Unlimited OCR Works

Welcome the Era of One-shot Long-horizon Parsing

Baidu Inc.

Abstract

Recently, end-to-end OCR models, exemplified by DeepSeek OCR, have once again thrust OCR into the spotlight. A widely held view is that employing a large language model (LLM) as the decoder allows the model to leverage the prior distribution of language, leading to improved OCR performance. However, the downside is equally evident: as the output sequence lengthens, the accumulated KV cache drives up memory consumption and progressively slows down generation. This stands in stark contrast to humans, who exhibit no such decline in efficiency during long-horizon copying tasks. In this technical report, we propose Unlimited OCR, a model designed to emulate human parsing working memory. Taking DeepSeek OCR as the baseline, we replace all attention layers in the decoder with our proposed Reference Sliding Window Attention (R-SWA), which reduces attention computation costs while maintaining a constant KV cache throughout the entire decoding process. By combining the high compression rate of DeepSeek OCR's encoder with our constant KV cache design, Unlimited OCR can transcribe dozens of pages of documents in a single forward pass under a standard maximum length of 32K. More importantly, R-SWA is a general-purpose parsing attention mechanism — beyond OCR, it is equally applicable to tasks such as ASR, translation, etc. Codes and model weights are publicly available at http://github.com/baidu/Unlimited-OCR.

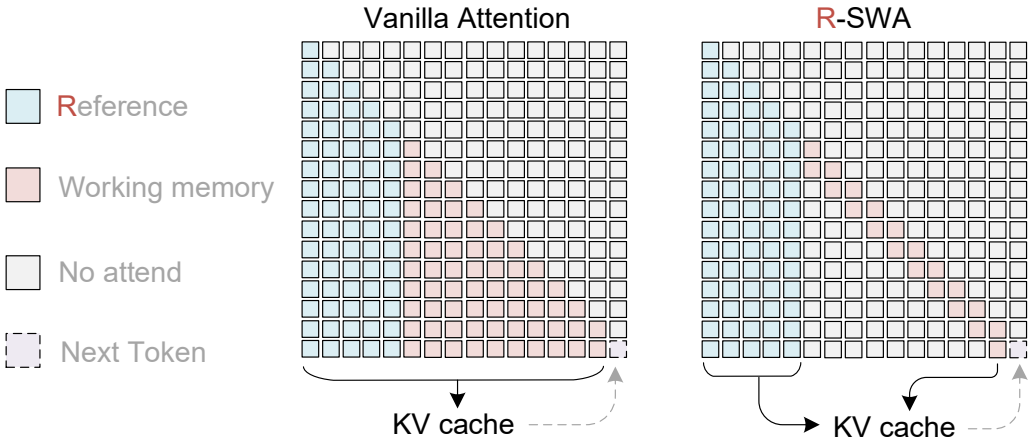

Figure 1 | Illustration of Reference Sliding Window Attention (R-SWA). Each generated token attends to all reference tokens (visual tokens in OCR) and the preceding n output tokens (128 by default). Compared to standard full attention, R-SWA maintains a constant KV cache throughout decoding. Compared to vanilla SWA, it preserves visual token fidelity by excluding them from state transitions, thereby avoiding progressive blurring.

Contents

1. Introduction

Humans are remarkably adept at seemingly straightforward long-horizon tasks: transcribing hundreds of book pages, translating hours-long audio recordings, and the like. Yet these are precisely the tasks where current models fall short. Take OCR as an example—no existing model [10, 30, 33, 34] can even parse ten of pages in a single forward pass. Instead, they resort to page-by-page processing in a for-loop fashion, resetting memory at every step. This divergence is far from superficial, and it cannot be reduced to a mere lack of sufficient context. When humans perform such tasks, they maintain a continuous cognitive state in which distant outputs fade softly from memory, while nearby context is used to track progress. The for-loop paradigm, by contrast, erases memory entirely at each page, fragmenting a coherent long-horizon process into isolated short tasks managed by an external scheduler. It works to some extent, but it remains an engineering workaround, not a step toward AGI-purpose intelligence.

Consider the act of transcribing a document. As we copy each character, we do not scan the entire text already written; we simply glance at the immediately surrounding context to stay oriented. This everyday behavior points to an attention pattern fundamentally different from those in current models. It is not standard full attention—the full history is never fully consulted. Nor does it resemble linear attention, since visual/reference tokens undergo no recurrent state updates; such updates would progressively blur the visual features and degrade recognition accuracy. To align more closely with this natural attention flow, and to explore how multimodal large language models (MLLMs) [8, 14, 22, 28] can handle simple long-horizon parsing tasks, we propose Unlimited OCR. Our main contributions are as follows:

- We introduce Reference Sliding Window Attention (R-SWA), illustrated in Figure 1. For each token, R-SWA attends to all reference tokens—visual tokens and the prompt—while limiting output attention to the preceding n tokens (n defaults to 128). In this way, each token perceives the full image and autonomously tracks OCR progress through state transitions within the causal sliding window. This design keeps the KV cache constant during inference, alleviating memory pressure and reducing the computational cost.
- Building on R-SWA, we propose Unlimited OCR. Using DeepSeek OCR as our baseline, we retain its DeepEncoder with high image compression rate, modifying all the decoder LLM's attention mechanism to R-SWA. This enables Unlimited OCR to parse dozens of paper pages in a single forward pass. R-SWA also yields a modest improvement in general OCR accuracy. Specifically, Unlimited OCR achieves 93% on the OmniDocBench v1.5 benchmark [23], outperforming the DeepSeek OCR baseline by 6%.
- We conduct a preliminary validation of MLLM architectures with linear-complexity attention on OCR tasks, particularly in long-horizon scenarios. Rather than brute-force scaling up the training context, we identify an elegant approach that achieves long-horizon OCR. Looking ahead, we see promise in extending R-SWA to ASR, translation, and other reference-based tasks that demand long-horizon dependency modeling.

In summary, we present R-SWA, which substantially reduces both the computational cost of attention and the memory footprint in the long-horizon inference. Building on R-SWA, Unlimited OCR not only enables one-shot parsing of an entire book, but also surpasses the DeepSeek OCR baseline by a large margin on popular document parsing benchmarks. Furthermore, we believe R-SWA holds promise well beyond OCR.

2. Related Works

2.1. Pipeline-based Framework

Traditional OCR models, particularly those designed for document parsing, typically adopt a pipeline architecture [10, 11, 13, 17, 30]: a detection model first identifies different types of document elements, followed by multiple recognition operators that further parse the content within those blocks. These components are often bridged by a variety of heuristic strategies, such as cropping, rectification, and so on. In recent years, with the powerful decoder capabilities of large language models (LLMs), the pipeline-based OCR paradigm has continued to evolve [17]. The most straightforward adaptation retains the detection model while consolidating the multiple recognition models into a single unified model—a pragmatic hybrid that combines mature traditional detection algorithms with the advanced decoder of an LLM. Beyond this, there is another pipeline variant that invokes the LLM twice, replacing even the detection model with the same LLM [13], so that the entire OCR workflow becomes: LLM detection–cropping strategy–LLM recognition. Thanks to the inherent flexibility in how OCR tasks can be decomposed, pipeline architectures still remain widely adopted to this day.

2.2. End-to-end Model

With the advancement of vision-language models (VLMs) [6, 8, 14, 16, 32] , end-to-end OCR, especially dense OCR models [9, 24, 26, 33–35] are on the rise. This approach fully leverages the powerful decoder capabilities of LLMs by merging text detection and recognition into a single unified function, allowing the entire content of a page to be parsed in a single forward pass. Compared with the pipeline approach, the end-to-end algorithm places higher demands on model capacity and poses greater training challenges. This, in turn, makes research on end-to-end OCR models all the more compelling: innovations in architectural design and iterative improvements in training methodologies can more directly inspire, or even advance, the development of general-purpose VLMs.

2.2.1. High-compression Encoder

In end-to-end models, the encoder is an indispensable module that extracts and compresses image information. To a certain extent, the encoder determines the upper bound of the model: taking generation efficiency as an example, if the input vision tokens are too long—meaning the encoder's token compression ratio is insufficient—the model's decoding efficiency will be hindered by excessively long prefix tokens, thereby affecting decoding speed. The same holds true for effective decoding length. DeepEncoder [34] achieves a 16× token compression rate under low activation values by cascading window attention ViT [15] and global attention one [25], making it an ideal choice for multi-page long-horizon OCR.

2.2.2. High-efficiency Decoder

What most directly affects inference cost is the decoder, including the activation value of the LLM and the KV cache size. Regarding the former, current end-to-end OCR models are typically under 3B parameters. In a related vein, DeepSeek OCR [34] uses an MoE architecture [18], keeping its activation at only 500M during inference. As for the KV cache, current models all see it grow continuously with decoding contexts, which limits both generation speed and length. This is exactly the key issue that our Unlimited OCR aims to address.

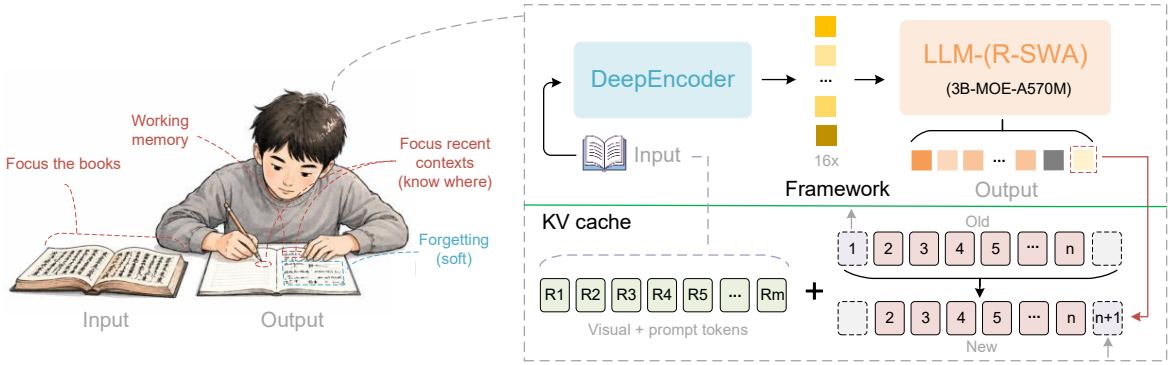

Figure 2 | Inspired by the process of humans copying books, we propose the Unlimited OCR. This model features a unified end-to-end architecture, consisting of an encoder and a MoE-LLM decoder in which all attention mechanisms are R-SWA. The KV cache is implemented as a queue with a capacity of $m + n$—each time a new token is generated, the KV corresponding to the $(m + 1)$-th token in the queue is evicted, ensuring that both computational cost and memory usage do not progressively increase during the generation process.

3. Methodology

3.1. Long-horizon Parsing

Our humans excel at long-horizon parsing tasks—continuously transcribing an entire book, translating even hundreds of pages in one sitting, or transcribing hours of audio without interruption. This continuous parsing capability appears closely linked to the working memory. As illustrated in Figure 2, when a person copies a book by hand, their attention typically centers on three points: the original source book, a small portion of what has just been written (usually only a few characters), and the next character about to be written. Rather than retaining a complete memory of everything already transcribed, they engage in a form of soft forgetting. This maybe the key to sustaining long-horizon parsing under low cognitive load. Inspired by this observation, we present Unlimited OCR.

3.2. Architecture

As shown in Figure 2, Unlimited OCR adopts DeepSeek OCR as its baseline. Specifically, it comprises the DeepEncoder paired with a Mixture-of-Experts (MoE) architecture that enjoys 3B total and 500M activated parameters. The DeepEncoder stands out for its exceptional visual token compression capability, which can dramatically reduce the KV cache footprint during the prefill stage while preserving robust optical text feature extraction. Departing from the original DeepSeek OCR, we replace the vanilla Multi-Head Attention (MHA) with our proposed R-SWA. With the new proposed attention, long-horizon parsing can be achieved by augmenting the original reference KV cache m with a fixed-capacity output KV buffer of width n. We will delve into the technical details in the following sections.

3.3. DeepEncoder

DeepEncoder is originally introduced in DeepSeek OCR [34]. It cascades SAM-ViT [15] with CLIP-ViT [25] and applies 16× [32] token compression at the bridge, so that the first half relies entirely on window attention to process the original image tokens, while global attention

is reserved exclusively for the compressed tokens. This design keeps the activation values low when encoding high-resolution images, thereby conserving GPU memory. DeepEncoder natively supports five resolution modes; we retain two of them: the "Base" model (1024×1024 for multi-page), and the "Gundam" mode (dynamic resolution for single-page). Specifically, DeepEncoder can compress a 1024×1024 PDF-image to just 256 tokens. This high compression ratio is critically important for unlimited OCR works, because visual tokens do not undergo state transitions alongside the output - they are encoded once and remain static throughout the entire long-horizon parsing process.

3.4. Reference Sliding Window Attention

Despite the satisfactory compression of visual tokens that DeepEncoder achieves on the input side, the real bottleneck for one-shot parsing of an entire book lies in the decoding stage. Assume a compression ratio of 1:10 between visual and text tokens — *i.e.*, one visual token can decode around ten text tokens. In that case, 10K visual tokens (equivalent to roughly 20 – 30 pages at 1024×1024 resolution) demand an output length of 100k+ tokens for full decoding. This has long been a formidable challenge for vanilla LLM-driven OCR models, due to the massive KV cache storage and attention computation that sequences beyond 128k tokens entail. To address this, we propose Reference Sliding Window Attention (R-SWA).

3.4.1. Attention computation

In essence, R-SWA constrains attention within a two-segment window of size $m + n$, as illustrated in Figure 2. Here, m denotes the window for prefix tokens, which includes both visual tokens and the prompt. During a single inference pass, m remains fixed; it depends only on the number of book pages or the resolution size of the document being decoded, and does not vary with decoding length. The window n for the decode region is also fixed in size and slides in a causal manner. Specifically, the formulation is as follows:

$$\mathcal{N}(t) = \mathcal{P} \cup \mathcal{D}_n(t); \quad \mathcal{P} = \{1, \ldots, L_m\}, \tag{1}$$

$$\mathcal{D}_n(t) = \{\, j \mid \max(L_m + 1,\ L_m + t - n) \le j \le L_m + t - 1 \}\,, \tag{2}$$

where $\mathcal{P}$ denotes the prefix segment of length L_m, which is globally visible to all subsequent tokens, and $\mathcal{D}_n(t)$ denotes the causal sliding window of width n over the decode region. The attention weight from token t to position $j \in \mathcal{N}(t)$ is then computed as

$$\alpha_{tj} = \frac{\exp\left(\frac{\mathbf{q}_t^\top \mathbf{k}_j}{\sqrt{d_k}}\right)}{\sum_{i \in \mathcal{N}(t)} \exp\left(\frac{\mathbf{q}_t^\top \mathbf{k}_i}{\sqrt{d_k}}\right)}, \quad j \in \mathcal{N}(t), \tag{3}$$

where $\mathbf{q}_t$, $\mathbf{k}_j$, and $\mathbf{v}_j$ are the query, key, and value vectors, respectively, and d_k is the dimension of the key-vector. The output representation is obtained by aggregating values over the same accessible set:

$$\mathbf{o}_t = \sum_{j \in \mathcal{N}(t)} \alpha_{tj} \mathbf{v}_j. \tag{4}$$

This formulation makes explicit that each decoding token can attend to all prefix tokens as persistent global context, while only attending locally within a bounded causal window over previously generated tokens. As a result, the model preserves access to the full prefix information while reducing the attention cost over the growing decode sequence.

3.4.2. *KV cache management*

For DeepSeek OCR baseline, it employs standard Multi-Head Attention (MHA)—the most classical form of attention, which offers strong expressiveness but imposes enormous KV cache pressure, the KV cache size is calculated as follows:

$$C_{\text{MHA}}(T) = L_m + T. \tag{5}$$

In contrast, under R-SWA, the model always retains the full prefix cache of size L_m, but for the generated continuation it only needs to keep the most recent n tokens. Therefore, after generating a total of T tokens, the required KV cache size is

$$C_{\text{R-SWA}}(T) = L_m + \min(n, T) \le L_m + n. \tag{6}$$

This shows that, unlike standard MHA whose cache size increases unboundedly with T, the decode-side cache of R-SWA is upper-bounded by a constant window size. To quantify the reduction, we define the cache ratio

$$\rho(T) = \frac{C_{\text{R-SWA}}(T)}{C_{\text{MHA}}(T)} = \frac{L_m + \min(n, T)}{L_m + T}. \tag{7}$$

If the generated length is sufficiently long such that $T \gg n$, then

$$\rho(T) = \frac{L_m + n}{L_m + T}. \tag{8}$$

which decreases as T grows. In particular, when the decode length dominates both the prefix length and the window size, we have

$$\rho(T) \approx \frac{L_m + n}{T} \to 0. \tag{9}$$

Therefore, for long-sequence decoding, R-SWA reduces the KV cache requirement from linear growth in T to a bounded quantity $L_m + n$, yielding a substantial memory saving compared with standard MHA. Accordingly, R-SWA serves as the cornerstone to enabling near-unlimited parsing works under limited resources.

3.4.3. *Kernel study*

As shown in Figure 3, we plot the per-call duration of the Flash Attention v3 kernel for both the DeepSeek OCR baseline and Unlimited OCR Works (denoted as UOW in the figure). The figure clearly shows that the standard MHA kernel in DeepSeek OCR incurs growing latency with each successive decoding step, whereas in Unlimited OCR the duration remains constant—a direct benefit of adopting R-SWA across all layers of the LLM decoder. The spike in the DeepSeek OCR occurs when the KV cache length crosses a certain alignment boundary, causing an abrupt drop in data transfer efficiency; this issue also does not arise with R-SWA. Besides, the same pattern will hold for GPU memory usage during inference: in the original DeepSeek OCR it scales linearly, while in Unlimited OCR it stays fixed. This joint stability in both computational cost and memory footprint is precisely what makes long-horizon parsing possible.

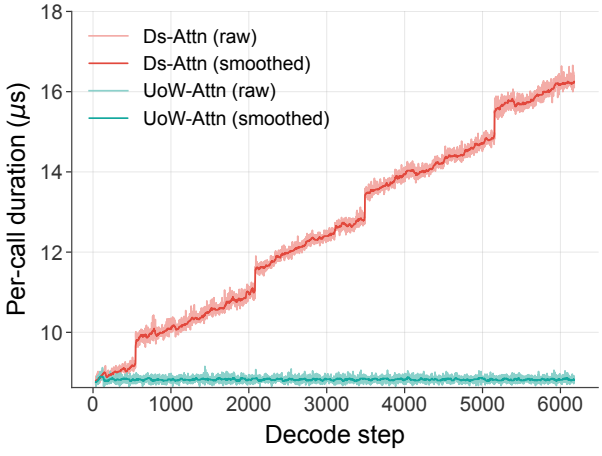

Figure 3 | The latency of the Flash Attention v3 kernel as decoding length increases.

4. Experimental Settings

4.1. Data Engine

We construct approximately 2 million document OCR data samples to train Unlimited OCR, with a 9:1 ratio of single-page to multi-page data. For the single-page PDF data, we use Paddle OCR [11] for annotation, concatenating the coordinates and content of each block to construct end-to-end detection and parsing ground truth. The coordinates of each element are normalized to the range of 0–1000. All multi-page data are synthesized by concatenating single-page data. We randomly generate around 200k samples, each consisting of 2 to 50 pages, with <*page*> used as a separator between pages. All data are packed into a sequence length of 32K tokens.

4.2. Implementation Details

Starting from the DeepSeek OCR checkpoint [34], we continue training Unlimited OCR for 4,000 steps with a global batch size of 256 and a maximum sequence length of 32K on 8×16 A800 GPUs, using random packing for all data. During training, we freeze the DeepEncoder and only train the LLM parameters, as the DeepEncoder is already sufficiently optimized in DeepSeek OCR. We use the AdamW [21] optimizer and a cosine annealing scheduler [20] with an initial learning rate of 1e-4. To support 32K training, we adopt DeepEP [18], with expert parallelism (EP) set to 4. The entire training pipeline is built on the Megatron-LM [27] framework. For inference, we implement KV cache management for R-SWA in the Transformers library, along with corresponding support and optimizations in the SGLang inference engine. Both inference frameworks can operate Unlimited OCR under constant TPS (tokens/S) and GPU memory.

5. Evaluation

5.1. Benchmark and Metrics

We select OmniDocBench [23] as the main benchmark for evaluating foundational document OCR capabilities, and test the Unlimited OCR on both v1.5 and v1.6 versions. OmniDocBench v1.6 includes 296 more test images than v1.5 and represents the latest benchmark, while v1.5 provides official metrics from more classic models—including our baseline DeepSeek OCR—which facilitates performance comparisons. For long-horizon OCR evaluation, an in-house test set is constructed, where we select a number of novels, documents, and papers and divide them by page count to assess the multi-page performance of Unlimited OCR. Specifically, we select books of 2, 5, 10, 20, and 40+ pages for testing, with no fewer than ten books for each category.

OmniDocBench is designed to evaluate document parsing capabilities across multiple dimensions, including text recognition, formula recognition, table structure extraction, and reading order prediction. It adopts task-specific metrics for a well-rounded evaluation: (1) Text Edit Distance (Edit ↓), which measures character-level accuracy for text recognition; (2) Formula CDM (CDM ↑), which evaluates the quality of mathematical formula recognition; (3) Table TEDS (TEDS ↑) and Table TEDS-S (TEDS-S ↑), which assess table structure extraction accuracy with and without content recognition; and (4) Reading Order Edit Distance (Edit ↓), which quantifies the correctness of predicted reading sequences. The overall score is then computed as a weighted average across text, formula, and table recognition tasks. For the in-house benchmark, we report both the Distinct-n and the Edit Distance. Distinct-n is the ratio of the number of unique n-grams to the total number of n-grams in the generated text.

Table 1 | Comparison on OmniDocBench (v1.5/v1.6). All models in the table are end-to-end VLM-based architectures. v1.5 is primarily intended for comparison with classic end-to-end algorithms and the baseline DeepSeek OCR. v1.6 mainly compares against current end-to-end SOTA models. Except for the proposed Unlimited OCR, all other models are selected from the OmniDocBench repository.

Model	Size	Overall ↑	Text^{Edit} ↓	Formula^{CDM} ↑	Table^{TEDs} ↑	Table^{TEDS_s} ↑	Read-order^{Edit} ↓
End-to-end Model (v1.5)							
OCRFlux [3]	3B	74.82	0.193	68.03	75.75	80.23	0.202
GPT-4o [22]	-	75.02	0.217	79.70	67.07	76.09	0.148
InternVL3 [37]	78B	80.33	0.131	83.42	70.64	77.74	0.113
POINTS-Reader [19]	3B	80.98	0.134	79.20	77.13	81.66	0.145
olmOCR [24]	7B	81.79	0.096	86.04	68.92	74.77	0.121
InternVL3.5 [31]	241B	82.67	0.142	87.23	75.00	81.28	0.125
MinerU2-VLM [30]	0.9B	85.56	0.078	80.95	83.54	87.66	0.086
Nanonets-OCR-s [1]	3B	85.59	0.093	85.90	80.14	85.57	0.108
Qwen2.5-VL [8]	72B	87.02	0.094	88.27	82.15	86.22	0.102
Gemini-2.5 Pro[4]	-	88.03	0.075	85.82	85.71	90.29	0.097
dots.ocr [26]	3B	88.41	0.048	83.22	86.78	90.62	0.053
OCRVerse [2]	4B	88.56	0.058	86.91	84.55	88.45	0.071
Qwen3-VL[7]	235B	89.15	0.069	88.14	86.21	90.55	0.068
DeepSeek-OCR 2 [35]	3B-A0.5B	89.17	0.049	86.85	85.60	90.06	0.060
DeepSeek-OCR	3B-A0.5B	87.01	0.073	83.37	84.97	88.80	0.086
Unlimited-OCR	3B-A0.5B	93.23	0.038	92.61	90.93	94.07	0.045
		↑ 6.22	↓ 0.035	↑ 9.24	↑ 5.96	↑ 5.27	↓ 0.041
End-to-end Model (v1.6)							
HunyuanOCR [29]	1B	89.95	0.088	87.68	91.01	92.23	0.171
DeepSeek-OCR 2 [35]	3B-A0.5B	90.25	0.050	91.84	83.89	87.75	0.144
dots.ocr [26]	3B	90.77	0.048	89.95	87.18	90.58	0.138
FireRed-OCR [36]	2B	93.26	0.037	95.44	88.04	91.06	0.131
Logics-Parsing-v2 [5]	4B	93.33	0.041	95.65	88.42	91.98	0.137
Qianfan-OCR [12]	4B	93.90	0.040	95.08	90.53	93.31	0.13
Unlimited-OCR	3B-A0.5B	93.92	0.042	95.79	90.16	93.32	0.129

5.2. Main Results

As shown in Table 1, by continue-training on merely 2M PDF-document-specific data based on DeepSeek OCR, Unlimited OCR achieves end-to-end SOTA performance. This demonstrates the effectiveness of R-SWA on parsing tasks. First, compared with the standard attention in DeepSeek OCR, R-SWA may allow the model to focus more on dense OCR tasks, whereas full attention could lead to divergence as the output length increases. On the other hand, the state transition across intra-page content under R-SWA is both workable and solid. Specifically, on OmniDocBench v1.5, compared with DeepSeek OCR, the text edit distance drops by 0.035, and the table TEDS improves by 5.96%, indicating that historical information is causally and continuously fed into the sliding window, enabling the model to clearly locate its OCR progress even though it sees only a few tokens. On the OmniDocBench v1.6 benchmark, Unlimited OCR again achieves end-to-end SOTA (93.92% on overall metric), further proving that for single-page PDF-level document OCR tasks, replacing all standard attention entirely with R-SWA of width 128 is both effective and lossless.

Moreover, Unlimited OCR gains all the benefits of DeepSeek OCR, such as the MoE architecture with only 0.5B activated parameters, resulting in very high inference efficiency. In

the OmniDocBench, Unlimited OCR achieves 5580 TPS (tokens/s/512 concurrency) compared to DeepSeek OCR's 4951 TPS under "Base" DeepEncoder mode, representing a 12.7% speed increase. Of course, the average document length in OmniDocBench is relatively short—the longer the output length, the more pronounced the advantage of Unlimited OCR becomes.

5.3. Subcategory Study

OmniDocBench (v1.5) provides 9 types of documents, and conducting a subcategory comparison is crucial for a more systematic and comprehensive analysis of R-SWA. As shown in Table 2, compared to DeepSeek OCR, Unlimited OCR shows clear and consistent gains across every metric, demonstrating that our decoder-side optimization, *i.e.*, R-SWA, delivers a genuine "free lunch"—improvements without compromises. Compared to DeepSeek OCR 2, Unlimited OCR also holds a clear advantage, with seven-ninths of both the text edit distance and reading order scores surpassing those of DeepSeek OCR 2. For documents with complex layouts such as PPT, newspapers, magazines, and note, Unlimited OCR shows no disadvantage either, further demonstrating that replacing all standard attention with R-SWA for LLM-decoder is complete and sound for parsing tasks.

Table 2 | Detailed subcategory comparison between Unlimited OCR and the DeepSeek-OCR series across nine document types. R-order denotes reading order. All metrics are edit distances, where lower is better. Red cells indicate that the corresponding metric of DeepSeek-OCR or DeepSeek-OCR 2 is better than that of Unlimited OCR.

Model	Edit ↓	PPT	Academic Paper	Book	Colorful Textbook	Exam Paper	Magazine	Newspaper	Note	Research Report
DS-OCR	Text	0.052	0.028	0.022	0.130	0.074	0.049	0.131	0.145	0.015
	R-order	0.052	0.021	0.040	0.125	0.083	0.101	0.217	0.089	0.016
DS-OCR 2	Text	0.031	0.013	0.033	0.053	0.047	0.026	0.139	0.068	0.008
	R-order	0.025	0.013	0.027	0.066	0.048	0.100	0.176	0.035	0.011
UOW	Text	0.025	0.023	0.019	0.046	0.049	0.020	0.081	0.066	0.008
	R-order	0.023	0.012	0.025	0.051	0.049	0.061	0.134	0.018	0.013

5.4. Long-horizon Parsing

Long-horizon parsing is one of the novel capabilities of Unlimited OCR. Two main obstacles have hindered previous models from achieving this: first, excessively long output sequences can easily exceed the maximum token limit; second, output latency grows with sequence length, causing the OCR of documents spanning dozens of pages to become progressively slower. Unlimited OCR, equipped with R-SWA, can prefill tens to hundreds of document pages in a single pass and parse continuously from the first page to the last. Throughout this process, the KV cache remains fixed, so output latency stays constant—making long-horizon parsing feasible. As shown in Table 3, our model delivers satisfactory performance in multi-page one-shot OCR scenarios, maintaining strong results even with 20 pages input simultaneously. At 40+ pages, the edit distance remains below 0.11 along with 97% Distinct-35. We examine the cases with repeated errors and find that most occur where small text in the PDF is difficult to discern, primarily due to the use of DeepEncoder's "Base" mode (1024×1024 resolution) under multi-page conditions, rather than R-SWA losing direction in long-horizon parsing process.

Table 3 | Performance of long-horizon OCR. We test the distinct-n and edit distance under different page numbers. Distinct-n is the higher the better.

Metric \ Pages	2	5	10	15	20	40+
Distinct-20 ↑	99.76%	99.78%	97.49%	99.92%	98.73%	96.08%
Distinct-35 ↑	99.87%	99.98%	99.83%	99.99%	99.89%	96.90%
Edit Distance ↓	0.0362	0.0452	0.0526	0.0787	0.0572	0.1069

Table 4 | Theoretical inference performance ceiling comparison. We compare the TPS upper limits of DeepSeek OCR and Unlimited OCR across varying output lengths.

Model \ TPS	256	512	1024	2048	3072	4096	6144
Deepseek OCR	7229.32	7468.27	7422.50	7166.85	6790.72	6430.21	5822.87
Unlimited OCR	7229.52	7714.78	7840.94	7881.11	7881.93	7905.18	7847.71

6. Efficiency Analysis

As presented in Table 4, we compare the output tokens per second (TPS) of Unlimited OCR and DeepSeek OCR under ideal concurrency conditions. The prefill length is fixed at 10, with all other settings held identical. The results show that at 256 tokens, the inference speeds of the two models are virtually the same. As the output length grows, however, the TPS of DeepSeek OCR steadily declines, and at 6,000 tokens, it lags behind Unlimited OCR—which incorporates R-SWA—by 35%. These findings further validate the effectiveness of R-SWA and underscore that consistent generation speed is a critical requirement for long-horizon OCR tasks.

7. Limitation and Future Work

Our model cannot achieve truly unlimited parsing under a finite context length (*e.g.*, 32K), as it is also constrained by the prefill length. Although DeepEncoder already achieves a high compression rate for image tokens, the prefill still becomes very long as the number of pages accumulates. In the short term, we will train models with longer context lengths, such as 128K, to support the prefill of more pages. In the long term, we plan to build a prefill pool and enable the model to learn to automatically fetch prefill KV chunks, thereby simulating the effect of a human flipping through pages, so as to achieve truly unlimited OCR works. In addition, we will also transfer R-SWA to reference-based tasks such as ASR and translation.

8. Conclusion

In this technical report, we propose the Unlimited OCR model and present the R-SWA algorithm to support its capability for long-horizon parsing. We verify that when all standard attention in the decoder of an end-to-end model is replaced with causal reference-based SWA, the model's performance on parsing tasks remains lossless. This indicates that the model learns to continuously pass useful information from historical outputs into the window, and this soft form of forgetting is consistent with how we humans behave when transcribing a book. We believe that R-SWA will be applied to more tasks in the future, making attention computation and memory footprint no longer the bottleneck for long-horizon parsing field.

9. Author List

* indicates project leader; † indicates technical director

Core Contributors: Youyang Yin, Huanhuan Liu*, YY†

Contributors: Qunyi Xie, Chaorun Liu, Shiqi Yang, Shaohua Wang, Zhanlong Liu, Hao Zou, Jinyue Chen, Shu Wei, Jingjing Wu, Mingxin Huang, Zhen Wu, Guibin Wang, Tengyu Du, Lei Jia

References

[1] Nanonets-ocr-s, 2025. URL https://huggingface.co/nanonets/Nanonets-OCR-s.

[2] Ocrverse, 2025. URL https://github.com/DocTron-hub/OCRVerse.

[3] Ocrflux, 2025. URL https://github.com/chatdoc-com/OCRFlux.

[4] G. AI. Gemini 2.5-pro, 2025. URL https://gemini.google.com/.

[5] alibaba, 2026. URL https://github.com/alibaba/Logics-Parsing.

[6] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966, 2023.

[7] S. Bai, Y. Cai, R. Chen, et al. Qwen3-vl technical report. arXiv preprint arXiv:2511.21631, 2025. URL https://arxiv.org/abs/2511.21631.

[8] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin. Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923, 2025.

[9] L. Blecher, G. Cucurull, T. Scialom, and R. Stojnic. Nougat: Neural optical understanding for academic documents. arXiv preprint arXiv:2308.13418, 2023.

[10] C. Cui, T. Sun, S. Liang, et al. Paddleocr-vl: Boosting multilingual document parsing via a 0.9 b ultra-compact vision-language model. arXiv preprint arXiv:2510.14528, 2025.

[11] C. Cui, T. Sun, M. Lin, T. Gao, Y. Zhang, J. Liu, X. Wang, Z. Zhang, C. Zhou, H. Liu, et al. Paddleocr 3.0 technical report. arXiv preprint arXiv:2507.05595, 2025.

[12] D. Dong, M. Zheng, D. Xu, C. Luo, B. Zhuang, Y. Li, R. He, H. Wang, W. Zhang, W. Wang, et al. Qianfan-ocr: A unified end-to-end model for document intelligence. arXiv preprint arXiv:2603.13398, 2026.

[13] H. Feng, S. Wei, X. Fei, W. Shi, Y. Han, L. Liao, J. Lu, B. Wu, Q. Liu, C. Lin, et al. Dolphin: Document image parsing via heterogeneous anchor prompting. arXiv preprint arXiv:2505.14059, 2025.

[14] A. Huang, C. Yao, C. Han, F. Wan, H. Guo, H. Lv, H. Zhou, J. Wang, J. Zhou, J. Sun, et al. Step3-vl-10b technical report. arXiv preprint arXiv:2601.09668, 2026.

[15] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. arXiv preprint arXiv:2304.02643, 2023.

[16] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In International conference on machine learning, pages 19730–19742. PMLR, 2023.

[17] Z. Li, Y. Liu, Q. Liu, Z. Ma, Z. Zhang, S. Zhang, Z. Guo, J. Zhang, X. Wang, and X. Bai. Monkeyocr: Document parsing with a structure-recognition-relation triplet paradigm. arXiv preprint arXiv:2506.05218, 2025.

[18] A. Liu, A. Mei, B. Lin, B. Xue, B. Wang, B. Xu, B. Wu, B. Zhang, C. Lin, C. Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. arXiv preprint arXiv:2512.02556, 2025.

[19] Y. Liu, Z. Zhao, L. Tian, et al. Points-reader: Distillation-free adaptation of vision-language models for document conversion. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 1576–1601, November 2025.

[20] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983, 2016.

[21] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In ICLR, 2019.

[22] OpenAI. Gpt-4 technical report, 2023.

[23] L. Ouyang, Y. Qu, H. Zhou, J. Zhu, R. Zhang, Q. Lin, B. Wang, Z. Zhao, M. Jiang, X. Zhao, et al. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 24838–24848, 2025.

[24] J. Poznanski, A. Rangapur, J. Borchardt, J. Dunkelberger, R. Huff, D. Lin, C. Wilhelm, K. Lo, and L. Soldaini. olmocr: Unlocking trillions of tokens in pdfs with vision language models. arXiv preprint arXiv:2502.18443, 2025.

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pages 8748–8763. PMLR, 2021.

[26] Rednote. dots.ocr, 2025. URL https://github.com/rednote-hilab/dots.ocr.

[27] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. arXiv preprint arXiv:1909.08053, 2019.

[28] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.

[29] H. V. Team, P. Lyu, X. Wan, G. Li, S. Peng, W. Wang, L. Wu, H. Shen, Y. Zhou, C. Tang, et al. Hunyuanocr technical report. arXiv preprint arXiv:2511.19575, 2025.

[30] B. Wang, C. Xu, X. Zhao, L. Ouyang, F. Wu, Z. Zhao, R. Xu, K. Liu, Y. Qu, F. Shang, et al. Mineru: An open-source solution for precise document content extraction. arXiv preprint arXiv:2409.18839, 2024.

[31] W. Wang, Z. Gao, L. Gu, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265, 2025.

[32] H. Wei, L. Kong, J. Chen, L. Zhao, Z. Ge, J. Yang, J. Sun, C. Han, and X. Zhang. Vary: Scaling up the vision vocabulary for large vision-language model. In European Conference on Computer Vision, pages 408–424. Springer, 2024.

[33] H. Wei, C. Liu, J. Chen, J. Wang, L. Kong, Y. Xu, Z. Ge, L. Zhao, J. Sun, Y. Peng, et al. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. arXiv preprint arXiv:2409.01704, 2024.

[34] H. Wei, Y. Sun, and Y. Li. Deepseek-ocr: Contexts optical compression. arXiv preprint arXiv:2510.18234, 2025.

[35] H. Wei, Y. Sun, and Y. Li. Deepseek-ocr 2: Visual causal flow. arXiv preprint arXiv:2601.20552, 2026.

[36] H. Wu, H. Lou, X. Li, Z. Zhong, Z. Sun, P. Chen, X. Zhou, K. Zuo, Y. Chen, X. Tang, et al. Firered-ocr technical report. arXiv preprint arXiv:2603.01840, 2026.

[37] J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, H. Tian, Y. Duan, W. Su, J. Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479, 2025.