TL;DR - OmniDocBench is saturating. GLM-OCR scores 94.6%, PaddleOCR-VL hits 94.5%, Hunyuan reaches 94.1%. Three models above 94% on a 1,355-page benchmark - and yet every one of them breaks on real scanned documents. Our 1,331-page benchmark on scan-heavy chemistry PDFs tells a different story: hallucinated chemical dosages, spaced-letter artifacts, collapsed table structures, and models that cannot detect a blank page. The gap between benchmark performance and production reliability is not closing. It is hiding.

The saturation problem

In March 2026, LlamaIndex's Jerry Liu flagged what many practitioners had already noticed: OmniDocBench is saturating. The top-ranked open OCR models now cluster above 94% accuracy on the benchmark, with less than a percentage point separating the leaders.

Model	Params	OmniDocBench (reported)
GLM-OCR	0.9B	94.62
PaddleOCR-VL-1.5	0.9B	94.50
HunyuanOCR	1B	94.10
FireRed-OCR	2B	92.94
DeepSeek-OCR-2	3B MoE	91.09

When the top three models are within half a point of each other, the benchmark has stopped being a useful discriminator. But the problem runs deeper than score compression.

What OmniDocBench actually measures

OmniDocBench v1.5 contains **1,355 pages across 9 document types**. It uses exact-match evaluation - character error rate (CER) for text recognition and tree-edit-distance similarity (TEDS) for table structure. These are well-defined metrics, and the benchmark has been valuable. But the coverage has limits.

What is in OmniDocBench: academic papers, textbook pages, financial reports, government documents, newspapers, notes, and a few other categories. The

pages are generally clean, digitally sourced, and structurally predictable.

What is not in OmniDocBench:

- Complex financial presentations with nested tables and footnotes
- Dense legal filings with multi-column text and margin annotations
- Insurance intake forms with mixed handwriting and print
- Multi-language documents with CJK and Latin script on the same page
- Handwritten annotations on printed text
- Scan-heavy PDFs with degradation artifacts, stamps, and curved pages
- Near-blank pages (cover pages, separator sheets, blank backs of printed pages)

The Hacker News discussion around this saturation echoed what the r/rpa community has been saying for years: every solution works on demo data. The gap is always in the last mile, on the documents that matter most to the people processing them.

What our benchmark tests that OmniDocBench does not

We built a benchmark from 31 scanned chemistry PDFs - 1,331 pages of upper-secondary notes and worksheets. These are not clean digital documents. They are scanned pages with curved spines, stamps, handwritten annotations, chemical apparatus diagrams, particle models, multi-column tables, and formula-heavy layouts.

The full methodology is documented in our benchmark methodology post. What matters here is the failure modes that emerged - failure modes that a 94%+ OmniDocBench score does not predict.

Blank-page blindness

Scanned PDFs routinely contain near-blank pages: cover sheets, separator pages, blank backs of single-sided prints. A production pipeline needs to detect these and skip them. If a model hallucinates text onto a blank page, every downstream step - chunking, indexing, retrieval - ingests garbage.

In our full-50 workflow benchmark, **GLM-OCR detected 0 out of 3 blank pages.** It produced text output for pages that contained no meaningful content.

DeepSeek detected all 3. This is not a minor detail - it is the difference between a pipeline that silently pollutes its own index and one that handles edge cases correctly.

No public benchmark treats blank-page detection as a first-class metric.

Spaced-letter artifacts

When a model struggles with character spacing on scanned text, it often inserts spaces between every letter: "c a r b o n a t e" instead of "carbonate," "s u l f u r i c" instead of "sulfuric." These artifacts are invisible to character error rate because each character is technically correct - the spaces are just extra characters. But they break search, indexing, and any downstream NLP.

On our scan-heavy corpus, spaced-letter artifacts appeared across every model tested, at varying rates depending on scan quality and font size.

Fused and bunched-up tokens

The opposite failure: characters that should be separated get merged. "SubstanceUislikelytobealiquidat" instead of "Substance U is likely to be a liquid at." This happens when OCR models fail to detect word boundaries on justified or tightly kerned text.

Fused tokens are harder to catch than spaced letters because the resulting string is not obviously malformed - it just looks like a long, unfamiliar word. Downstream spell-checkers may flag it, but automated correction is unreliable without domain-specific dictionaries.

Chemical equation hallucination

This is where the hallucination problem becomes concrete and dangerous. On chemistry worksheets, models hallucinated:

- Dosages that did not appear in the source (80ml instead of 5-20ml on a label)
- Chemical formulas with wrong subscripts or coefficients
- Reaction equations with invented products
- Notation shifts between columns (values off by orders of magnitude)

These are not random garbage characters that a human reviewer would catch. They are plausible, correctly formatted chemical notation that happens to be

wrong. This is the silent failure mode that the community identifies as the number-one production risk with VLM-based OCR.

Table structure collapse

Complex tables with merged cells, multi-level headers, and parent-child cell relationships break consistently. OmniDocBench measures table structure via TEDS, but its table samples tend toward simple, regular layouts. On our chemistry worksheets - which include data tables with units in sub-headers, multi-row answer spaces, and tables embedded within question blocks - every model tested showed structural degradation.

The practical consequence: extracted data from complex tables cannot be trusted without manual verification. For financial tables, SEC filings, or construction invoices, this means the OCR step has not actually automated anything - it has just moved the manual work downstream.

Diagram-dependent pages

Some pages are only interpretable with reference to a diagram - an apparatus setup, a particle model, a reaction scheme. Text-only OCR output for these pages is semantically incomplete regardless of how accurately the text was transcribed.

This is a failure mode that no text-accuracy benchmark can capture, because the benchmark does not know that the text on the page is meaningless without the accompanying visual.

The evaluation methodology gap

The gap between benchmark scores and production reliability is partly a measurement problem. OmniDocBench uses exact-match metrics. Our framework uses a different approach, designed to catch the failure modes above.

Text artifact scoring

Instead of character-level accuracy, we score text artifacts - patterns that indicate OCR failure even when individual characters are correct.

Artifact type	Weight	Why
Duplicate lines	1x	Common but low-severity; usually a pagination artifact
Spaced-letter artifacts	2x	Breaks search and indexing; invisible to CER
Fake image references	3x	Model hallucinated a reference to an image that does not exist
Repeated suffix patterns	10x	Strong signal of model degeneration or looping

The weighting reflects production impact: a duplicate line is a nuisance; a repeated suffix pattern means the model has entered a failure loop and everything after that point is unreliable.

Coordinate-aware anchor matching

For models that produce bounding-box coordinates (Hunyuan, DeepSeek), we evaluate spatial accuracy using IOU (intersection over union) plus a center-inside check with 18-pixel tolerance. This matters because a model can extract the right text but assign it to the wrong region of the page - which breaks any downstream layout-dependent processing.

Per-archetype aggregation

We classify pages into archetypes (clean text-first, diagram-dominant, anchor-critical dense, and others) and aggregate metrics per archetype rather than across the full corpus. This prevents easy pages from masking hard ones. A model that scores 95% overall but 60% on diagram-dependent pages is not a 95%-accurate model for a corpus that contains diagrams.

Blank-page detection as a first-class metric

We report blank-page detection rate separately. A model that hallucinates text onto blank pages is unsafe for production use regardless of its accuracy on content-bearing pages.

What the five-model benchmark showed

We ran five models - GLM-OCR, FireRed-OCR, HunyuanOCR, DeepSeek-OCR-2, and dots.ocr-1.5 - across a full-50 page benchmark and a mixed-10 diagnostic set. The results are documented across our leaderboard post and benchmark methodology post.

The headline finding: **no single model wins.**

Model	Speed (sec/page)	Blank detection (of 3)	Strength	Main risk
FireRed-OCR	~3.4	Partial	Best balanced - lowest cleanup burden on text-first pages	Loses question-local visuals on diagram-dependent pages
GLM-OCR	~0.9	0/3	Fastest - best throughput for high-volume workflows	Noisier Markdown, blind to blank pages
HunyuanOCR	~6.6	Partial	Strongest grounded output - 1,517 visual anchors with coordinates	Slowest; high latency for interactive use
DeepSeek-OCR-2	~4.2	3/3	Best blank-page handling; second-strongest grounded workflow	Moderate speed, smaller coordinate vocabulary
dots.ocr-1.5	~3.8	Partial	Broadest scope - handles web, scene, and SVG-style content	Not the safest default for scan-heavy document OCR

Each model has a failure mode that another model handles well. GLM is fast but hallucinates on blank pages. Hunyuan is thorough but slow. FireRed is balanced but misses diagram context. DeepSeek catches blanks but has fewer coordinate anchors than Hunyuan.

This is not a leaderboard problem. It is a routing problem.

Why routing beats picking a winner

If no single model wins across all page types, the practical answer is not to pick the best average model. It is to route each page to the model that handles its specific characteristics best.

The concept is a **page-archetype router**: classify each incoming page by its layout characteristics (text density, diagram presence, blank probability, table complexity), then dispatch it to the model with the best observed performance on that archetype.

In our benchmarks, a routing strategy that assigned different models to different page types consistently outperformed any single model used across the full corpus. The details are in our workflow-fit guide, but the principle is straightforward:

- **Text-first pages** FireRed (cleanest Markdown, lowest cleanup cost)
- **Diagram-dependent pages** GLM or Hunyuan (better visual region preservation)
- **Coordinate-critical pages** Hunyuan (richest grounded output)
- **Mixed/unknown pages** DeepSeek (safest default with blank-page handling)

This is not theoretical. It is what emerged from running five models across 1,331 pages and comparing the output page by page.

What needs to change in OCR evaluation

The saturation of OmniDocBench is not just a benchmark problem. It signals a broader gap between how the field measures progress and how production systems actually fail. Here is what needs to change.

Semantic evaluation over exact-match

Character error rate tells you whether the right characters were extracted. It does not tell you whether the extracted text means the right thing. A model that produces "NaOH" when the source says "NaOH" scores perfectly. A model that produces "NaOh" loses points - but the semantic content is preserved. Conversely, a model that produces "KOH" when the source says "NaOH" scores well on character overlap (2 of 4 characters correct) while being factually wrong.

Semantic evaluation - does the extracted text preserve the meaning of the source? - is harder to define and harder to automate. But it is what production systems actually need.

Domain-specific benchmark slices

A single benchmark across 9 document types cannot predict performance on legal filings, financial tables, medical records, or scientific notation. The field needs curated benchmark slices for specific domains, maintained by practitioners in those domains.

Hallucination-specific metrics

No major public benchmark reports a hallucination rate - the frequency with which a model produces confident text that does not appear in the source image. This is the number-one concern in every production OCR thread on Hacker News, Reddit, and practitioner forums. It should be the first metric reported, not an afterthought.

Cost and latency alongside accuracy

The OmniAI OCR Benchmark (February 2025) was the only public benchmark to include cost and latency data. That benchmark is now static. A living benchmark that reports accuracy, speed, and cost per page would be more useful than another accuracy-only leaderboard.

Production failure modes as first-class test cases

Blank pages, spaced-letter artifacts, fused tokens, hallucinated notation, table structure collapse on complex layouts, diagram-dependent semantic completeness - these should be named, measured, and reported as separate metrics, not absorbed into a single accuracy score.

The bottom line

OmniDocBench told us which models are good at document OCR. It can no longer tell us which models are better, because the top performers have converged.

More importantly, it never told us which models fail - and how they fail - on the documents that production systems actually need to process. A 94% benchmark score is compatible with blank-page hallucination, spaced-letter artifacts on scanned text, collapsed table structures on complex layouts, and invented chemical notation.

Benchmarks tell you which model is best on average. Production tells you which model fails least on **your** pages. The gap between these two answers is where the real work happens.

Further reading:

- LLM vs OCR Is the Wrong Debate - Here's the Actual Taxonomy in 2026 - the four-tier taxonomy for understanding document text extraction in 2026
- How We Benchmark OCR Models on Scan-Heavy PDFs - the full methodology behind our 1,331-page benchmark
- Which OCR Model Fits Which Workflow in 2026 - the routing-first decision guide for choosing models by page type
- OCR SOTA Feb 2026 - Open Document AI Leaderboard - the broader market map and model shortlist builder