controlled evaluation (Peper et al., 2025; Verma et al., 2026), but introduces a different challenge where generating a document is not the same as generating a realistic *document ecosystem*. Multi-document generation requires documents whose information is meaningfully related rather than independently generated around arbitrary topics. For example, a liquidity report may depend on accounts receivable collections, while a compliance review may depend on underlying operational metrics. These dependencies create the reasoning structures through which document failures emerge. To address this challenge, we introduce **SynthFailMD**, a failure-conditioned synthetic multi-document generation framework. SynthFailMD combines three forms of control: domain-specific information graphs ground the content and relationships across connected documents; empirically derived *document recipes* govern their visual and structural composition; and generation is conditioned on one of 27 target failure types spanning text, tables, charts, forms, images, annotations, and OCR (Appendix A.4). Together, these components capture *document realism*, *visual realism*, and *behavioral realism*. In other words, SynthFailMD controls *what* information is present, *how* that information is expressed, and *which* failure condition is introduced.

We observe that controlled synthetic document generation can introduce unintended errors such as answer leakage into surrounding text, duplicated information, contradictions, and incoherent narratives. Such errors can confound evaluation by making it unclear whether performance degradation results from the intended failure condition or from errors introduced during generation. We therefore introduce a *Document Critique-and-Repair (DCR)* agent that reviews generated document pairs and repairs leakage, consistency, contradiction, and coherence issues while preserving the intended document artifacts and failure condition. The resulting pipeline produces controlled document environments together with question–answer pairs and known ground truth.

Using this framework, we construct an initial benchmark of 410 PDFs and 3,798 question–answer pairs. For 14 of the 27 target failure types, we additionally construct matched control and treatment variants, allowing performance to be compared when the failure condition is absent vs. present. This framework design moves beyond measuring whether a benchmark is simply difficult; it enables us to test whether introducing a specific document condition produces measurable degradation in document-agent performance. Our contributions are as follows:

1. We introduce **failure-conditioned synthetic multi-document generation** as a framework for constructing controlled document environments around specific target failure conditions.
2. We develop **SynthFailMD**, which combines five domain-specific information graphs, empirically derived document recipes, and 27 target failure types across seven artifact categories to jointly model document realism, visual realism, and behavioral realism.
3. We introduce a **Document Critique-and-Repair (DCR)** stage that reduces unintended generation artifacts such as answer leakage, incoherence, inconsistencies, duplication, and contradictions that could otherwise confound failure-oriented evaluation.
4. We instantiate SynthFailMD to construct a benchmark of **410** PDFs and 3,798 question–answer pairs, including matched control–treatment variants for 14 failure types, and evaluate whether targeted failure conditions induce measurable degradation in document-agent performance.

2 Related Work

2.1 Document Understanding and Evaluation

A growing body of work evaluates multimodal understanding over documents containing text, tables, charts, and figures. SPIQA (Pramanick et al., 2024), WikiMixQA (Foroutan et al., 2025), and MP-DocVQA (Tito et al., 2023) study cross-modal and multi-page document reasoning, while VisDoMBench (Suri et al., 2025), M3DocVQA (Cho et al., 2025), and OfficeQA Pro (Opsahl-Ong et al.) extend evaluation to multi-document settings. Long-document understanding and retrieval are further studied by MMLongBench-Doc (Ma et al., 2024), M-LongDoc (Chia et al., 2025), LongDocURL (Deng et al., 2025), MMLongBench (Wang et al., 2026), and MMDocIR (Dong et al., 2025). MHier-RAG (Gong et al., 2026) and G^2-Reader (Du et al., 2026) explore retrieval and evidence organization over visually rich documents. Related work also evaluates document parsing and

Table 1: Comparison of SynthFailMD with related real and synthetic document benchmarks across multi-document, multi-artifact, and failure-conditioned settings.

	Related Work	Multi-Document	Multi-Artifact	Failure-Conditioned
Real	MP-DocVQA	✗	✓	✗
	VisDoMBench	✓	✓	✗
	M3DocVQA	✓	✓	✗
	UniDoc-Bench	✓	✓	✗
	ParseBench	✗	✓	✗
	MPDocBench-Parse	✗	✓	✗
	RoDLA	✗	✓	✗
Synthetic	SynthDocBench	✗	✓	✗
	MDBench	✓	✗	✗
	DocGenie / DocDjinn	✗	✓	✗
	SynthFailMD (ours)	✓	✓	✓

robustness. ParseBench (Zhang et al., 2026) and MPDocBench-Parse (Zhou et al., 2026) evaluate extraction of structural and visual elements, while RoDLA (Chen et al., 2024) studies robustness to document perturbations. UniDoc-Bench (Peng et al., 2025), which provides the source corpus for our information grounding and document recipes, contains real PDFs across diverse domains. These benchmarks primarily evaluate existing documents or transformations of them. In contrast, SynthFailMD generates connected document environments conditioned on target failure types and, for 14 failure types, provides matched control–treatment variants.

2.2 Synthetic Document and Multi-Document Generation

Synthetic generation provides an alternative to benchmarks constructed from existing document collections. SynthDocBench (Verma et al., 2026) generates controlled synthetic visual documents across properties such as length, layout, and modality composition. Closest to our multi-document setting, MDBench (Peper et al., 2025) generates multi-document reasoning tasks from structured knowledge, but represents documents as text rather than visually rendered, heterogeneous artifacts. DocGenie (Harikrishnan et al., 2025) and DocDjinn (Lamott et al., 2026) instead generate visually realistic synthetic documents that reproduce properties of real document distributions, but focus on individual documents. SynthFailMD combines multi-document and multimodal synthetic generation with failure conditioning. Information graphs ground relationships across documents, document recipes control their structural and visual composition, and target failure types determine the document condition being evaluated. This enables connected, visually rich document environments to be generated around controlled failure conditions.

3 SynthFailMD

We introduce SynthFailMD, a failure-conditioned framework for generating synthetic multi-document for controlled documen evaluation. Figure 1 provides an overview of the pipeline.

3.1 Problem Formulation

We formulate synthetic multi-document generation as constructing a *document ecosystem* conditioned on a target failure type. An ecosystem consists of a pair of documents $\mathcal{D} = \{D_1, D_2\}$ whose information is jointly required to answer questions in $\mathcal{Q}$. Let $e = (v_1, \rho, v_2) \in E$ denote a sampled relationship from an information graph, $r \in \mathcal{R}$ a document recipe, and $f \in \mathcal{F}$ a target failure condition. We define the SynthFailMD generation process as

$$\mathcal{G}(e, r, f) \to (\mathcal{D}, \mathcal{Q}), \tag{1}$$

where e determines the information and relationship across documents, r controls their visual and structural composition, and f determines the target artifact condition instantiated during generation.

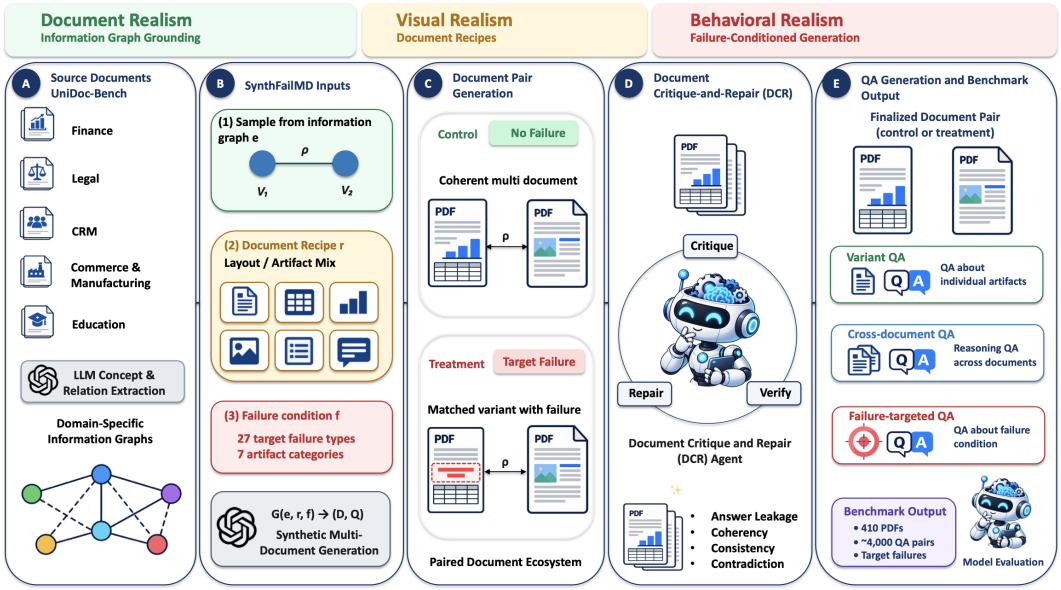

Figure 1: Overview of the SynthFailMD pipeline for generating synthetic multi-document.

These correspond to three dimensions of SynthFailMD: **document realism** through information-graph grounding, **visual realism** through document recipes, and **behavioral realism** through failure-conditioned generation.

3.2 Document Realism: Information Graph Construction

Realistic multi-document generation requires topic seeds that are meaningfully connected, rather than independently generated around related topics. Relying only on LLM-generated multi-document stories can result in relationships that lack diversity, grounding, and robustness. We therefore construct domain-specific information graphs from 1,017 documents in UniDoc-Bench (Peng et al., 2025). For each document in each domain, an LLM (GPT-5.6 Luna) extracts the primary concepts represented in the document and the semantic relationships between them. We represent the information graph for domain d as

$$G_d = (V_d, E_d), \tag{2}$$

where V_d is the set of concepts and E_d is the set of semantic relationships between concepts. Across five domains, the initial extraction produces 7,887 concepts and 5,713 relationships. We validate these using deterministic checks and an LLM judge (GPT-5.6 Sol), approving 715 documents (70.3%), containing 5,567 concepts and 4,289 relationships. We then canonicalize the graphs by merging semantically equivalent concepts and removing noisy or redundant elements, resulting in 4,942 concepts and 4,083 relationships across Finance, Legal, Education, CRM, and Commerce & Manufacturing. Additional construction, validation, and domain-level statistics are provided in Appendix A.1.

3.3 Topic Grounding via Graph Sampling

Because each SynthFailMD ecosystem contains two documents, we retain one-hop relationships from the canonical information graphs, yielding 2,304 candidate relationships across the five domains (detailed in Appendix A.2). To generate an ecosystem, we sample

$$v_1 \xrightarrow{\rho} v_2, \tag{3}$$

where v_1 and v_2 are related concepts and ρ denotes their semantic relationship. The source concept v_1 grounds the content of the first document D_1, while the target concept v_2 grounds the second document D_2. The relationship ρ provides the semantic connection between the two documents and determines how information in one document relates to information in the other. For example, *accounting and reporting compliance* $\xrightarrow{\text{informs}}$ *financial liquidity* produces two documents grounded in these concepts and connected through the *informs* relationship. Thus, each sampled edge provides both the document topics and their cross-document connection.

3.4 Visual Realism: Data-Driven Document Recipes

To generate documents with realistic visual and structural composition, we derive document recipes from artifact distributions observed in UniDoc-Bench (Peng et al., 2025). We estimate these distributions using a VLM-based tagging pipeline to identify document characteristics and artifact types, and a layout-based counting pipeline to measure elements such as text blocks, tables, charts, images, formulas, and form fields. The resulting distributions define four document profiles based on page length: *brief*, *standard*, *detailed*, and *comprehensive*, with artifact composition sampled from the corresponding empirical distributions. For benchmark generation, we use the *standard* recipe. Based on the observed Finance distribution, standard documents contain 8–12 pages and 6–10 text sections; tables occur in 84% of documents and charts in 65%, with an average of four of each when present. These statistics determine the document length and artifact composition used during generation. SynthFailMD additionally supports *brief*, *detailed*, and *comprehensive* recipes, allowing document complexity to be varied through the same data-driven procedure. The selected recipe therefore controls how information is presented and provides the basis for visual realism in SynthFailMD. Detailed recipe statistics in A.3.

3.5 Behavioral Realism: Failure-Conditioned Generation

Rather than generating documents first and identifying difficult cases afterward, SynthFailMD conditions generation directly on a target failure type. We define 27 failure types spanning seven artifact categories: text (6), tables (5), charts (5), images (5), OCR (4), forms (1), and annotations (1). These include challenges such as preserving reading order, reasoning over merged table cells, interpreting scanned charts, recognizing handwritten content, and understanding annotations. The complete taxonomy and definitions are provided in Appendix A.4. For a target failure type $f \in \mathcal{F}$, its definition determines which artifact carries the relevant information and how that information is represented. For example, a merged-cell condition places required information within a table whose interpretation depends on merged rows or columns. Thus, the failure condition is incorporated during document construction rather than assigned as a post-hoc label.

To isolate the effect of the target failure condition from the difficulty of the underlying content, we construct matched control and treatment variants for 14 of the 27 failure types. The complete set of paired failure types and their corresponding control and treatment conditions is provided in Appendix A.5. Each pair preserves the same underlying information while varying the presence of the target condition. For example, if the treatment contains a table with merged cells, the control presents the same information in an unmerged table. Similarly, if the treatment uses a two-column layout that introduces a reading-order challenge, the control presents the same content in a standard single-column layout. By varying the target condition while preserving the underlying information, these matched variants enable direct measurement of the performance degradation associated with introducing the failure condition.

3.6 Document Generation and Critique-and-Repair

Given an information-graph relationship, document recipe, and target failure condition, SynthFailMD jointly generates the corresponding documents. The sampled relationship is first grounded into paired report specifications and shared-key primary tables, which provide fixed evidence for generating the remaining report content and artifacts. The document recipe determines their structural and visual composition, while the target failure determines how the relevant artifact is instantiated. The resulting document specifications are then rendered as PDFs containing the required textual and visual artifacts. Further details of the LLM-based multi-document generation procedure

are provided in Appendix A.6. Synthetic generation can introduce unintended errors that are unrelated to the target failure. For example, narrative text may reveal answers intended to be recoverable only from a table, chart, or form; it may also duplicate information, contradict generated artifacts, or introduce inconsistencies or incoherent prose. These artifacts can confound evaluation by changing the intended information requirements of a benchmark instance. We therefore apply a Document Critique-and-Repair (DCR) agent, implemented with GPT-5.6 Terra, to each eligible generated document. DCR is restricted to narrative prose and does not modify generated artifacts or the target failure condition. It follows a *critique–repair–verify* procedure for at most two iterations, checking for answer leakage, contradiction, repetition, inconsistency, and incoherence. Text-failure cases are excluded from repair by design to avoid altering the failure being evaluated. The repaired document specifications are subsequently rendered into the final PDFs used for evaluation. Across eligible documents, DCR modified 190 of 210 (90%); among the repaired documents, answer leakage remained in 6.3%. Additional implementation details and repair statistics are reported in Appendix A.8.

3.7 Question–Answer Generation

For each generated document pair, SynthFailMD constructs question–answer instances across three categories: *variant QA*, *cross-document QA*, and *failure-targeted QA*. Variant QA evaluates information represented within individual document artifacts, cross-document QA requires integrating information distributed across the document ecosystem, and failure-targeted QA directly probes the document condition associated with the target failure. The final benchmark contains 3,798 questions: 1,796 variant QA questions, 1,166 cross-document QA questions, and 836 failure-targeted QA questions. Detailed QA types, reasoning operations, complexity levels, and examples are provided in Appendix A.11. Questions and ground-truth answers are constructed from the structured document specification during generation rather than inferred retrospectively from the rendered PDFs. Required values are computed deterministically from the underlying specification and verified for artifact presence and consistency prior to rendering. This ensures that ground-truth answers remain consistent with the generated document content while reducing reliance on LLM-generated answers.

4 Experimental Setup

4.1 Benchmark Construction

We instantiate SynthFailMD across five domains: Finance, Legal, Education, CRM, and Commerce & Manufacturing. For all experiments, we use the *standard* document recipe described in Section 3.4. The resulting benchmark contains 410 PDFs and 3,798 question–answer pairs covering 27 target failure types across seven artifact categories: text, tables, charts, images, OCR, forms, and annotations. Each document ecosystem consists of two connected PDFs grounded in a sampled relationship from the corresponding domain-specific information graph. For 14 of the 27 failure types, we additionally construct matched control and treatment variants for which a direct control condition can be reliably replicated while preserving the same underlying information. This paired setting allows us to measure how model performance changes when the target failure condition is introduced. Statistics about benchmark with count of treatmeant and control PDFs are in Appendix A.9.

4.2 Evaluation Settings

We evaluate SynthFailMD along two complementary dimensions: **QA Performance** and **Parser Performance**. For QA performance, we evaluate GPT-5.6 Sol, GPT-5.5, and GPT-5.4 under two document-access settings: *Canonical Markdown* and *VLM*. In the Canonical Markdown setting, the model receives the ideal Markdown representation produced during document generation, prior to PDF rendering. This representation preserves the document content and structure without requiring the model to recover them from the rendered document. In the VLM setting, each PDF is rendered into page images at 150 DPI with JPEG quality 85 and provided directly to the model with high image detail. Pages are presented in their original document order. All pages are retained; when a document exceeds the 25-image input limit, adjacent pages are combined vertically into grouped images rather than discarded. Canonical Markdown evaluates question answering and reasoning when document content is directly accessible, whereas VLM additionally requires visual interpre-

Table 2: Overall QA accuracy across models under Canonical Markdown and VLM document-access settings, with and without Document Critique-and-Repair (DCR). Δ denotes the VLM–Canonical Markdown difference in percentage points (pp).

	With DCR			Without DCR		
Model	**Markdown**	**VLM**	**Δ**	**Markdown**	**VLM**	**Δ**
GPT-5.6 Sol	55.06%	74.68%	+19.62	72.17%	83.70%	+11.53
GPT-5.5	52.50%	75.70%	+23.20	71.76%	85.12%	+13.36
GPT-5.4	51.76%	73.04%	+21.28	70.62%	83.76%	+13.14
Overall	**53.11%**	**74.47%**	**+21.36**	**71.52%**	**84.19%**	**+12.67**

tation of the rendered documents. Ground-truth answers are obtained from the structured document specification as described in Section 3.7. To evaluate the contribution of the Document Critique-and-Repair (DCR) agent, we additionally compare QA performance on documents generated with and without DCR agent under both Canonical Markdown and VLM settings. For parser performance, we evaluate four document parsers: Docling, MinerU, PyMuPDF4LLM, and pdfplumber.

4.3 Evaluation Metrics

For question answering, we report overall QA Accuracy by GPT-5.4-mini as judge against the question, expected answer, and corresponding assertion. The judge assigns one of three verdicts: PASS, PARTIAL, or FAIL, corresponding to scores of 1.0, 0.5, and 0.0, respectively. Instances marked as ERROR or SKIP are excluded from the calculation. We compute QA accuracy as

$$\text{Accuracy} = \frac{\text{PASS} + 0.5 \times \text{PARTIAL}}{\text{PASS} + \text{PARTIAL} + \text{FAIL}}. \tag{4}$$

This evaluation allows the judge to determine whether a predicted answer fully, partially, or unsuccessfully satisfies the required answer content. For comparisons between document-access settings, we report the difference between VLM and Canonical Markdown accuracy in percentage points. For parser evaluation, we report performance across eight dimensions: text quality, table cells, table structure, image F1, caption detection, element recall, reading order, and alignment. These metrics measure the parser's ability to recover the content and structural organization of the generated PDFs.

5 Results

5.1 Overall QA Performance

Table 2 reports overall QA performance under the Canonical Markdown and VLM settings. Across all models, VLM consistently outperforms Canonical Markdown, with an average gain of 21.36 percentage points with DCR and 12.67 points without DCR. This suggests that access to rendered pages provides useful visual information, such as layout, spatial organization, and artifact structure, that is not fully preserved in the Markdown representation. QA accuracy is also consistently lower after DCR, decreasing from 71.52% to 53.11% under Canonical Markdown and from 84.19% to 74.47% under VLM. Because DCR removes answer leakage and other unintended generation artifacts while preserving the target failure condition, this decrease suggests that unrepaired documents can make questions artificially easier and that SynthFailMD with DCR agent produces a more challenging evaluation setting. Details about three QA types and QA complexity are in Appendices A.10, A.11.

5.2 Parser Performance

Figure 2 compares four document parsers across eight extraction dimensions over all 410 generated PDFs. Performance is consistently high for text quality, table structure, and reading order, while substantially larger differences emerge for table-cell, image, caption, and element extraction. MinerU achieves the highest image F1 (77.02%) and caption detection (79.01%), whereas Docling performs strongly on table-cell extraction (79.93%). Element recall remains relatively low across all parsers, ranging from 22.90% to 45.03%. Some differences reflect parser output capabilities rather than extraction quality alone. PyMuPDF4LLM and pdfplumber do not provide structured caption

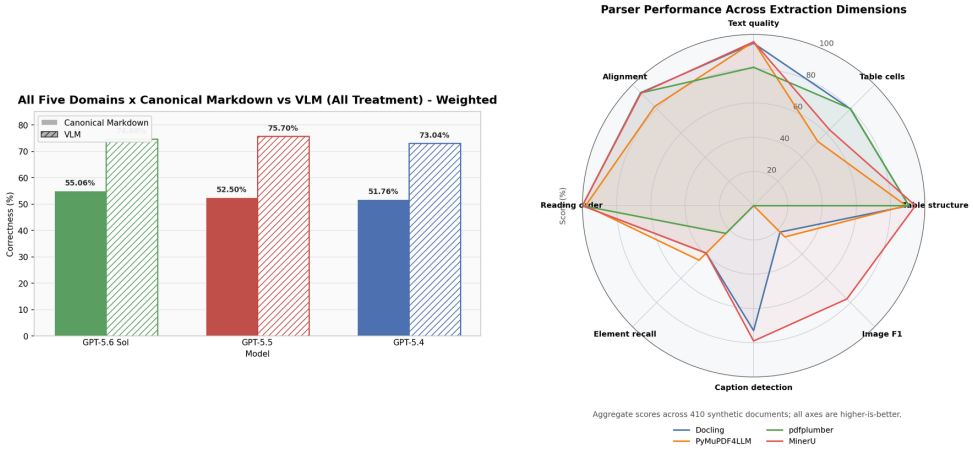

Figure 2: Document understanding and parsing performance. Left: QA accuracy under Canonical Markdown and VLM settings. Right: extraction quality across four document parsers and document elements.

outputs, and pdfplumber does not provide comparable visual-element extraction, resulting in zero scores for these dimensions. MinerU shows the consistent overall performance across the evaluated dimensions, particularly for image and caption extraction, while Docling performs better on table-cell extraction. Overall, the results demonstrate substantial variation in how existing parsers recover different structural and visual elements from the generated PDFs. Parser details in Appendix: A.12.

5.3 Synthetic Document Diversity

Visual Distribution. Figure 3 compares CLIP page embeddings from 315 real from UniDoc-Bench and 315 synthetic documents from SynthFailMD, sampled equally across the five domains. The UMAP projections show substantial overlap between real and synthetic pages, indicating shared visual characteristics, while distinct regions remain unique to each distribution. The first-principal-component distributions similarly overlap but exhibit a shift toward higher values for synthetic pages. Overall, SynthFailMD reproduces substantial visual variation observed in the sampled real documents while retaining measurable distributional differences. This analysis characterizes visual similarity rather than equivalence in document quality.

Synthetic Corpus Diversity. We additionally measure internal diversity across the 410 SynthFailMD PDFs using Vendi Score (Friedman & Dieng, 2022) over structural, topical, content, and combined representations in Figure 3. Because the RBF-based scores are sensitive to kernel bandwidth, we evaluate gamma multipliers from $0.25\times$ to $4\times$ the calibrated value. The normalized combined Vendi Score ranges from 16.6% to 72.3% across these settings, with structural and content diversity showing similar sensitivity. Topical diversity remains at 4.1%, due to most of the topics being synthetically generated by an LLM Because it uses cosine similarity, this score is unaffected by the RBF bandwidth. We therefore treat the RBF-based results as bandwidth-dependent effective-rank estimates rather than a single absolute measure of corpus diversity. Together, these analyses characterize complementary properties of SynthFailMD: its visual relationship to real documents and its internal structural, topical, and content diversity.

6 Ablation of Target Failure Conditions

We evaluate the effect of target failure conditions using the 14 failure types with matched control–treatment variants, where each pair preserves the underlying information while introducing the target failure only in the treatment. Figure 6 in Appendix A.13 visualizes the control-to-treatment changes, and Table 3 reports the corresponding results. Introducing target failures consistently reduces accuracy across all three models and both document-access settings. Overall Canonical Markdown accuracy decreases from 70.70% to 57.06%, a 13.63 percentage-point degradation, while VLM accuracy decreases from 79.64% to 74.35%, a 5.29-point degradation. All control–treatment differences are

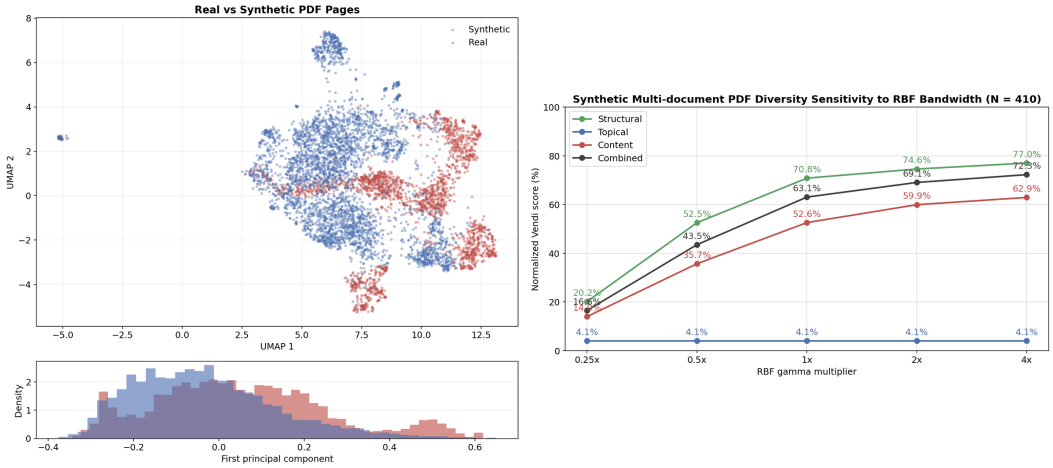

Figure 3: Visual similarity and internal diversity of SynthFailMD documents. Left: UMAP projection of CLIP embeddings for real and synthetic PDF pages. Right: normalized Vendi Scores for structural, topical, content, and combined document representations across RBF bandwidth settings.

Table 3: QA accuracy for matched control and treatment variants across the 14 paired target failure types. Δ denotes the degradation from control to treatment in percentage points (pp).

	Canonical Markdown				VLM			
Model	**Control (No Failure)**	**Treatment (Failure)**	**Δ (pp)**	p	**Control (No Failure)**	**Treatment (Failure)**	**Δ (pp)**	p
GPT-5.6 Sol	72.75%	59.38%	13.37	$< .0001$	79.86%	74.20%	5.65	$< .0001$
GPT-5.5	70.24%	55.96%	14.29	$< .0001$	79.45%	75.76%	3.69	.0018
GPT-5.4	69.09%	55.86%	13.23	$< .0001$	79.62%	73.09%	6.53	$< .0001$
Overall	**70.70%**	**57.06%**	**13.63**	$< .0001$	**79.64%**	**74.35%**	**5.29**	$< .0001$

statistically significant ($p \leq .0018$). The larger degradation under Canonical Markdown suggests that target failures have a stronger effect when models operate over the Markdown representation, while visual access partially mitigates their impact. Overall, the matched comparison shows that the injected failure conditions produce measurable performance degradation while holding the underlying document information fixed.

7 Conclusion

We introduced SynthFailMD, a framework for generating controlled, visually rich multi-document environments for evaluating document understanding under targeted failure conditions. SynthFailMD combines information-graph grounding, document recipes, failure-conditioned generation, and critique-and-repair to produce connected synthetic PDFs with structured ground-truth QA. Unlike benchmarks built primarily from existing document collections, the framework provides control over both the information relationships across documents and the document conditions under which models are evaluated. Our experiments show that target failures cause significant performance degradation across all evaluated models, while visual document access partially mitigates their impact compared with Markdown-only representations. We also find substantial variation across document parsers, particularly for visual and fine-grained elements, highlighting the role of the document-processing pipeline in downstream performance. Finally, our diversity analyses further show that the synthetic documents overlap visually with sampled real documents while retaining variation in their structure and content. Together, these findings highlight the value of controlled synthetic generation for studying document-agent failures. Rather than relying only on failures observed in existing corpora, SynthFailMD makes it possible to introduce specific failure conditions and measure their effects under controlled settings. Future work can extend SynthFailMD to larger document ecosystems containing more than two documents, additional failure types, and more diverse real-world document distributions.

REFERENCES

Yufan Chen, Jiaming Zhang, Kunyu Peng, Junwei Zheng, Ruiping Liu, Philip Torr, and Rainer Stiefelhagen. Rodla: Benchmarking the robustness of document layout analysis models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15556–15566. IEEE, 2024.

Yew Ken Chia, Liying Cheng, Hou Pong Chan, Maojia Song, Chaoqun Liu, Mahani Aljunied, Soujanya Poria, and Lidong Bing. M-LongDoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9233–9250, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.469. URL https://aclanthology.org/2025.emnlp-main.469/.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. M3docvqa: Multi-modal multi-page multi-document understanding. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 6237–6247. IEEE, 2025.

Chao Deng, Jiale Yuan, Pi Bu, Peijie Wang, Zhong-Zhi Li, Jian Xu, Xiao-Hui Li, Yuan Gao, Jun Song, Bo Zheng, and Cheng-Lin Liu. LongDocURL: a comprehensive multimodal long document benchmark integrating understanding, reasoning, and locating. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1135–1159, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.57. URL https://aclanthology.org/2025.acl-long.57/.

Kuicai Dong, Yujing Chang, Derrick Goh Xin Deik, Dexun Li, Ruiming Tang, and Yong Liu. MMDocIR: Benchmarking multimodal retrieval for long documents. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 30971–31005, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1576. URL https://aclanthology.org/2025.emnlp-main.1576/.

Yaxin Du, Junru Song, Yifan Zhou, Cheng Wang, Jiahao Gu, Zimeng Chen, Menglan Chen, Wen Yao, Yang Yang, Ying Wen, et al. g^2-reader: Dual evolving graphs for multimodal document qa. *arXiv preprint arXiv:2601.22055*, 2026.

Negar Foroutan, Angelika Romanou, Matin Ansaripour, Julian Martin Eisenschlos, Karl Aberer, and Rémi Lebret. Wikimixqa: A multimodal benchmark for question answering over tables and charts. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24941–24958, 2025.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

Ziyu Gong, Chengcheng Mai, and Yihua Huang. Mhier-rag: Multi-modal rag for visual-rich document question-answering via hierarchical and multi-granularity reasoning. In *International Conference on Intelligent Computing*, pp. 285–297. Springer, 2026.

PM Harikrishnan, Siddartha Reddy, Goutham Vignesh, Rohit Agrawal, and Vishal Vaddina. Docgenie: A framework for high-fidelity synthetic document generation via seed-guided multimodal llm and document-aware evaluation. In *Synthetic Data for Computer Vision Workshop@ CVPR 2025*, 2025.

Marcel Lamott, Saifullah Saifullah, Nauman Riaz, Yves-Noel Weweler, Tobias Alt-Veit, Ahmad Sarmad Ali, Muhammad Armaghan Shakir, Adrian Kalwa, Momina Moetesum, Andreas Dengel, et al. Docdjinn: controllable synthetic document generation with vlms and handwriting diffusion. *International Journal on Document Analysis and Recognition (IJDAR)*, pp. 1–16, 2026.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Advances in Neural Information Processing Systems*, 37:95963–96010, 2024.

Krista Opsahl-Ong, Arnav Singhvi, Jasmine Collins, Ivan Zhou, Cindy Wang, Ashutosh Baheti, Owen Oertell, Jacob Portes, Sam Havens, Erich Elsen, et al. Officeqa pro: An enterprise benchmark for end-to-end grounded reasoning, 2026. *URL: https://arxiv. org/abs/2603.08655*.

Xiangyu Peng, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, and Chien-Sheng Wu. Unidoc-bench: A unified benchmark for document-centric multimodal rag. *arXiv preprint arXiv:2510.03663*, 2025.

Joseph J Peper, Wenzhao Qiu, Ali Payani, and Lu Wang. Mdbench: A synthetic multi-document reasoning benchmark generated with knowledge guidance. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25592–25621, 2025.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. Spiqa: A dataset for multimodal question answering on scientific papers. *Advances in Neural Information Processing Systems*, 37:118807–118833, 2024.

Manan Suri, Puneet Mathur, Franck Dernoncourt, Kanika Goswami, Ryan A Rossi, and Dinesh Manocha. Visdom: Multi-document qa with visually rich elements using multimodal retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6088–6109, 2025.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multipage docvqa. *Pattern Recognition*, 144:109834, 2023.

Abhigya Verma, Khyati Mahajan, Amit Kumar Saha, Shruthan Radhakrishna, Sagar Davasam, Vikas Yadav, and Sai Rajeswar Mudumba. Synthdocbench: Controlled benchmark for long-context visual document understanding. *arXiv preprint arXiv:2607.10400*, 2026.

Zhaowei Wang, Wenhao Yu, Xiyu Ren, Jipeng Zhang, Yu Zhao, Rohit Saxena, Liang Cheng, Ginny Wong, Simon See, Pasquale Minervini, et al. Mmlongbench: Benchmarking long-context vision-language models effectively and thoroughly. *Advances in Neural Information Processing Systems*, 38, 2026.

Boyang Zhang, Sebastián G Acosta, Preston Carlson, Sacha Bron, Pierre-Loïc Doulcet, Daniel B Ospina, and Simon Suo. Parsebench: A document parsing benchmark for ai agents. *arXiv preprint arXiv:2604.08538*, 2026.

Bangbang Zhou, Hangdi Xing, Yifan Chen, Jianjun Xu, Qi Zheng, Feiyu Gao, Zhibo Yang, Shuai Bai, Ming Yan, Jieping Ye, et al. Mpdocbench-parse: Benchmarking practical multi-page document parsing. *arXiv preprint arXiv:2605.22100*, 2026.

A Appendix

A.1 Information Graph Construction and Validation

We construct domain-specific information graphs from 1,017 documents in UniDoc-Bench across five domains: Finance, Legal, Education, CRM, and Commerce & Manufacturing. For each source document, GPT-5.6 Luna extracts the primary concepts and semantic relationships between them, producing a document-level graph. Across all source documents, the initial extraction contains 7,887 concepts and 5,713 relationships.

A.1.1 Graph Validation

We validate each extracted graph using both deterministic checks and an LLM-based faithfulness assessment. Deterministic validation evaluates three properties. First, *structural validation* verifies the integrity of concepts, relationships, and graph structure. Second, *evidence grounding* verifies that extracted concepts and relationships are supported by evidence in the source document. Third, *graph quality analysis* checks graph-level properties such as connectivity, orphan nodes, and sparsity.

Graphs that satisfy the deterministic checks are additionally evaluated for semantic faithfulness using GPT-5.6 Sol as an LLM judge. The judge scores each graph on four dimensions using a 1–5 scale: *concept fidelity*, which measures whether extracted concepts accurately represent the source document; *theme coverage*, which measures coverage of the document's major themes; *relationship validity*, which evaluates whether graph edges represent valid semantic relationships; and *evidence quality*, which measures the strength of the supporting source evidence. Across the 1,017 evaluated document graphs, the average scores were 3.99 for concept fidelity, 3.38 for theme coverage, 3.13 for relationship validity, and 3.79 for evidence quality.

A.1.2 Graph Canonicalization

A graph is approved only if it passes structural validation, achieves an average LLM faithfulness score of at least 3.0, and no individual graph faithfulness dimension falls below 2. Under these criteria, 715 of the 1,017 document graphs (70.3%) are approved, containing 5,567 concepts and 4,289 relationships. We subsequently canonicalize the approved graphs by merging semantically equivalent concepts and removing redundant or noisy relationships. The resulting domain-level graphs contain 4,942 canonical concepts and 4,083 canonical relationships. Table 4 reports the complete domain-level statistics.

Table 4: Information graph validation results across domains. A graph passes validation if it satisfies the deterministic structural checks, achieves an average LLM faithfulness score of at least 3.0, and no individual faithfulness dimension falls below 2.

Domain	Documents	Passed	Failed	Pass Rate	Graph Score
CRM	163	112	51	68.71%	3.6 / 5
Education	62	42	20	67.74%	3.7 / 5
Finance	400	285	115	71.25%	3.5 / 5
Legal	301	205	96	68.11%	3.6 / 5
Commerce	91	71	20	78.02%	3.7 / 5
Overall	**1,017**	**715**	**302**	**70.30%**	**3.6 / 5**

A.1.3 Domain Graph Examples

The resulting domain graphs are also reusable resources for multi-document generation, providing grounded topic and relationship seeds for constructing connected document collections. Domain graphs are publicly available at https://synthfailmd.netlify.app/.

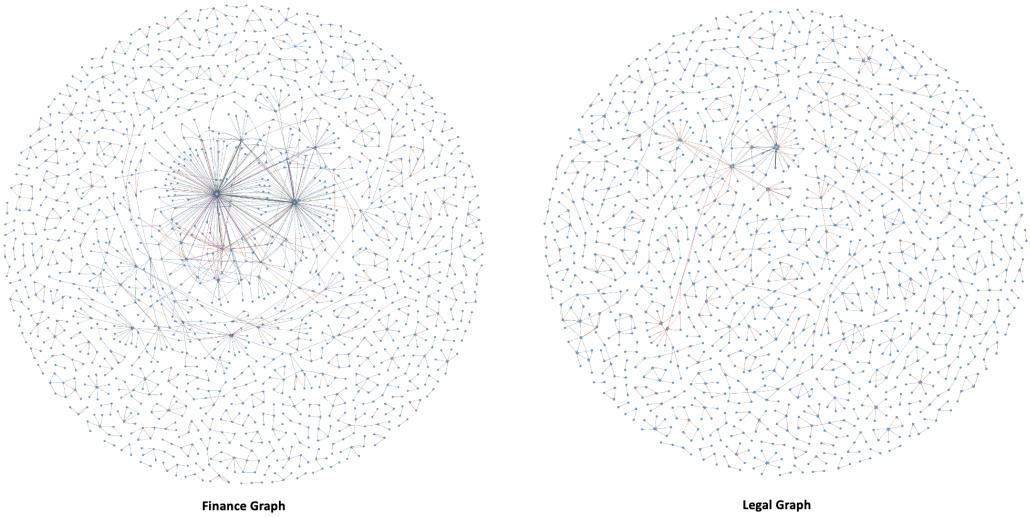

Figure 4: Domain information graph for finance and legal domain

A.2 Topic Grounding and Graph Sampling

SynthFailMD constructs document-pair templates from the canonical information graphs by filtering each domain graph to one-hop *informs* relationships,

$$v_1 \xrightarrow{\textit{informs}} v_2, \tag{5}$$

where v_1 and v_2 are canonical concepts. We use *informs* relationships to represent an information dependency between the two documents and select 10 such relationships from each of the five domains for benchmark generation. For each selected relationship, the source and target concepts determine the topics of the two reports. A report description is then constructed for each concept to specify the expected document content. The template additionally defines shared keys, such as reporting period, organization, country, region, or product, whose values are kept consistent across the document pair. These shared keys provide explicit dimensions for linking information across the two reports. The resulting template specifies the semantic relationship, report content, and cross-document keys used as constraints during document generation. Table 5 shows an example.

Table 5: Example transformation of a sampled information-graph relationship into a connected document-pair template.

Component	**Value**	**Role in Generation**
Graph Relationship	Accounts receivable $\xrightarrow{\textit{informs}}$ Financial position and liquidity	Provides the semantic relationship connecting the two documents.
Document 1	Accounts receivable collections	Report describing outstanding invoices, credit risk, disputed amounts, and collection activities.
Document 2	Treasury liquidity report	Report describing cash position, funding sources, credit utilization, FX exposure, and liquidity conditions.
Shared Keys	Reporting period, organization, region	Provides common dimensions for linking information across the reports.

A.3 Document Recipe Construction

Document recipes are derived from artifact distributions observed in UniDoc-Bench and specify the visual and structural composition used during synthetic document generation. Recipes are con-

Table 6: Example document recipes for the Finance domain. Artifact percentages indicate occurrence probability, and counts indicate the typical number of artifacts when present.

Recipe	Pages	Text Sections	Tables/ Count	Charts/ Count
Brief	5–7	4–7	59% / 3	71% / 2
Standard	8–12	6–10	84% / 4	69% / 4
Detailed	13–19	10–16	93% / 10	61% / 3
Comprehensive	20–30	15–25	97% / 13	69% / 4

structed separately for each domain and document profile (*brief*, *standard*, *detailed*, and *comprehensive*) using observed page-length and artifact distributions. For each artifact type, the recipe records its probability of occurrence and the observed count distribution when the artifact is present. Table 6 shows the all recipes for the Finance domain as an example. During generation, these statistics constrain document length and determine which artifacts are included and their approximate frequency. The same procedure is used to construct recipes for the remaining four domains.

A.4 Failure Type Taxonomy

SynthFailMD defines 27 target failure types across seven document artifact categories: text, tables, charts, images, OCR, forms, and annotations. These failure types capture distinct challenges that can arise when extracting, interpreting, or reasoning over heterogeneous document content. Table 7 provides the complete taxonomy and a detailed description of each failure type.

A.5 14 Paired Control Treatment Failure Types

For each paired RCA, control arm uses the original no failure PDF, whereas the treatment contains a localized failure. We construct matched control–treatment variants for the 14 failure types for which a direct control condition can be reliably replicated while preserving the same underlying information.

Table 7: SynthFailMD failure taxonomy. The taxonomy contains 27 target failure conditions across seven document artifact categories.

Category	Failure Type	Description
Text	Text extraction	Failure to correctly extract text from a document.
	Multi-section synthesis	Failure to integrate information distributed across multiple sections of a document.
	Cross-page reasoning	Failure to identify or reason over the correct information that appears across different pages or sections.
	Long-context retrieval	Failure to retrieve relevant information from long documents when the required evidence appears outside the immediately available context.
	Highlighted text	Failure to detect or associate visually emphasized text, such as highlighted content with its intended meaning.
	Reading order preservation	Failure to preserve the logical reading order of document text, resulting in interleaved, misplaced, or out-of-sequence content.
Table	Relational reasoning	Failure to correctly map a query to table dimensions, such as rows, columns, groups, or hierarchical relationships with misleading highlighted values.
	Merged cell handling	Failure to interpret the order and values of merged cells or multi-level headers, leading to incorrect associations between labels and values.
	Cross-page tables	Failure to reconstruct a table that continues across multiple pages while preserving its headers and structure.
	Column alignment	Failure to preserve row and column boundaries, causing values to be associated with incorrect headers or records.
	Blank/Missing values	Failure to recover blank/missing table values.
Chart	Value extraction	Failure to recover an explicitly displayed value from a chart or graph.
	Trend analysis	Failure to correctly interpret comparative patterns such as increases, decreases, rankings, or intersections.
	Scanned chart recognition	Failure to infer values from rasterized charts when the answer depends on visual properties such as bar height or line position.
	Chart type recognition	Failure to correctly identify the type of chart being presented.
	Legend interpretation	Failure to correctly interpret chart legends, axes, units, scales, or ranges.
Image	Descriptive extraction	Failure to correctly identify or describe salient objects, entities, properties, or spatial relationships in an image.
	Diagram reasoning	Failure to interpret relationships encoded through diagram structure, such as connectors, arrows, hierarchy, or grouping.
	Non-text visual content	Failure to use relevant information contained in non-text visual content when answering a question.
	Visual formatting semantics	Failure to interpret meaning conveyed through visual properties such as color, fill, line style, shape, or icons.
	Spatial layout understanding	Failure to correctly reason about the spatial position, alignment, or placement of document elements.
Form	Form interpretation	Failure to correctly interpret the state or value of interactive form controls such as checkboxes, radio buttons, dropdowns, or text fields.
OCR	Scanned text recognition	Failure to recognize machine-printed text contained in scanned or rasterized document regions.
	Scanned table recognition	Failure to reconstruct the rows, columns, headers, and values of a scanned or image-based table.
	Handwritten content	Failure to recognize handwritten text, numbers, signatures, or other manually entered content.
	Small/Graphical marks	Failure to detect graphical marks such as checkmarks, crosses, stamps, seals, or circled selections.
Annotation	Annotation understanding	Failure to detect, associate, or interpret document annotations such as comments, notes, tracked changes, or markup.

Table 8: Matched control–treatment variants for the 14 paired target failure types. Each pair preserves the underlying information while varying the presence of the target failure condition.

Category	Failure Type	Control	Treatment
Chart	Scanned chart recognition	Original chart with clearly rendered labels, values, legends, and visual elements.	Perturbed chart requiring values to be recovered from visual properties such as bar height or line position.
OCR	Scanned text recognition	Clean, digitally rendered machine-readable text.	Rasterized text requiring OCR to recover the relevant information.
	Scanned table recognition	Native table with recoverable rows, columns, headers, and values.	Rasterized table requiring OCR and reconstruction of its rows, columns, headers, and values.
	Handwritten content	Relevant information presented as clean machine-readable text.	Relevant information presented as handwritten text or numbers.
	Small/Graphical marks	Relevant information represented in a directly readable form.	Relevant information conveyed through graphical marks such as checkmarks, crosses, stamps, or circled selections.
Table	Relational reasoning	Table with the relevant row, column, or group clearly identifiable without misleading visual emphasis.	Table containing misleading highlighted values that can distract from the correct row, column, group, or hierarchical relationship required by the query.
	Merged cell handling	Table containing the same information without merged cells or multi-level headers.	Table containing merged cells or multi-level headers that require correct association of labels and values.
	Cross-page tables	Complete table presented without a page break.	Table divided across multiple pages, requiring reconstruction of headers and structure across the page boundary.
	Column alignment	Table with correctly aligned row and column boundaries.	Table layout in which altered column alignment can cause values to be associated with incorrect headers or records.
	Blank/Missing values	Table with relevant values explicitly populated.	Table containing blank or missing cells whose values must be correctly recovered or interpreted.
Text	Cross-page reasoning	Relevant information presented within a localized page or section.	Relevant information distributed across pages or regions, requiring correct localization and reasoning.
	Long-context retrieval	Required evidence positioned within readily accessible document context.	Required evidence positioned within a long document such that it must be retrieved from outside the immediately available context.
	Highlighted text	Relevant text presented without reliance on visual emphasis.	Relevant information identified through highlighting or other visual emphasis that must be correctly detected and associated with the text.
	Reading order preservation	Content presented in a standard single-column reading order.	Content presented in a layout that requires preservation of the correct reading sequence, such as a multi-column layout.

A.6 SYNTHETIC DOCUMENT GENERATION

Synthetic documents are generated from the information-graph relationships described in Section 3.3. For each connected document pair, the pipeline converts a sampled relationship into two enterprise reports while preserving the specified cross-document dependency.

1. Relationship grounding: A sampled relationship consists of a source concept, relationship type, and target concept. An LLM converts this relationship into source and target report specifications defining each report's title, purpose, and their semantic connection.

2. Primary table generation: An LLM generates one primary table for each report. The tables share a reader-facing key, such as reporting period, region, product, or business entity, with identical key values and ordering. Report-specific columns may differ. These tables establish the cross-document factual relationship and are fixed for subsequent generation.

3. Document planning: Given the report specification and fixed primary table, an LLM plans each report's theme, sections, and supporting artifacts. Primary-table values are deterministically extracted into a fact catalog that provides the factual evidence available during content generation.

4. Content and artifact generation: The LLM generates section prose, supporting tables, and charts according to the document plan and recipe. Content is conditioned on the assigned facts and table evidence, while the primary table remains unchanged. Generated tables are additionally checked for structural and cell-level validity.

5. Assembly and rendering: Generated content and artifacts are assembled according to the document specification and passed to the renderer to produce the final PDF. The LLM generates document content and artifact data but does not control the final PDF layout.

6. Control and treatment construction: For paired failure experiments, the generated document provides a shared base for the control and treatment conditions. The control retains the original content, while the specified RCA transformation is applied to the treatment. This preserves the underlying content and QA target while introducing the intended failure condition.

7. Model selection: Generation tasks are routed across three GPT-5.6 model tiers according to task complexity. **GPT-5.6 Luna** handles lightweight generation and extraction tasks such as titles, outlines, and section prose. **GPT-5.6 Terra** handles constrained structured generation, including tables, charts, document structure, and variant QA. **GPT-5.6 Sol** handles reasoning-intensive tasks, including document planning, coordinated fact construction, and multi-step table and QA reasoning.

For paired failure experiments, a single generated document serves as the shared base for both conditions. The control preserves this base unchanged, while the treatment is created by applying the specified RCA transformation to the corresponding artifact or content. The same underlying information and QA targets are retained across the pair, such that the injected RCA failure is the intended difference between the control and treatment documents.

Overall, the paired generation pipeline follows:

Information-graph relationship → paired report specifications → shared-key primary tables → table-grounded document plans → supporting content and artifacts → rendered PDFs → control/treatment variants.

A.7 Synthetic Document Examples

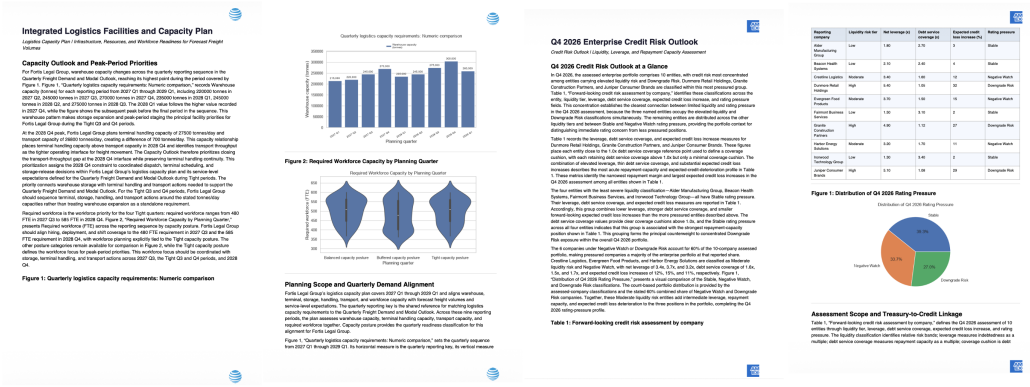

Figure 5: Synthetic multi-document examples

A.8 Document Critique and Repair Agent

The Document Critique and Repair (DCR) agent is applied after synthetic document generation to identify and repair unintended quality issues without modifying the artifacts or failure conditions being evaluated. DCR operates exclusively on editable texts. Tables, charts, questions, assertions, and injected RCA structures are not included in DCR.

1. Critique: For each generated case, DCR loads the document Markdown together with its generation specification, questions, and associated artifacts. An LLM (GPT 5.6 Terra) critiques the editable text for inconsistencies with the document brief and evidence. In parallel, deterministic checks identify answer leakage, where information intended to be recovered from an artifact is inadvertently stated in the surrounding prose. Detected leakage is added to the set of required repair issues.

2. Targeted repair: When repair is required, the LLM (GPT 5.6 Terra) generates replacements only for the affected text blocks. Proposed edits are applied only when they correspond to approved editable blocks and preserve the protected document structure. Cases with no detected issues bypass the repair stage.

3. Independent verification: After repair, a separate LLM call (GPT 5.6 Terra) verifies the candidate document for remaining answer leakage, factual inconsistencies, and unclear or unsupported prose. Deterministic leakage and structural-integrity checks are additionally applied. If repairable issues remain, DCR repeats the critique–repair–verification cycle on the updated document. We allow a maximum of two iterations and terminate earlier when verification passes or the remaining issue cannot be safely repaired.

4. Control–treatment consistency: For paired experiments, DCR operates on the treatment document and mirrors only safe prose edits to the corresponding control document. Control and treatment therefore remain identical in shared text after DCR, preserving the injected RCA as the intended experimental difference. If a treatment edit cannot be uniquely mapped to its corresponding control block, that edit is not propagated.

5. Rendering and fallback: Documents that pass verification are rendered back to PDF and checked for structural integrity. If required issues remain unresolved, document integrity is violated, or rendering fails, the original generated document is retained rather than publishing the repaired candidate. DCR records critiques, proposed and applied edits, verification outcomes, and fallback decisions to provide an audit trail for each case.

6. Scope: DCR is intentionally restricted to prose-level quality control and does not modify or validate failure-bearing artifacts. Consequently, text RCA cases are not included in DCR entirely, since modifying their text could alter the failure condition under evaluation.

7. DCR outcomes: Of the 270 treatment documents, 60 documents correspond to text failure cases and bypass DCR, leaving 210 eligible non-text treatment documents. DCR successfully edited 190 of these documents, while 20 document cases fell back to their original generated versions. Among the 190 edited documents, 12 retained answer leakage after repair, while 39 were classified as fully clean `repaired` cases. Table 9 summarizes these outcomes.

Outcome	Count	Share
Edited by DCR	190 / 210	90.5%
Fallback	20 / 210	9.5%
Remaining answer leakage	12 / 190	6.3%
Fully clean (`repaired`)	39 / 190	20.5%

Table 9: DCR outcomes for eligible treatment documents. Text RCA cases ($n = 60$) bypass DCR and are excluded from the eligible set. The final two rows are calculated over the 190 documents edited by DCR and are not mutually exclusive outcome categories.

A.9 SynthFailMD Benchmark Statistics

Table 10: Summary statistics for the SynthFailMD benchmark used in our experiments.

Benchmark Statistic	Value
Generation setup	
Domains	5
Failure types	27
Artifact categories	7
Failure types with matched controls	14
Documents per ecosystem	2
Document recipe used	Standard
Documents	
Treatment cases (document pairs)	135
Control cases (document pairs)	70
Treatment PDFs	270 (135 x 2)
Control PDFs	140 (70 x 2)
Total PDFs	**410**
Questions	
Variant QA	1,796
Cross-document QA	1,166
Failure-type QA	836
Total Questions	**3,798**

A.10 QA Performance by QA Type

The three QA types: **Cross-document QA** (1,166 questions) requires retrieving and combining information distributed across multiple documents. **Variant QA** (1,796 questions) evaluates reasoning over individual document artifacts, such as tables, charts, text, forms, or images, independent of the targeted failure condition. **Failure-targeted QA** (836 questions) directly evaluates artifact affected by the injected RCA failure and is designed to test whether the model can correctly interpret the failure-bearing artifact. Table 11 reports performance under the Canonical Markdown and VLM document-access settings. For cross-document QA, Canonical Markdown achieves 70.46% accuracy compared with 68.51% for VLM, a difference of 1.95 percentage points. This relatively small gap suggests that both representations support questions requiring information to be retrieved and combined across multiple documents. A different pattern emerges for questions that depend more directly on document artifacts. For variant QA, VLM achieves 82.88% accuracy compared with

64.93% for Canonical Markdown, a difference of 17.95 percentage points. The largest difference occurs for failure-targeted QA, where VLM achieves 56.28% accuracy compared with 14.44% for Canonical Markdown, a difference of 41.84 percentage points. This pattern is consistent across all three evaluated models for both variant and failure-targeted QA. Overall, neither document-access setting is uniformly advantageous across QA types. Canonical Markdown performs comparably to, and slightly better than, VLM on cross-document QA, whereas VLM substantially outperforms Canonical Markdown on variant and failure-targeted QA. These results indicate that access to the rendered document is particularly important for questions involving artifact-specific information, while cross-document reasoning can be performed effectively from the canonical textual representation.

Table 11: QA accuracy by question type under Canonical Markdown and VLM document-access settings. Results are evaluated on 270 DCR-agent treatment PDFs generated with the standard recipe and pooled across all five domains.

	GPT-5.6 Sol		GPT-5.5		GPT-5.4		Overall	
QA Type	**Markdown**	**VLM**	**Markdown**	**VLM**	**Markdown**	**VLM**	**Markdown**	**VLM**
Cross-document QA	69.04%	68.33%	69.67%	70.30%	72.67%	66.90%	**70.46%**	**68.51%**
Variant QA	69.15%	83.12%	63.70%	83.28%	61.95%	82.25%	**64.93%**	**82.88%**
Failure-targeted QA	15.17%	55.81%	14.93%	57.94%	13.22%	55.09%	**14.44%**	**56.28%**

A.11 Cross-Document QA Complexity

Cross-document QA contains 1,166 questions spanning 12 complex relational and numerical reasoning operations. These operations require models to combine information across document pairs through joined lookups, filtering and aggregation, minimum or maximum selection followed by lookup, ranking, differences, and multi-condition reasoning. For example, a question may require identifying the entity with the maximum value in one document and retrieving its corresponding value from the second document, or filtering entities by a specified category before aggregating their corresponding values across documents. Table 12 reports the complete distribution of cross-document reasoning operations.

Operation	**Count**	**Description**
Joined lookup	268	Retrieve corresponding values for the same entity or period across two documents.
Category filter then aggregate	205	Filter entries by category and compute an aggregate such as a sum or average.
Minimum then lookup	103	Identify the minimum value and retrieve corresponding information from both documents.
Maximum then lookup	101	Identify the maximum value and retrieve corresponding information from both documents.
Top-k then aggregate	100	Select the top-k entities and compute an aggregate over their corresponding values.
Bottom-k then aggregate	97	Select the bottom-k entities and compute an aggregate over their corresponding values.
Difference	76	Compute a difference between values or periods and retrieve associated information.
Above-average then aggregate	66	Compute an average, select values above it, and aggregate the resulting entries.
Dual-condition intersection	50	Identify entities satisfying conditions specified across both documents.
Filter, join, and compare	42	Filter entries in one document, join them with the second document, and compare the resulting values.
Rank then lookup	31	Identify an entity by rank and retrieve its corresponding values across documents.
Cross-table derived extremum	27	Compute a derived value across tables, identify its extremum, and retrieve the associated values.
Total	**1,166**	

Table 12: Complexity of cross-document QA operations in SynthFailMD.

A.12 Parser Performance Details

Table 13: Parser extraction quality on all 410 synthetic PDFs (higher is better; best per row in bold). Text quality is 1− normalized edit distance. Image, caption, OCR, and reading-order metrics are computed only on documents where they apply (n/a: field not exposed by the parser). An OCR score of 0% means the parser returns no text from inside images.

Metric	Docling	PyMuPDF4LLM	pdfplumber	MinerU
Text				
Text quality (1−NED)	94.88%	**95.73%**	80.82%	95.70%
Text BOW-F1	96.21%	96.47%	86.21%	**96.94%**
Reading order	**100.00%**	98.12%	**100.00%**	**100.00%**
Tables				
Table F1	**64.34%**	**64.34%**	54.67%	64.25%
Table cell BOW-F1	79.93%	53.05%	**80.10%**	62.70%
Table structure	90.74%	89.09%	90.70%	**94.49%**
Alignment	**93.32%**	81.87%	93.14%	92.94%
Images and OCR				
Figure F1	21.81%	25.80%	0.00%	**77.02%**
Caption found	72.95%	0.00%	n/a	**79.01%**
OCR text BOW-F1	**35.40%**	0.00%	n/a	0.00%
Coverage				
Element recall	39.18%	**45.03%**	22.90%	39.13%

A.13 Figure for Ablation of Target Failure Conditions

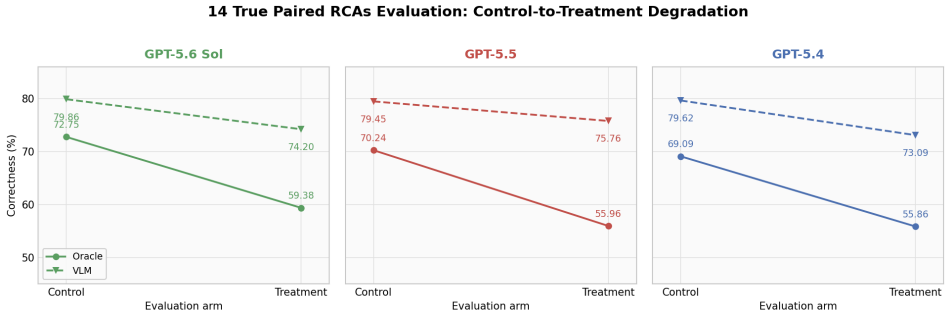

Figure 6: Control-to-treatment performance degradation for the 14 failure types with matched variants. Solid lines denote Canonical Markdown performance and dashed lines denote VLM performance.

A.14 Cross-Family Evaluation

To examine whether failure-difficulty profiles generalize beyond the primary model family, we additionally evaluate DeepSeek V4 Pro, Llama 4 Maverick, and Kimi K2.5 on the 270 treatment PDFs. Because matched controls were not collected for these additional models, this experiment measures treatment difficulty and RCA failure signatures rather than control-to-treatment degradation. Conditional scores exclude unscored responses and must therefore be read together with scored coverage. These results characterize treatment difficulty, not control-to-treatment degradation.

Table 14: Cross-family treatment-only results. Score intervals use case-cluster bootstrap resampling.

Model	Scored/attempted	Coverage	Conditional score [95% CI]	End-to-end score [95% CI]
GPT-5.6 Sol	3,798/3,798	100.0%	74.7% [72.4, 76.9]	74.7% [72.4, 76.9]
DeepSeek V4 Pro	3,793/3,798	99.9%	15.8% [14.2, 17.4]	15.8% [14.2, 17.4]
Llama 4 Maverick	1,580/3,798	41.6%	57.9% [54.9, 60.9]	24.1% [19.1, 29.2]
Kimi K2.5	3,256/3,798	85.7%	67.8% [65.1, 70.4]	58.1% [53.2, 62.8]

Table 15: Pairwise common-support Spearman correlations between RCA difficulty profiles. Holm correction is over the six model pairs.

Model A	Model B	Shared Q	RCAs	ρ	Holm-adjusted p
GPT-5.6 Sol	DeepSeek V4 Pro	3,793	27	0.357	0.133
GPT-5.6 Sol	Llama 4 Maverick	1,580	26	0.459	0.081
GPT-5.6 Sol	Kimi K2.5	3,256	27	0.734	2.4×10^{-4}
DeepSeek V4 Pro	Llama 4 Maverick	1,578	26	0.415	0.105
DeepSeek V4 Pro	Kimi K2.5	3,251	27	0.085	0.666
Llama 4 Maverick	Kimi K2.5	1,249	26	0.657	0.002

A.15 JUDGE ROBUSTNESS

We additionally assess QA accuracy evaluator robustness on a blinded 150-item subset using GPT-5.4-mini, Claude Sonnet 5, and Gemini 3.8 Flash.

Table 16: Verdict counts on the blinded 150-item robustness subset.

Judge	PASS	PARTIAL	FAIL
GPT-5.4-mini	74	24	52
Claude Sonnet 5	94	14	42
Gemini 3.8 Flash	96	16	38

All three judges agree on 114 of 150 items (76.0%).

Table 17: Pairwise agreement between judges on the blinded subset.

Pair	Exact three-class	Cohen's κ	PASS/non-PASS
GPT–Claude	79.3%	0.643	84.0%
GPT–Gemini	78.0%	0.620	85.3%
Claude–Gemini	94.7%	0.897	97.3%

A.16 PAIRED FAILURE ANALYSIS

For each model, paired ecosystem, and document-access setting, degradation is the mean QA score on the matched control minus the matched treatment, using PASS$= 1$, PARTIAL$= 0.5$, and FAIL$= 0$. The analyses retain complete valid question pairs and use paired case-cluster bootstrap intervals over ecosystems. Each RCA contains only five paired ecosystems, one per domain; consequently, RCA-level intervals are descriptive effect-size summaries rather than standalone significance tests.

For matched control–treatment experiments, we define degradation as

$$\Delta = 100(\mathrm{Score}_{\mathrm{control}} - \mathrm{Score}_{\mathrm{treatment}}),$$

where positive values indicate performance degradation. We report paired 95% confidence intervals using 20,000 ecosystem-level bootstrap resamples. Since each RCA contains five paired ecosystems, RCA-level effects are treated as descriptive. For cross-family evaluation, we compare RCA-level treatment difficulty using Spearman rank correlation over questions scored by both models.

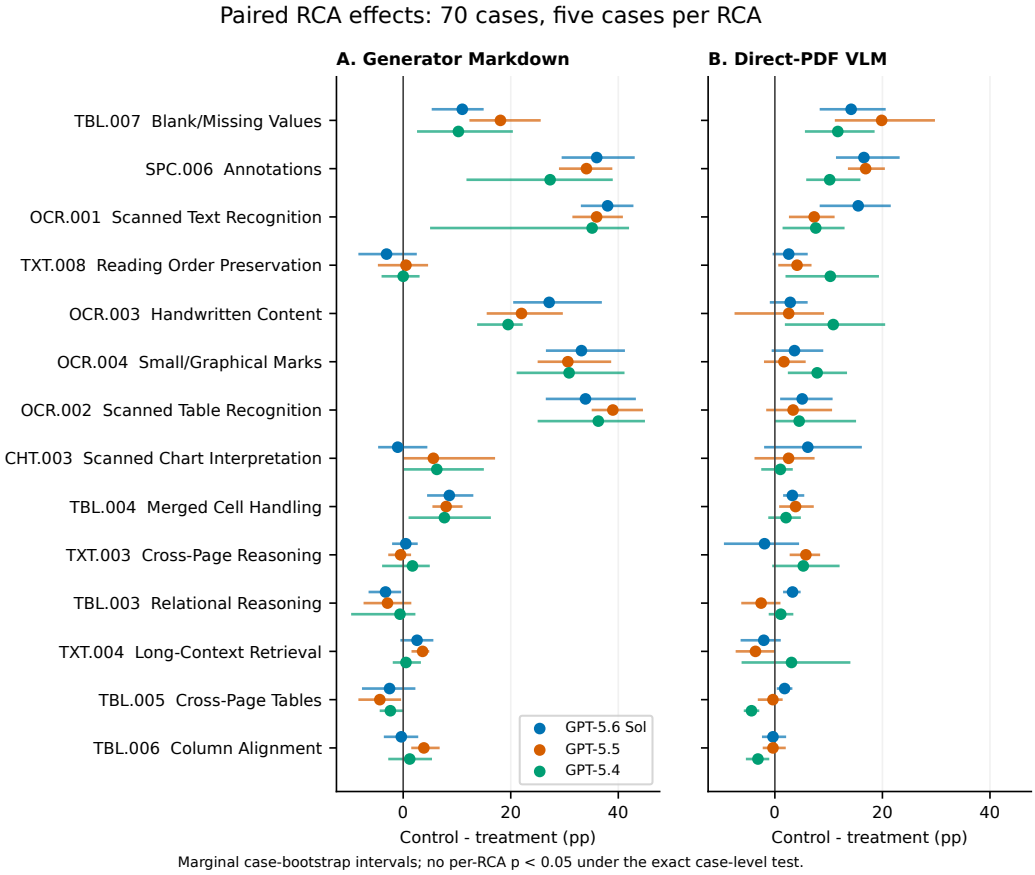

Figure 7: Per-model control-to-treatment degradation for all 14 paired RCA types under Canonical Markdown and direct-PDF VLM access. Points separate the three primary models; whiskers are paired 95% whole-case bootstrap intervals. Positive values indicate lower treatment performance. With five ecosystems per RCA, the minimum exact two-sided case sign-flip p-value is 0.0625.

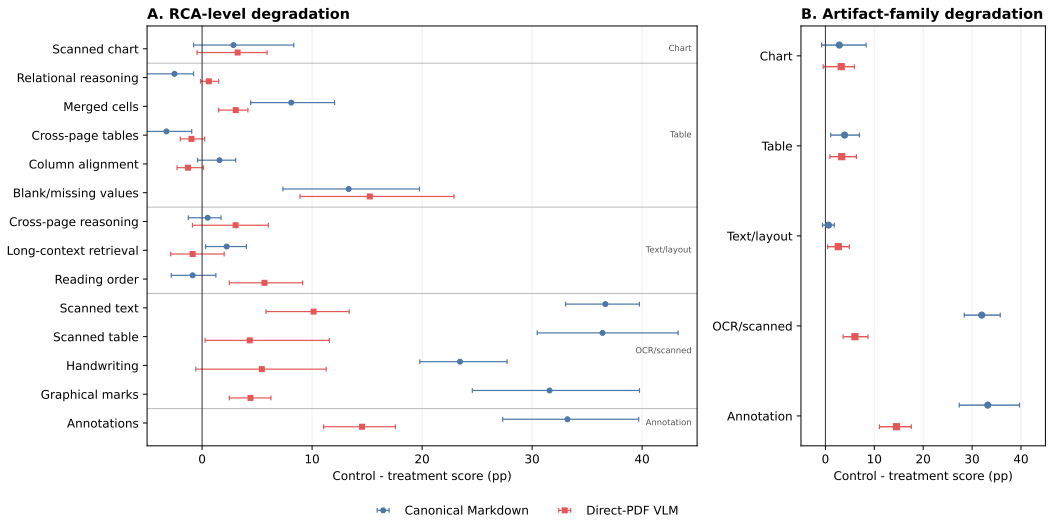

Figure 8: **Failure-specific paired intervention effects.** (a) Control-to-treatment degradation for each of the 14 RCA types with matched variants. (b) The same paired effects aggregated by artifact family. Positive values indicate lower treatment performance. Points show descriptive pooled estimates across the three primary models and whiskers show paired 95% case-cluster bootstrap intervals. Each RCA contains five paired ecosystems, one per domain

A.17 Realism and Distribution Analysis

For enterprise-domain classification, we encode rendered pages with a frozen `openai/clip-vit-base-patch32` model, mean-pool page embeddings per document, and train a held-out linear classifier to distinguish UniDoc documents from selected academic/Wikipedia negatives. The classifier achieves ROC–AUC 0.9622. At its 0.5 decision threshold, 397 of 410 SynthFailMD PDFs (96.8%) fall on the UniDoc side. This measures similarity to the selected reference distribution, not document authenticity.

Table 18: Held-out enterprise-domain classifier validation. The calibration-selected threshold targets at most 5% false positives on the calibration negatives; this target is not guaranteed on the test set.

Metric	Threshold 0.5	Calibrated threshold
Threshold	0.5000	0.5088
Test PDFs	128	128
ROC–AUC	0.9622	0.9622
Average precision	0.9711	0.9711
Balanced accuracy	90.6%	89.1%
Enterprise recall	89.1%	85.9%
Non-enterprise specificity	92.2%	92.2%
False-positive rate	7.8%	7.8%

For the real–synthetic comparison, we use 315 UniDoc-Bench and 315 SynthFailMD documents, balanced at 63 per corpus in each domain and rendered through the same pipeline. Table 19 reports distances and group-disjoint classifier two-sample tests. All MMD permutation tests have $p = .001$.

For internal diversity, we compute structural, topical, content, and combined Vendi Scores over the 410 PDFs while varying the calibrated RBF bandwidth from $0.25\times$ to $4\times$. The combined normalized score ranges from 16.6% to 72.3%; the complete sensitivity curve is shown in Figure 3. Because the effective rank changes substantially with bandwidth, we do not report a single Vendi value as an absolute diversity estimate.

Table 19: Real–synthetic comparison in CLIP first-page embedding space.

Subset	$\mathbf{MMD^2}$ **[95% CI]**	**Energy [95% CI]**	**C2ST AUC**
All domains	0.060 [0.053, 0.067]	0.115 [0.103, 0.127]	0.997
Commerce & Manufacturing	0.061 [0.043, 0.079]	0.147 [0.113, 0.181]	0.992
CRM	0.098 [0.077, 0.118]	0.211 [0.172, 0.250]	0.998
Education	0.076 [0.058, 0.093]	0.171 [0.137, 0.206]	0.997
Finance	0.119 [0.095, 0.143]	0.249 [0.204, 0.293]	0.995
Legal	0.086 [0.066, 0.106]	0.191 [0.153, 0.229]	0.999