Analysis

Enterprise Document AI Cannot Live on Benchmark Scores Alone: Reliability, Cost, and Human Gates for Unified OCR–Table–Formula Small Models

HX Research

HX Research · In-depth Analysis · November 3, 2026

Enterprise document parsing is rapidly entering production for order reconciliation, label/structured leaflet extraction, form filling, and retrieval-augmented generation. Compact unified OCR models at about one billion parameters now jointly handle text, tables, formulas, and seals, and report leading scores on OmniDocBench-style benchmarks. We argue that production usefulness is not determined by benchmark scores alone, but jointly by task-level end-to-end reliability, unit page cost, and auditable human gates. We formalize task reliability, field-weighted recall, layout-drift loss, unit cost, and review capacity, compare pipeline systems with end-to-end vision–language models, and present a parse–gate–review algorithm with a fixed-seed illustration. Mechanism analysis shows that benchmark leadership systematically overstates production value when layout drift, field-level recall, and review capacity are left outside the same frame; small unified models lower deployment barriers but must be evaluated inside that frame.

Keywords: document parsing; OCR; OmniDocBench; table and formula recognition; layout drift; field-level recall; human gates; unit page cost; unified small models

Introduction

OpenBMB recently released MiniCPM-OCR 1.0: about 1.3B parameters under Apache 2.0, covering text, tables, formulas, charts, and seals, with a companion regional correctness model MiniCPM-OCR Jev. According to the official model card, it scores 93.68 on Real5-OmniDocBench and processes about 1.2 pages/s on an RTX 5090—roughly 1.49× PaddleOCR-VL-1.6 and 3.62× MinerU2.5 Pro—while the full pipeline still uses PP-DocLayoutV3 for layout before recognition and SDK assembly into Markdown/JSON[1–5]. In the same window, PaddleOCR-VL, MinerU2.5, olmOCR, and other ~1B document VLMs posted dense leaderboard claims[6–11]. For buyers, the surface story is simple: higher scores, fewer parameters, faster pages.

Usefulness is not that curve. Enterprises parse FMCG order sheets, distributor reconciliations, pharma leaflets, safety tables, and SOP scans: layouts change by season, fields have asymmetric costs, and review headcount is capped. OmniDocBench advances fair comparison across nine sources and multi-level metrics[12]; Real5-OmniDocBench adds physical degradations[13,14]. Benchmark pages are still not your schema; edit distance is not a missed amount. DocVQA and form benchmarks push toward task answers[15–18]; tables and formulas keep dedicated protocols[19–23]. Treating public totals as go-live gates substitutes average page quality for task reliability.

The historical arc matters. Classical OCR (e.g., Tesseract) still underpins many pipelines[24]; PubLayNet moved layout into deep detectors[25]; LayoutLM-family models brought text–layout–image pretraining to forms and receipts[26–28]; Donut and Nougat showed page-level sequence generation without an external OCR engine[29,30]. Tables and formulas matured as specialist modules with TEDS and Im2LaTeX-style protocols[19,21,22]. In 2024–2026, OmniDocBench/Real5 and a wave of ~1B document VLMs compressed industry talk into “pick from the board.” This article pulls the talk back to Eq. (6). OCRBench and v2 further show that “chatty multimodal” is not “stable table reading”[31–33]. Closed APIs often raise unit cost and data-egress friction, so gates become more—not less—necessary. Open small models restore the right to experiment in-domain; they do not grant a waiver from acceptance tests.

Three gaps dominate practice. First, evaluation mismatch: public suites optimize page- or module-level recognition, while production cares about field recall, cross-page consistency, and post-redesign drop. Second, cost fragmentation: GPU time, engineering, layout models, and reviewers sit in different budgets, so cost per successful page is rarely reported. Third, human gates are missing or set as ad-hoc sample rates, ignoring reject/defer theory that couples error–reject trade-offs with capacity[34–38].

Thesis. Enterprise document-parsing usefulness is jointly determined by task-level end-to-end reliability × unit page cost × auditable human gates—not by OmniDocBench-style scores. Small unified models lower deployment barriers, but without layout drift, field-level recall, and review capacity in one frame, benchmark leadership systematically overstates production value. Contribu-

tions:

1. Six numbered equations and a notation table uniting task reliability, weighted field recall, drift loss, unit cost, reject bands, and capacity.
2. A comparison of pipeline, end-to-end unified, and cascade-with-judge routes with supporting and contrary evidence.
3. An executable Document Parse–Gate–Review algorithm, a fixed-seed illustration, and five falsifiable forward judgments.

Problem Formalization and Notation

Task-level end-to-end reliability

Let page x require fields F={f_1,...,f_m} with truths y_f and predictions ŷ_f. Let I_f=1[match(ŷ_f,y_f)] under a business match rule, and weights w_f≥0 summing to 1. Task reliability is

$$R_{task} = \Sigma_f w_f \cdot E[I_f] \quad (1)$$

When a critical field's weight nears one, high page-level character accuracy can still yield low R_task—the formal source of "great benchmark, failed launch." DocVQA is closer to tasks but not to weighted schemas[15,39,40]; form benchmarks report entity F1 without loss weights[16,17,41].

Layout drift

With benchmark distribution D_0 and week-t traffic D_t,

$$\Delta_t = R_{task}(D_0) - R_{task}(D_t) \quad (2)$$

Enterprise Δ_t usually comes from template redesigns, new vendors, scanner changes, and mixed scripts—not only from degradations already listed in public suites[12,13]. Layout datasets such as PubLayNet, DocBank, and DocLayNet provide detection supervision but do not replace weekly in-domain audits[25,42,43].

Unit page cost

$$C_{page} = C_{compute} + C_{eng} + C_{human}, \qquad C* = C_{page} / R_{task} \quad (3)$$

Official throughput enters only C_compute; ignoring C_human overstates small-model savings[1,6,8].

Reject band and human gates

With quality score s∈[0,1] from a judge, agreement, or calibrated confidence[4,44,45],

$$\textit{accept if } s \geq \tau_{hi}\textit{; route to humans if } s \leq \tau_{lo}\textit{; else sample by policy} \quad (4)$$

This instantiates classical reject/selective prediction on fields[34,35]. Jev productizes regional correctness scoring[2,4].

Review capacity

$$N_{low} \leq K\textit{, else raise } \tau_{lo}\textit{, reduce auto-accept, or add staff—never silently treat overflow as correct} \quad (5)$$

Learning-to-defer work shows fatigue and bias belong in the objective[36–38,46,47]; explanations need not create complementary teams[48].

Usefulness objective

$$\textit{maximize } R_{task} / C_{page} \textit{ subject to } N_{low} \leq K \textit{ and critical miss rate} \leq \alpha \quad (6)$$

Benchmark gains are instruments, not the objective. Table 1 lists notation.

Table 1 | Main notation

Symbol	Meaning	Where
F, w_f, I_f, R_task	fields, weights, correctness, task reliability	Eq. (1)
D_0, D_t, Δ_t	benchmark vs live distributions; drift loss	Eq. (2)
C_compute, C_eng, C_human, C_page, C∗	compute, engineering, human, unit, effective cost	Eq. (3)
s, τ_hi, τ_lo	quality score and dual thresholds	Eq. (4)
K, N_low, α	capacity, low-score volume, critical miss cap	Eqs. (5)(6)

Analysis Frame: Three Routes and Four Layers

We split systems into representation → parsing → gates → feedback, and group products into three routes. Surveys cast the field as modular pipelines versus unified VLMs[49–51], with specialist surveys on tables, forms, and scientific graphics[22,41,52].

Route A (pipeline): layout (e.g., PP-DocLayout / DocLayNet-style detectors) → specialist OCR/table/formula → reading order. MinerU, classical PaddleOCR stacks, Docling, Marker, and GROBID (scientific PDF) sit here[5,53–59]. Route B (end-to-end unified): page image → Markdown/HTML/JSON. Nougat, Donut, GOT-OCR2.0, Fox, olmOCR, PaddleOCR-VL, MiniCPM-OCR, dots.ocr, Dolphin, LightOnOCR are in or near this family[1,6,9,11,29,30,60–63]. Route C (cascade + judge): parse, then score with Jev-like judges or a second VLM; low scores escalate[4,64].

Unified small models change C_compute and integration cost; they do not by themselves solve R_task and K.

Mechanism Analysis: Why Benchmark Scores Overstate Production Value

Page-level metrics hide field asymmetry

OmniDocBench's end-to-end suite uses normalized edit distance, BLEU, TEDS, and reading-order metrics with type/attribute slices[12]. Enterprise tasks are sparse: only a handful of fields enter rules or scorecards. If amount weight is 0.4 and twenty other fields share 0.6, one amount miss hits R_task like many other errors combined. PubTabNet's TEDS is still roughly uniform over table structure[19]; receipt/form F1 treats entities equally[16–18]. Benchmarks optimize near-uniform symbol error; production loss is highly asymmetric—aligned only when field importance is nearly flat.

Contrary cases exist: full-page ingest for RAG with near-uniform weights, or scientific PDF→markup where page quality ≈ the task[30,58]. Even then, Nougat reports

Table 2 | Three technical routes (figures are author/vendor self-reports, not independently reproduced here)

Route	Mechanism	Strengths	Weaknesses	Fit	Evidence
Pipeline	layout→experts→merge	swappable modules; localizable errors	cascading errors; integration cost	stable layouts; need boxes	MinerU; PP-DocLayout ms-level layout on T4
End-to-end unified	page→structure	simple deploy; small models cut barriers	hallucination/loops; weak coords	throughput; diverse layouts	MiniCPM-OCR ~1.2 pages/s; Nougat repetition
Cascade + judge	parse + score + defer	aligns with Eqs. (4)(5)	extra inference; judges err	high-cost fields; review teams	MiniCPM-OCR Jev; learning-to-defer

~1.5% repetition collapse (higher out of domain) and much slower generation than GROBID on digital-born PDFs[30]—stability and cost still enter Eq. (3).

Layout drift turns a one-shot score into an expired certificate

Δ_t in Eq. (2) is almost surely nonzero: promotional redesigns, regulatory leaflet updates, scanner swaps. Real5 narrows the clean-PDF vs dirty-capture gap[12,13]; PaddleOCR-VL-1.5 likewise stresses in-the-wild robustness[7], but "dirty images in the suite" ≠ "your next-quarter template." LayoutLM-style pretraining learns cross-layout invariance yet still needs in-domain fine-tuning for entity F1[26–28]; DocLayNet shows industrial diversity needs ongoing labels[43]. Benchmark scores proxy $R_task(D_0)$; without monitoring Δ_t, leadership is a certificate date.

Small unified models cut deployment barriers, not the need for gates

MiniCPM-OCR, PaddleOCR-VL (0.9B), and LightOnOCR (~1B) make single-GPU/edge deployment realistic and cut $C_compute$[1,3,6,11]. MinerU2.5's coarse-to-fine split attacks high-resolution efficiency[8]; parallel-token prediction further cuts latency[64]. Three structural risks remain for generative OCR: hallucination/loops[30–33]; weak localization raising C_eng; and non-local repair (whole-model finetune vs swapping a table expert). Donut proved OCR-free VDU while burying errors in decoding[29]. Cheaper pages raise traffic, so N_low hits K sooner. Judges like Jev (reported AUROC×100 93.6 on JevOCRBench)[2,4] are the right productization of s—but judges err and may share failure modes; softmax baselines still belong in reports[44,45].

Tables and formulas set the floor for task reliability

Table detection/structure remains hard under diverse layouts[19,20,22,65]; formula work spans Im2LaTeX to UniMER[21,23]. Unified decoders can be pulled up by text-heavy pages while structure regions fail. OmniDocBench's modular tracks help—if buyers only read one total, they discard that help[12]. FMCG reconciliations and pharma leaflets put high weight exactly on tables/multi-column regions; report R_task on those subsets. Scientific table/figure surveys likewise demand detection-to-application metrics[52].

Capacity and audit, not a slogan sample rate

Eq. (5) is a hard constraint. Workload-aware deferral and fatigue models show how to optimize under K[36–38,46]; oversight capacity is finite[47]. Auditable gates keep model/layout/judge/threshold versions, scores, and human edits for estimating Δ_t. The myth "90+ score ⇒ 5% QA is enough" fails when low-score mass is 20% and K covers 8%. Optimize decision consequences, not persuasive UI[48].

From "can read" to "can post": RAG and feature pipelines amplify errors

Parsed text feeds RAG, rules, and variable derivation. Reading-order errors corrupt chunk boundaries; a shifted quantity column biases time×product×status×value×statistic features. Surveys treat structured extraction as upstream of knowledge bases and RAG[49,50]; visually rich QA surveys stress multi-page multimodal demand[40,51]. Downstream loss is nonlinear: a decimal shift can be catastrophic while page edit distance moves <1%. Hence R_task must be defined before posting, not after decoding. Swapping to a higher-scoring unified model can even raise C_eng when post-processing regexes break. Fox and DocFusion attack localization/unification[61,66], but gains still clear Eqs. (1) and (3).

When benchmark scores nearly suffice

Benchmarks proxy well when (i) the task is near-uniform full-text digitization, (ii) traffic matches a public subset without redesign, (iii) α is loose, or (iv) digital-born PDFs use text layers (GROBID/Marker/Docling)[55,57,58]. Our critique targets extrapolating that regime to high-asymmetry, high-drift, high-α enterprise forms—common in FMCG and pharma[67,68]. Classical engines (Tesseract, PP-OCR, TrOCR) err differently from VLM hallucinations; calibrate s separately per error family[24,54,69].

A purchasable acceptance sheet

Minimum columns: in-domain pages and sampling; schema and w_f; R_task and critical miss rate; Δ on old/new templates; $C_compute$, C_human, $C*$; τ_lo/τ_hi and K; version hashes; readable failure cases. OCRBench's spirit of decomposition beats a single trophy number[31,32]; RVL-CDIP classification accuracy is not parsing reliability[70].

Table 3 | Metric mismatch between public suites and production goals

Public metric	Measures	Production need	Mismatch	Remedy
Page edit distance / BLEU	average symbol quality	weighted R_task	uniform ≠ asymmetric loss	schema-weighted eval
TEDS	whole-table structure	critical cell recall	totals hide fatal cells	critical-cell subset
Layout mAP	box detection	correct crops for OCR	high mAP, wrong table cut	end-to-end field eval
Real5 degradation score	imaging robustness	redesign robustness	physical ≠ semantic drift	weekly in-domain Δ_t
Pages/s	part of C_compute	C∗ = C_page / R_task	ignores humans/eng	unified cost ledger
Single total leaderboard	average over slices	worst business slice	averages hide floors	slice by type/field

Cost accounting: pages/s → C∗

Vendor throughput fixes hardware, concurrency, and acceleration—e.g., MiniCPM-OCR's ~1.2 pages/s on RTX 5090 over 1,651 OmniDocBench v1.6 images at concurrency 1[1]. Enterprise stacks add other GPUs, quantization, CPU layout, PDF rasterization, and object-store round-trips. Force POC reports of per-1,000-page compute/human costs plus R_task and critical misses, then C∗. olmOCR, MinerU, and PaddleOCR-VL optimize different objectives; one total cannot rank them[6,9,49,53]. Quantization/speculative decoding (e.g., reported ~55% end-to-end efficiency gains with DSpark) must regress R_task[1,2].

Algorithm: Parse–Gate–Review

Step 5 is the hard discipline, aligned with capacity-constrained deferral[38,46]. Conformal risk control can language α under exchangeability, which redesign breaks[71].

Illustration: Same Proxy Score, Different C∗

Fixed seed 20261103 (not a vendor benchmark). Two systems share proxy score 0.93 but differ in weights and gates. Daily 10k pages × 10 fields; two critical fields weight 0.35 each; eight others share 0.30. Critical auto-accuracy 0.90 (L) vs 0.93 (G); others 0.96. Compute 0.008 vs 0.010 yuan/page; 0.05 yuan per reviewed field; critical misses carry large loss. L reviews 5% uniformly; G reviews ~12% with priority on critical fields. Script: 'src/sim.py'.

After gates, R_task(L)≈0.927 and R_task(G)≈0.961. Compute+human C∗ may favor L, but including expected miss loss, G's total effective cost is about half—showing why pages/s alone mis-ranks systems. Figures 1–3 plot R_task vs critical accuracy, C∗ vs review rate, and overflow under capacity.

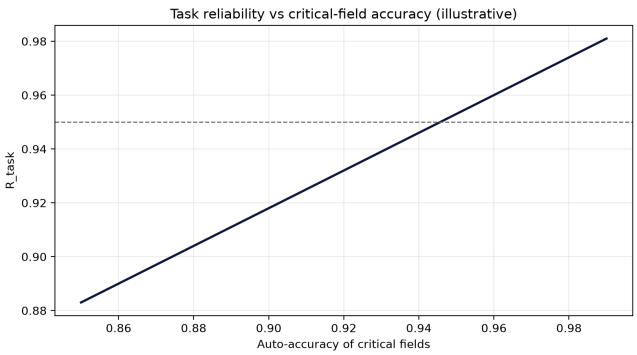


Fig. 1 | Task reliability vs critical-field auto-accuracy (illustrative). Weighted by Eq. (1); critical weights sum to 0.70.

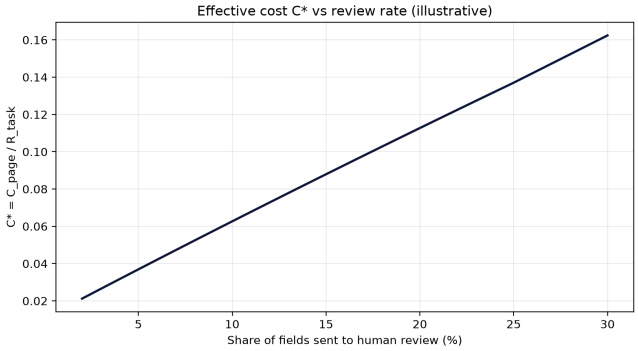


Fig. 2 | Effective cost C∗ vs human-review share (illustrative). Too little review under-delivers R_task; too much lets C_human dominate.

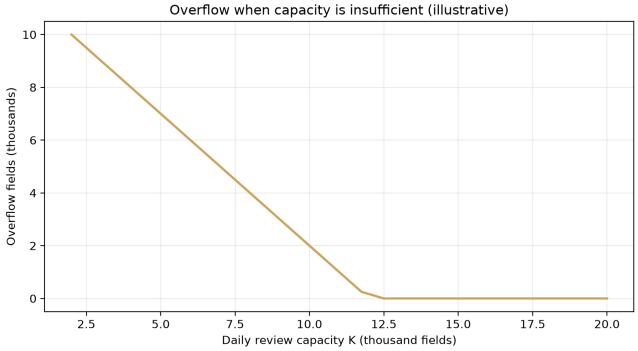


Fig. 3 | Overflow volume when N_low > K (illustrative). Silent auto-pass of overflow raises critical miss rates.

Data-governance interface (before posting)

In a variable-derivation setting, document fields are raw detail records that become explainable features only after time × product × status × value × statistic transforms[72]. If upstream parsing merges gift lines into primary SKU lines, downstream statistics will be precisely wrong. Scenario agents should treat parse–validate–post–exception as a supervised flow where human gates are nodes, not afterthoughts[73]. Multimodal understanding on a train-while-serving stack should emit scoreable structures rather than unauditable prose posing as fields[74]. A three-week MVP still applies: freeze schema/weights, run candidates, then dual-threshold pilot.

Algorithm 1 | Document Parse–Gate–Review (DPGR)

Input: pages X; schema F with weights w; validation set V; daily capacity K; critical miss cap α
Output: structures, gate logs, updated (τ_lo, τ_hi)

1: For each page: run layout (if any) and parser → regions and field candidates $\hat{y}$_f
2: Score each field/region s_f; missing fields get s_f=0
3: On V, choose (τ_lo, τ_hi) meeting miss rate $\leq\alpha$ and expected deferrals $\leq$K
4: Online: accept if s_f$\geq\tau$_hi; queue if s_f$\leq\tau$_lo; mid-band sample by residual capacity
5: If N_low>K: rank by w_f·(1−s_f), send top K to humans, mark overflow as pending capacity—never silent auto-pass
6: Write human edits back; estimate weekly Δ_t and C$*$; trigger finetune or rollback if Δ_t exceeds band

Complexity / fit: O(m) scoring per page (m = number of fields); needs a schema and small in-domain V; pure full-text ingest can degrade to page-level s

Table 4 | Illustration summary (seed 20261103; not measured production)

System	Proxy score	R_task	Review share	Compute	Notes
L: page-centric + 5% QA	0.93	≈0.927	5%	lower	higher loss-inclusive total
G: weighted dual thresholds	0.93	≈0.961	≈12%	slightly higher	~half loss-inclusive total
G with too-small K	0.93	drops	nominal 12%	same	α breach risk

Deployment: FMCG Forms and Pharma Leaflets, Not Another Leaderboard

Before FMCG orders and reconciliations feed variable derivation and channel models, fields must be trustworthy; missed gift lines bias five-dimension features[67,72]. RFPs should demand 200–500 in-domain field-labeled pages, weighted R_task, weekly Δ_t protocols, and gate logs for scenario agents—not only OmniDocBench screenshots[73]. In pharma, dose, contraindication, and batch fields face tight α; errors propagating into safety workflows dwarf extra inference cost[68]. Multimodal understanding on a train-while-serving stack should still wear Algorithm 1[74]. Product mention stays interfacial: VDC consumes trusted fields for derivation/governance; scenario agents orchestrate gated flows—they do not replace OCR benchmarks, they constrain OCR via Eq. (6).

A practical three-week MVP: week 1 freeze schema/weights and double-annotate ~300 pages; week 2 run candidates vs a pipeline baseline for R_task and C$*$; week 3 dual-threshold pilot measuring N_low and review hours. Shared KPIs must include R_task, C$*$, and α breaches so org incentives do not optimize "board score + thinner QA." Open stacks (MinerU, Docling, Marker, olmOCR, PaddleOCR-VL, MiniCPM-OCR) cut trial cost only if compared on one acceptance sheet[1,6,9,53,55,57]. Fine-tune when Δ_t is structured by layout clusters[25,42]; add judges/humans when errors look like generative hallucinations.

Five engineering rules: (1) schema + weights → R_task; (2) physical and semantic drift tests; (3) unified C$*$ ledger; (4) dual thresholds + capacity alarms; (5) pin parser/layout/judge/threshold versions. Pipelines and small models can coexist—pipelines for local repair, small models for cost—under Eq. (6), not leaderboard rank[1,6,49,53,55].

Forward Judgments

J1 (6–12 months). ~1B unified OCR models become default Chinese enterprise candidates, while contracts shift to in-domain field acceptance. Evidence: MiniCPM-OCR / PaddleOCR-VL / LightOnOCR release densification[1,6,11]. Falsify if public totals correlate >0.9 with field POC across industries for six months.

J2 (12 months). Parser + quality judge becomes standard for high-cost fields. Evidence: Jev and cascade ideas; deferral/capacity literature[4,38,64]. Falsify if ungated E2E stably meets α on critical fields.

J3 (12–18 months). Semantic layout-drift monitoring enters MLOps; physical-degradation headlines lose marginal value. Evidence: Real5 vs redesign incidents[13,43]. Falsify if vendors ship continuously refreshed suites matched to client templates.

J4 (18–24 months). C$*$ partly replaces pages/s in comparable quotes. Falsify if human review becomes negligible and error-free.

J5 (24 months). Cross-page field consistency becomes a disclosed shortfall; single-page totals no longer suffice to win bids. Evidence: multi-page DocVQA demand[39,40,61].

Responses to common views

"Top of the board is safest." — Tops average public symbol quality; correlation with R_task(D_t) is unproven. "Critical docs need larger models." — Criticality is α and gates, not parameter count. "We sample manually, so automation can be aggressive." — Without dual thresholds and capacity, sampling statistics are biased. "End-to-end killed pipelines." — Pipelines still win on local repair; many "E2E" products still call layout modules (e.g., MiniCPM-OCR + PP-DocLayoutV3)[1,5]. "Open weights cannot go to production." — Productionality is gates, versions, and audit—not closed labels. Application-layer metrics keep detection rates from becoming hallucinated prosperity[41,52].

Research Agenda

Open problems: sensitivity of w_f and α under tiny labels; cross-parser calibration of s without common-mode failure; automatic discovery of causal table-cell misses; audit liability when scenario agents rewrite downstream actions. Surveys list complex layouts, VLM reliability, and efficiency as open challenges aligned with Eq. (6)[49–51]. Whether synthetic layout augmentation reduces Δ_t—or only fits the augmenter—needs preregistered in-domain tests. Seasonal weight shifts and cross-vendor header alignment are engineering agenda items beyond pure OCR.

Limitations

We did not independently re-benchmark MiniCPM-OCR or competitors; figures marked "per vendor/paper" are theirs. The simulation is a structured hypothesis for Eqs. (1)(3)(5), not an industry mean. Handwriting-heavy forms, low-resource scripts, and air-gapped settings are out of scope. Product lines are interface context, not performance promises.

Conclusion

The core question is not who gained another OmniDocBench point, but whether unit cost per usable page is acceptable under real field weights, drift, and review capacity. Unified small models deserve serious evaluation for deployment and compute; without task reliability, cost ledgers, and auditable gates, benchmark leadership translates into false go-live confidence. Select and accept with Eq. (6); operate with Algorithm 1. Small models are worth using and benchmarks worth reading—only when reliability, cost, and human gates are present together does document AI move from demo to usable.

Data availability

This article produces no new experimental data. Cited figures come from the listed references, project pages and the company site; illustrative calculations are generated by scripts under the stated assumptions.

Code availability

Scripts for the illustrative calculations and figures are in src/sim.py (Python, fixed random seed), provided with this package for verification.

Competing interests

The authors are employees of Great Unicorn Intelligence Technology (Shenzhen) Co., Ltd. Company products mentioned here are its public products.

References

1. OpenBMB. MiniCPM-OCR 1.0 (model card). Hugging Face, `https://huggingface.co/openbmb/MiniCPM-OCR-1.0` (2026).
2. OpenBMB. MiniCPM-OCR: Fast, accurate OCR with correctness assessment. GitHub, `https://github.com/OpenBMB/MiniCPM-OCR` (2026).
3. ModelBest & OpenBMB. ModelBest and OpenBMB Open-Source MiniCPM-OCR Series. PR Newswire, `https://www.prnewswire.com/news-releases/modelbest-and-openbmb-open-source-minicpm-ocr-series-delivering-high-performance-document-parsing-and-ai-powered-quality-control-302925594.html` (2026).
4. OpenBMB. MiniCPM-OCR Jev (model card). Hugging Face / GitHub MiniCPM-OCR docs, `https://github.com/OpenBMB/MiniCPM-OCR` (2026).
5. PaddlePaddle Team. PP-DocLayout: A Unified Document Layout Detection Model to Accelerate Large-Scale Data Construction. Preprint at `https://arxiv.org/abs/2503.17213` (2025).
6. Cui, C. et al. PaddleOCR-VL: Boosting Multilingual Document Parsing via a 0.9B Ultra-Compact Vision-Language Model. Preprint at `https://arxiv.org/abs/2510.14528` (2025).
7. Niu, J. et al. PaddleOCR-VL-1.5: Towards a Multi-Task 0.9B VLM for Robust In-the-Wild Document Parsing. Preprint at `https://arxiv.org/abs/2601.21957` (2026).
8. Niu, J. et al. MinerU2.5: A Decoupled Vision-Language Model for Efficient High-Resolution Document Parsing. Preprint at `https://arxiv.org/abs/2509.22186` (2025).
9. Poznanski, J. et al. olmOCR: Unlocking Trillions of Tokens in PDFs with Vision Language Models. Preprint at `https://arxiv.org/abs/2502.18443` (2025).
10. Poznanski, J. et al. olmOCR 2: Unit Test Rewards for Document OCR. Preprint at `https://arxiv.org/abs/2510.19817` (2025).
11. LightOn. LightOnOCR: A 1B End-to-End Multilingual Vision-Language Model for State-of-the-Art Document Understanding. Preprint at `https://arxiv.org/abs/2601.14251` (2026).
12. Ouyang, L. et al. OmniDocBench: Benchmarking Diverse PDF Document Parsing with Comprehensive Annotations. Preprint at `https://arxiv.org/abs/2412.07626` (2024); CVPR 2025.
13. Real5-OmniDocBench authors. Real5-OmniDocBench: A Full-Scale Physical Reconstruction Benchmark for Document Parsing. Preprint at `https://arxiv.org/abs/2603.04205` (2026).
14. PaddlePaddle. Real5-OmniDocBench dataset and leaderboard. Hugging Face, `https://huggingface.co/datasets/PaddlePaddle/Real5-OmniDocBench` (2026).
15. Mathew, M., Karatzas, D. & Jawahar, C. V. DocVQA: A Dataset for VQA on Document Images. In *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision* 2200–2209 (2021); `https://arxiv.org/abs/2007.00398`.
16. Jaume, G., Ekenel, H. K. & Thiran, J.-P. FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. In *Proc. ICDAR Workshops* (2019); `https://arxiv.org/abs/1905.11640`.
17. Park, S. et al. CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In *Proc. Document Intelligence Workshop at NeurIPS* (2019).
18. Huang, Z. et al. ICDAR 2019 Competition on Scanned Receipt OCR and Information Extraction. In *Proc. International Conference on Document Analysis and Recognition* (2019).
19. Zhong, X., ShafieiBavani, E. & Jimeno Yepes, A. Image-based table recognition: data, model, and evaluation. In *Proc. European Conference on Computer Vision* (2020); `https://arxiv.org/abs/1911.10683`.
20. Li, M. et al. TableBank: Table Benchmark for Image-based Table Detection and Recognition. In *Proc. Language Resources and Evaluation Conference* (2020); `https://arxiv.org/abs/1903.01949`.
21. Deng, Y., Kanervisto, A., Ling, J. & Rush, A. M. Image-to-Markup Generation with Coarse-to-Fine Attention. In *Proc. 34th International Conference on Machine Learning* 980–989 (PMLR, 2017); `https://arxiv.org/abs/1609.04938`.
22. Kasem, M. S. et al. Deep Learning for Table Detection and Structure Recognition: A Survey. *ACM Comput. Surv.* **56**, 1–41 (2024); `https://doi.org/10.1145/3657281`.
23. Wang, B. et al. UniMER: Universal Mathematical Expression Recognition. Preprint at `https://arxiv.org/abs/2404.15254` (2024).
24. Smith, R. An Overview of the Tesseract OCR Engine. In *Proc. International Conference on Document Analysis and Recognition*

629–633 (2007); https://doi.org/10.1109/ICDAR.2007.4376991.

25. Zhong, X., Tang, J. & Yepes, A. J. PubLayNet: largest dataset ever for document layout analysis. In *Proc. International Conference on Document Analysis and Recognition* 1015–1022 (2019); https://arxiv.org/abs/1908.07836.
26. Xu, Y. et al. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *Proc. 26th ACM SIGKDD* 1192–1200 (2020); https://arxiv.org/abs/1912.13318.
27. Xu, Y. et al. LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding. In *Proc. 59th Annual Meeting of the Association for Computational Linguistics* 2572–2586 (2021); https://arxiv.org/abs/2012.14740.
28. Huang, Y., Lv, T., Cui, L., Lu, Y. & Wei, F. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In *Proc. 30th ACM International Conference on Multimedia* 4083–4091 (2022); https://arxiv.org/abs/2204.08387.
29. Kim, G. et al. OCR-free Document Understanding Transformer. In *Proc. European Conference on Computer Vision* (2022); https://arxiv.org/abs/2111.15664.
30. Blecher, L., Cucurull, G., Scialom, T. & Stojnic, R. Nougat: Neural Optical Understanding for Academic Documents. Preprint at https://arxiv.org/abs/2308.13418 (2023).
31. Liu, Y. et al. OCRBench: On the Hidden Mystery of OCR in Large Multimodal Models. Preprint at https://arxiv.org/abs/2305.07895 (2023).
32. Fu, L. et al. OCRBench v2: An Improved Benchmark for Evaluating Large Multimodal Models on Visual Text Localization and Reasoning. Preprint at https://arxiv.org/abs/2501.00321 (2024).
33. Shi, Y. et al. Exploring OCR Capabilities of GPT-4V(ision): A Quantitative and In-depth Evaluation. Preprint at https://arxiv.org/abs/2310.16809 (2023).
34. Chow, C. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theory* **16**, 41–46 (1970); https://doi.org/10.1109/TIT.1970.1054406.
35. Geifman, Y. & El-Yaniv, R. Selective Classification for Deep Neural Networks. In *Advances in Neural Information Processing Systems* **30** (2017); https://arxiv.org/abs/1705.08500.
36. Madras, D., Pitassi, T. & Zemel, R. Predict Responsibly: Improving Fairness and Accuracy by Learning to Defer. In *Advances in Neural Information Processing Systems* **31** (2018); https://arxiv.org/abs/1711.06664.
37. Mozannar, H. & Sontag, D. Consistent Estimators for Learning to Defer to an Expert. In *Proc. 37th International Conference on Machine Learning* (PMLR, 2020); https://arxiv.org/abs/2006.01862.
38. Alves, J. V. et al. Cost-Sensitive Learning to Defer to Multiple Experts with Workload Constraints. Preprint at https://arxiv.org/abs/2403.06906 (2024).
39. Tito, R., Karatzas, D. & Valveny, E. Hierarchical multimodal transformers for Multi-Page DocVQA. Preprint at https://arxiv.org/abs/2212.05935 (2022).
40. Barboule, C., Piwowarski, B. & Chabot, Y. Survey on Question Answering over Visually Rich Documents: Methods, Challenges, and Trends. Preprint at https://arxiv.org/abs/2501.02235 (2025).
41. Abdallah, A., Eberharter, D., Pfister, Z. & Jatowt, A. A survey of recent approaches to form understanding in scanned documents. *Artif. Intell. Rev.* **57**, 288 (2024); https://doi.org/10.1007/s10462-024-11000-0.
42. Li, M. et al. DocBank: A Benchmark Dataset for Document Layout Analysis. In *Proc. International Conference on Computational Linguistics* (2020); https://arxiv.org/abs/2006.01038.
43. Pfitzmann, B., Auer, C., Dolfi, M., Nassar, A. S. & Staar, P. DocLayNet: A Large Human-Annotated Dataset for Document-Layout Segmentation. In *Proc. 28th ACM SIGKDD* 3743–3751 (2022); https://arxiv.org/abs/2206.01062.
44. Hendrycks, D. & Gimpel, K. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *Proc. International Conference on Learning Representations* (2017); https://arxiv.org/abs/1610.02136.
45. Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. On Calibration of Modern Neural Networks. In *Proc. 34th International Conference on Machine Learning* 1321–1330 (PMLR, 2017); https://arxiv.org/abs/1706.04599.
46. Zhang, Z., Nguyen, C. C., Rosewarne, D., Wells, K. & Carneiro, G. Fatigue-Aware Learning to Defer via Constrained Optimisation. Preprint at https://arxiv.org/abs/2604.00904 (2026).
47. Turan, E. Oversight Has a Capacity: Calibrating Agent Guards to a Subjective, Fatiguing Human. Preprint at https://arxiv.org/abs/2606.08919 (2026).
48. Bansal, G. et al. Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance. In *Proc. 2021 CHI Conference on Human Factors in Computing Systems* 1–16 (ACM, 2021); https://doi.org/10.1145/3411764.3445717.
49. Zhang, Q. et al. Document Parsing Unveiled: Techniques, Challenges, and Prospects for Structured Information Extraction. Preprint at https://arxiv.org/abs/2410.21169 (2024).
50. Wang, W. et al. Document Intelligence in the Era of Large Language Models: A Survey. Preprint at https://arxiv.org/abs/2510.13366 (2025).
51. Ding, Y., Han, S. C., Lee, J. Y. H. & Hovy, E. H. Deep Learning based Visually Rich Document Content Understanding: A Survey. Preprint at https://arxiv.org/abs/2408.01287 (2024).
52. Huang, J., Chen, H., Yu, F. & Lu, W. From Detection to Application: Recent Advances in Understanding Scientific Tables and Figures. *ACM Comput. Surv.* (2024); https://doi.org/10.1145/3657285.
53. Wang, B. et al. MinerU: An Open-Source Solution for Precise Document Content Extraction. Preprint at https://arxiv.org/abs/2409.18839 (2024).
54. Du, Y. et al. PP-OCR: A Practical Ultra Lightweight OCR System. Preprint at https://arxiv.org/abs/2009.09941 (2020).
55. Livathinos, N. et al. Docling: An Efficient Open-Source Toolkit for AI-driven Document Conversion. Preprint at https://arxiv.org/abs/2501.17887 (2025).
56. Livathinos, N. et al. Docling Technical Report. Preprint at https://arxiv.org/abs/2408.09869 (2024).
57. VikParuchuri. Marker: Convert PDF to markdown quickly with high accuracy. GitHub, https://github.com/VikParuchuri/marker (2024).
58. GROBID. GeneRation Of BIbliographic Data. GitHub, https://github.com/kermitt2/grobid (2008–2024).
59. DocXChain: A Powerful Open-Source Toolchain for Document Parsing and Beyond. Preprint at https://arxiv.org/abs/2310.12430 (2023).
60. Wei, H. et al. General OCR Theory: Towards OCR-2.0 via a Unified End-to-end Model. Preprint at https://arxiv.org/abs/2409.01704 (2024).
61. Liu, C. et al. Focus Anywhere for Fine-grained Multi-page Document Understanding. Preprint at https://arxiv.org/abs/2405.14295 (2024).
62. Li, G. et al. dots.ocr: Multilingual Document Layout Parsing in a Single Vision-Language Model. Preprint at https://arxiv.org/abs/2512.02452 (2025).
63. Feng, H. et al. Dolphin: Document Image Parsing via Heterogeneous Anchor Prompting. Preprint at https://arxiv.org/abs/2505.14059 (2025).
64. Efficient Document Parsing via Parallel Token Prediction. Preprint at https://arxiv.org/abs/2603.15206 (2026).
65. Prasad, D., Gadpal, A., Kapadni, K., Visave, M. & Sultanpure, K. CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents. In *Proc. CVPR Workshops* (2020); https://arxiv.org/abs/2004.12629.
66. Chai, M. et al. DocFusion: A Unified Framework for Document Parsing Tasks. Preprint at https://arxiv.org/abs/2412.12505 (2024).
67. 弘犀智能（Great Unicorn Intelligence）. 快消方案（FMCG Solutions）. https://www.hxallai.com/en/solutions/fmcg/ (2026).

68. 弘犀智能（Great Unicorn Intelligence）. 制药方案（Pharma Solutions）. https://www.hxallai.com/en/solutions/pharma/ (2026).
69. Li, M. et al. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. In *Proc. AAAI Conference on Artificial Intelligence* **37**, 13094–13102 (2023); https://arxiv.org/abs/2109.10282.
70. Harley, A. W., Ufkes, A. & Derpanis, K. G. Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval. In *Proc. International Conference on Document Analysis and Recognition* 991–995 (2015); https://arxiv.org/abs/1502.07058.
71. Angelopoulos, A. N. & Bates, S. A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification. Preprint at https://arxiv.org/abs/2107.07511 (2021).
72. 弘犀智能（Great Unicorn Intelligence）. VDC 变量衍生平台（VDC Variable Derivation Platform）. https://www.hxallai.com (2026).
73. 弘犀智能（Great Unicorn Intelligence）. 企业智能体平台 · 场景智能体（Enterprise Agent Platform · Scenario Agents）. https://www.hxallai.com/en/platform/scenario-agents/ (2026).
74. 弘犀智能（Great Unicorn Intelligence）. Deep Hope S1（推训一体平台）. https://www.hxallai.com/en/platform/deep-hope/ (2026).

Copyright Notice: © 2026 Great Unicorn Intelligence Technology (Shenzhen) Co., Ltd. All rights reserved. Any reposting, quotation in commentary, or adaptation must credit the authors (Great Unicorn Intelligence Research Team) and cite the source (Great Unicorn Intelligence, https://www.hxallai.com).