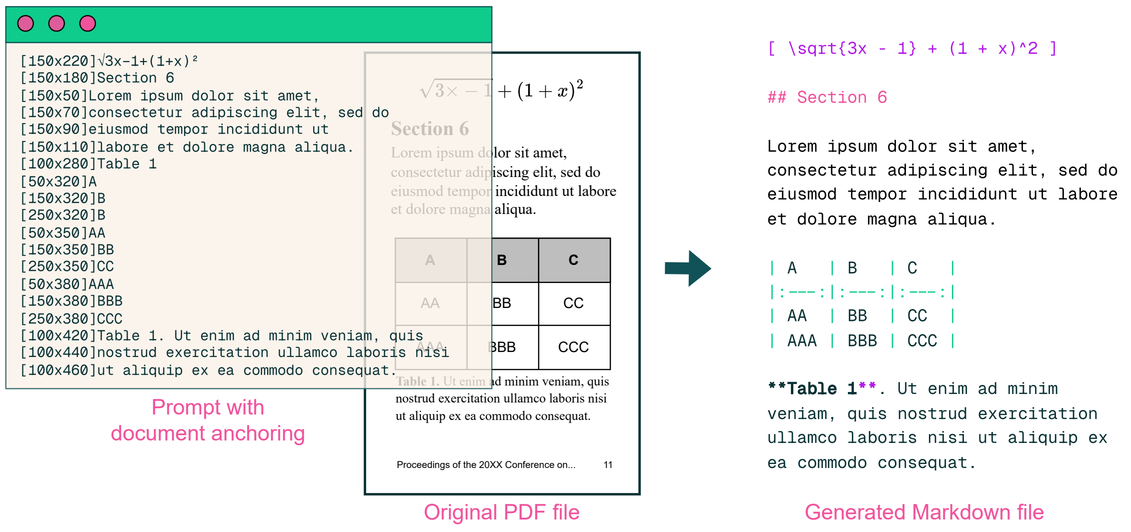
olmOCR: Unlocking Trillions of Tokens in PDFs with Vision Language Models

Jake Poznanski♥

Jason Dunkelberger Regan Huff Daniel Lin Aman Rangapur Christopher Wilhelm

Kyle Lo♥ Luca Soldaini♥

Allen Institute for AI, Seattle, USA `{jakep|kylel|lucas}@allenai.org` ♥ indicates core contributors.



```
[150x220]√3x−1+(1+x)²
[150x180]Section 6
[150x50]Lorem ipsum dolor sit amet,
[150x70]consectetur adipiscing elit, sed do
[150x90]eiusmod tempor incididunt ut
[150x110]labore et dolore magna aliqua.
[100x280]Table 1
[50x320]A
[150x320]B
[250x320]B
[50x350]AA
[150x350]BB
[250x350]CC
[50x380]AAA
[150x380]BBB
[250x380]CCC
[100x420]Table 1. Ut enim ad minim veniam, quis
[100x440]nostrud exercitation ullamco laboris nisi
[100x460]ut aliquip ex ea commodo consequat.
```

$\sqrt{3x - 1} + (1 + x)^2$

Section 6

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

A	B	C
AA	BB	CC
AAA	BBB	CCC

Table 1. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Proceedings of the 20XX Conference on... 11

```
[ \sqrt{3x - 1} + (1 + x)^2 ]

## Section 6

Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore
et dolore magna aliqua.

| A   | B   | C   |
|:---:|:---:|:---:|
| AA  | BB  | CC  |
| AAA | BBB | CCC |

**Table 1**. Ut enim ad minim
veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex
ea commodo consequat.
```

Abstract

PDF documents have the potential to provide trillions of novel, high-quality tokens for training language models. However, these documents come in a diversity of types with differing formats and visual layouts that pose a challenge when attempting to extract and faithfully represent the underlying content for language model use. We present OLMOCR, an open-source Python toolkit for processing PDFs into clean, linearized plain text in natural reading order while preserving structured content like sections, tables, lists, equations, and more. Our toolkit runs a fine-tuned 7B vision language model (VLM) trained on a sample of 260,000 pages from over 100,000 crawled PDFs with diverse properties, including graphics, handwritten text and poor quality scans. OLMOCR is optimized for large-scale batch processing, able to scale flexibly to different hardware setups and convert a million PDF pages for only $190 USD. We release all components of OLMOCR including VLM weights, data and training code, as well as inference code built on serving frameworks including vLLM and SGLang.

Code	`allenai/olmocr`
Weights & Data	`allenai/olmocr`
Demo	`olmocr.allenai.org`

1 Introduction

Access to clean, coherent textual data is a crucial component in the life cycle of modern language models (LMs). During model development, LMs require training on trillions of tokens derived from billions of documents (Soldaini et al., 2024; Penedo et al., 2024; Li et al., 2024); errors from noisy or low fidelity content extraction and representation can result in training instabilities or even worse downstream performance (Penedo et al., 2023; Li et al., 2024; OLMo et al., 2024). During inference, LMs are often prompted with plain text representations of relevant document context to ground user prompts; for example, consider information extraction (Kim et al., 2021) or AI reading assistance (Lo et al., 2024) over a user-provided document and cascading downstream errors due to low quality representation of the source document.

While the internet remains a valuable source of textual content for language models, large amounts of content are not readily available through web pages. Electronic documents (*e.g.*, PDF, PS, DjVu formats) and word processing files (*e.g.*, DOC, ODT, RTF) are widely-used formats to store textual content. However, these formats present a unique challenge: unlike modern web standards, they encode content to facilitate rendering on fixed-size physical pages, at the expense of preserving logical text structure. For example, consider the PDF format, which originated as a means to specify how digital documents should be printed onto physical paper. As seen in Figure 1, PDFs store not units of text—headings, paragraphs, or other meaningful prose elements—but single characters alongside their spacing, placement, and any metadata used for visual rendering on a page. As more and more documents became digital, users have relied this file format to create trillions of documents (PDF Association staff, 2015); yet, these documents remain difficult to leverage in LM pipelines because PDFs lack basic structure necessary for coherent prose, such as ground truth reading order.

```
Character: 'o'
Transform Matrix: (1.02, 0.0, 0, 1, 70.866, 709.481)
Font: JURTWD+Manrope-Bold, Size: 24.78710000000001
------------------------------
Character: 'l'
Transform Matrix: (1.02, 0.0, 0, 1, 86.490796356, 709.481)
Font: JURTWD+Manrope-Bold, Size: 24.78710000000001
------------------------------
Character: 'm'
Transform Matrix: (1.02, 0.0, 0, 1, 93.56999211600001, 709.481)
Font: JURTWD+Manrope-Bold, Size: 24.78710000000001
------------------------------
Character: 'O'
Transform Matrix: (1.02, 0.0, 0, 1, 116.299267074, 709.481)
Font: JURTWD+Manrope-Bold, Size: 24.78710000000001
------------------------------
Character: 'C'
Transform Matrix: (1.02, 0.0, 0, 1, 135.236115732, 709.481)
Font: JURTWD+Manrope-Bold, Size: 24.78710000000001
------------------------------
Character: 'R'
Transform Matrix: (1.02, 0.0, 0, 1, 153.894853128, 709.481)
Font: JURTWD+Manrope-Bold, Size: 24.78710000000001
```

Figure 1 Example of how PDFs represent textual content, such as this paper title, as individual glyphs with metadata.

Faithful content extraction and representation of digitized print documents has long been of interest, with early research efforts in the 1950s, and first commercial optical character recognition (OCR) tools debuting in the late 1970s (Mori et al., 1992). The release of Tesseract in 2006 represented a significant milestone, as the first high-quality, open-source OCR toolkit (Smith, 2013). The current landscape of PDF extraction toolkits can be partitioned in pipeline-based systems and end-to-end models. Pipeline-based systems (MinerU, Wang et al. 2024a; Marker, Paruchuri 2025) are comprised of multiple ML components (*e.g.*, section segmentation, table parsing) chained together; some, such as Grobid (GRO, 2008–2025), VILA (Shen et al., 2022), and PaperMage (Lo et al., 2023), are tailored to scientific papers. On the other hand, end-to-end models parse a document with a single model. For example, Nougat (Blecher et al., 2023) and GOT Theory 2.0 (Wei et al., 2024) take images of PDF pages as input, and return plain text. Notably, while pipeline-based systems have historically focused on simply faithful extraction, end-to-end-systems have also made strides to enable *linearization* of this content—prescribing a flattening of this content to adhere to logical reading order—which can be quite challenging for layout-rich documents with many floating elements (e.g. multi-column documents with floating diagrams, headers, footnotes, and more). Recently, rapid advances in the proprietary LMs have led to significant improvements in end-to-end text extraction capabilities (Bai et al., 2025; Google, 2025). However, this capability comes at a steep price: for example, converting a million pages using GPT-4o can cost over $6,200 USD.[1]

We introduce **olmOCR**, a general-purpose context extraction and linearization toolkit to convert PDFs or images of documents into clean plain text:

- olmOCR is capable of processing a diversity of document types, covering different domains as well as visual layouts. It uses Markdown (Gruber, 2004) to represent structured content, such as sections, lists, equations and tables.
- Unlike other end-to-end models, olmOCR uses *both* text and visual information to obtain an accurate text representation of a documents. We develop document-anchoring, a technique to extract text and layout information from born-digital PDF documents. document-anchoring can be used to prompt VLMs alongside images of document pages to significantly improve extraction.
- To build olmOCR, we curate `olmOCR-mix-0225`, a dataset of nearly 260,000 PDF pages from a diverse set of PDFs crawled from the web and public domain books. This corpus is used to fine-tune `olmOCR-7B-0225-preview` from Qwen2-VL-7B-Instruct (Wang et al., 2024b). We release `olmOCR-mix-0225` to facilitate further research in document extraction, and open source model weights and code as part of our toolkit.
- olmOCR is a fully optimized pipeline compatible with both SGLang (Zheng et al., 2024) and vLLM (Kwon et al., 2023) inference engines. In our tests, we have been able to scale it efficiently from one to hundreds of GPUs. It achieves an amortized cost of less than $190 per million pages converted, or $1/32^{nd}$ of the price of calling GPT-4o APIs. Furthermore, olmOCR is resilient: it includes several heuristics to handle common parsing failures and metadata errors.

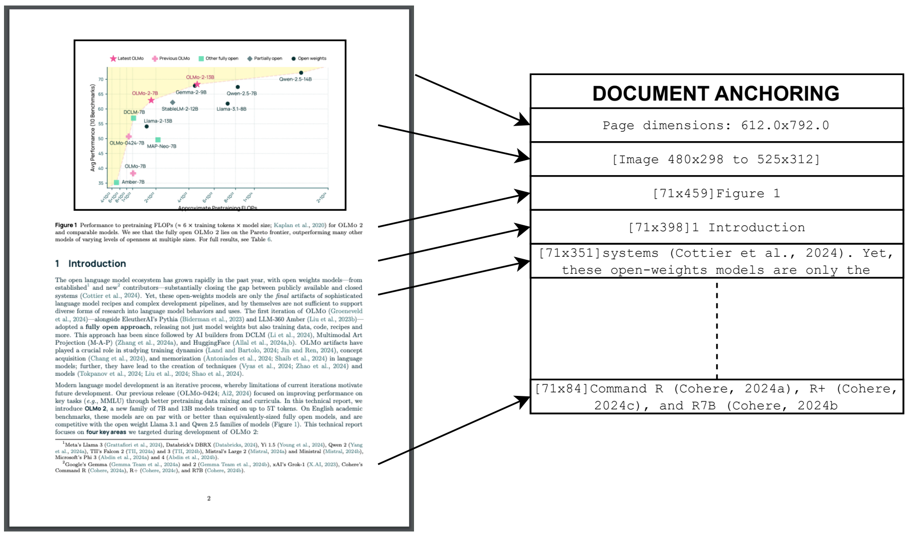
2 Methodology

Approach Many end-to-end OCR models, such as GOT Theory 2.0 (Wei et al., 2024) and Nougat (Blecher et al., 2023), exclusively rely on rasterized pages to convert documents to plain text; that is, they process images of the document pages as input to autoregressively decode text tokens. This approach, while offering great compatibility with image-only digitization pipelines, misses the fact that most PDFs are born-digital documents, thus already contain either digitized text or other metadata that would help in correctly linearizing the content.

In contrast, the olmOCR pipeline leverages document text and metadata. We call this approach **document-anchoring**. Figure 2 provides an overview of our method; document-anchoring extracts coordinates of salient elements in each page (*e.g.*, text blocks and images) and injects them alongside raw text extracted from the PDF binary file. Crucially, the anchored text is provide as input to any VLM *alongside* a rasterized image of the page.

Our approach increases the quality of our content extraction. We apply document-anchoring when prompting GPT-4o to collect silver training samples, when fine-tuning `olmOCR-7B-0225-preview`, and when performing inference with the olmOCR toolkit.

[1]With batch pricing, at $1.25 USD (input) and $5.00 USD (output) per million tokens in Feb 2025.



```
Page dimensions: 612.0x792.0
[Image 480x298 to 525x312]
[71x459]Figure 1
[71x398]1 Introduction
[71x351]systems (Cottier et al., 2024). Yet,
these open-weights models are only the
[71x84]Command R (Cohere, 2024a), R+ (Cohere,
2024c), and R7B (Cohere, 2024b
```

Figure 2 Example of how DOCUMENT-ANCHORING works for a typical page. Relevant image locations and text blocks get extracted, concatenated, and inserted into the model prompt. When prompting a VLM for a plain text version of the document, the anchored text is used in conjunction with the rasterized image of a page.

Implementation DOCUMENT-ANCHORING processes PDF document pages via the `pypdf` (PyPDF, 2012–2025) library to extract a representation of the page's structure from the underlying PDF. All of the text blocks and images in the page are extracted, including position information. Starting with the most relevant text blocks and images[2], these are sampled and added to the prompt of the VLM, up to a defined maximum character limit[3]. This extra information is then available to the model when processing the document.

Overall, we find that using prompts constructed using DOCUMENT-ANCHORING results in significantly fewer hallucinations. Prompting with just the page image was prone to models completing unfinished sentences, or to invent larger texts when the image data was ambiguous. Finally, while DOCUMENT-ANCHORING helps with quality on born-digital documents, our pipeline maintains high performance on documents that do not have any digital metadata encoded in them. In these cases, the model will not have the benefit of seeing the internal structure of the PDF document, instead relying on just the rasterized image of a page to process the underlying document.

3 Fine-tuning Models for olmOCR

While the DOCUMENT-ANCHORING method presented in Section §2 can be used to prompt any language model, we find that fine-tuning a smaller VLM on this linearization task yields models that are as accurate as larger, general-purpose models and are much more efficient at inference time. In this section, we summarize the procedure we followed to distill the output of a larger VLM; Section §3.1 details how we collect the training data, while Section §3.2 provides an overview of fine-tuning.

3.1 Dataset

Choice of teacher model At the time of construction (October 2024), we evaluated several open-weights and API models to label our training data. Our analysis relied on qualitative assessment of linearization performance on a small collection of PDF pages with moderately complicated layout. At this stage, we evaluated model outputs visually across a set of several PDF documents that were known to cause problems for

[2]We prioritize text blocks and images which are located at the start and end of the document.

[3]We use a character limit for convenience and speed, but during training or inference, if a page's prompt exceeds the model's token limit, we just regenerate it with exponentially decreasing character limits until it is suitable.

traditional PDF content extraction tools, and contained varied elements such as equations, tables, multi-column layouts, and born-analog documents captured in poor lighting conditions.

Among possible options, we found GPT-4o, GPT-4o mini, Gemini 1.5, and Claude Sonnet 3.5 to have acceptable performance. Gemini was discarded because a high proportion of prompts returned `RECITATION` errors, meaning that the output was too similar to Gemini's own training data[4]. Ultimately, we chose `gpt-4o-2024-08-06` due to its high performance and relatively low cost in batch mode. GPT-4o mini produced too many hallucinations, and Claude Sonnet 3.5 was found to be cost prohibitive.

Choice of PDF tools olmOCR leverages two tools for PDF rasterization and metadata manipulation: Poppler (2005–2025) transforms pages in a PDF to images; PyPDF (2012–2025) extracts text blocks, images, and their positions as part of document-anchoring.

Prompting strategy We prompt GPT-4o with the image of a PDF page. We render each page using Poppler's `pdftoppm` tool, at a resolution such that its longest edge is 2048 pixels, the largest supported by the GPT-4o model at the time. We report the full prompt in Appendix A, as augmented by our document-anchoring technique. As mentioned in Section §2, we found that this significantly reduced hallucinations. In contrast, prompting with just the page image was prone to complete unfinished sentences, or to produce unfaithful output when the image data was ambiguous.

Finally, we instruct GPT-4o to respond with structured output to our requests. We report the full JSON schema in Appendix A. It forces the model to first extract page metadata, such as language, page orientation, and presence of tables, before moving onto outputting the text of the page in a natural reading order. We believe the order of the fields in the structured schema is helpful in improving output quality, because it forces the model to analyze the page as a whole first before moving on to extracting the text. The output schema encourages the model to return no text at all, if appropriate: this aspect is necessary to prevent GPT-4o from visually describing the content of a PDF in cases where no text is found on the page (*e.g.*, the page contains just a photographic image or non-textual diagram).

Source	Unique docs	Total pages
Web crawled PDFs	99,903	249,332
Internet Archive books	5,601	16,803
Total	*105,504*	*266,135*

Table 1 Training set composition by source. Web crawled PDFs are sampled from a set of over 240 million documents crawled from public websites. Books in the Internet Archive set are in the public domain.

Document type	Fraction
Academic	60%
Brochure	12%
Legal	11%
Table	6%
Diagram	5%
Slideshow	2%
Other	4%

Table 2 Web PDFs breakdown by document type. Distribution is estimating by sampling 707 pages, which are classified using `gpt-4o-2024-11-20`. Details of the prompt used in Appendix A.

Data acquisition and page sampling To generate the primary training dataset, we sample 100,000 PDFs from an internal dataset of 240 million PDFs crawled from public internet sites. Using the Lingua package (Emond, 2025), we identify and filter out documents that were not in English. Further, we remove any document that failed to be parsed by pypdf, contains spam keywords, is a fillable form, or whose text is too short[5]. We then sampled at most 3 pages uniformly at random from each PDF. This resulted in approximately 249,332 PDF pages. We also sampled 5,601 PDFs from a dataset public domain scanned books from the Internet Archive, and processed it similarly. Unlike the web crawled set, PDFs in this collection consist of image scans of book pages, as opposed to born-digital documents. We summarize the data distribution in Tables 1 and 2.

[4]We tried Gemini 1.5 Flash again at time of release (Feb 2025) and the proportion of RECITATION errors was below 0.3%.
[5]An implementation of these heuristics is available on GitHub: `olmocr/filter/filter.py#L14-L112`.

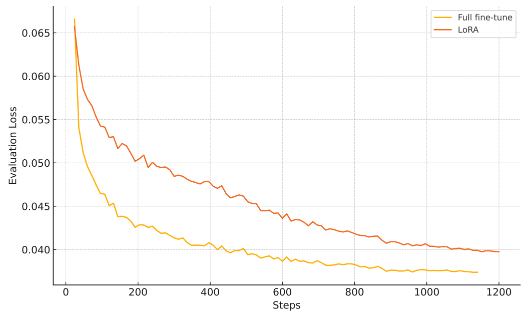


Figure 3 Validation Loss - Web PDFs

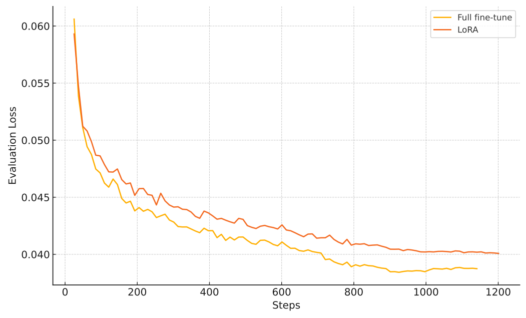


Figure 4 Validation Loss - Internet Archive Books

3.2 Model Training

Fine-tuning `olmOCR-7B-0225-preview` is fine-tuned from a Qwen2-VL-7B-Instruct checkpoint. Training is implemented using Hugging Face's `transformers` library (Wolf et al., 2020). Hyperparameters are set as follows: we use an effective batch size of 4 (batch size 1 with 4 gradient accumulation steps), a learning rate of $1e-6$, AdamW optimizer, and a cosine annealing schedule for 10,000 steps (roughly 1.2 epochs). We use single node with 8 NVIDIA H100 GPUs. We experimented with both full fine-tuning as well as LoRA (Hu et al., 2021).

Dataset formatting During fine-tuning, we slightly alter the prompt used for dataset labeling in Section §3.1. Namely, we remove some of the additional instructions from the prompt, and shrink the image size so that PDF pages get rendered to a maximum dimension of 1024 pixels on the longest edge. The simplified text prompt is listed in Appendix A. The prompt is capped to 6,000 characters, so a typical prompt uses about 1,000 tokens to encode a page image, 1,800 tokens for the anchor text, for about 3,000 total input tokens.

We keep the same structured JSON output that was present in the outputs of `olmOCR-mix-0225`. Each training example was truncated to 8,192 tokens to cover cases when the prompt was unusually large, and the loss was masked so only the final response tokens participated in the loss calculation.

Evaluation We track validation loss against a development subset of `olmOCR-mix-0225` during fine-tuning; Figure 3 and Figure 4, show the loss curves for both the web PDFs and the Internet Archive books subsets. LoRA resulted in higher loss values compared to full fine-tuning, which we use for the final model.

To set hyperparameters and make other decisions during development, we relied on manual side-by-side evaluation as shown in Figure 5. A random selection of 20 to 50 documents were processed using two different methods, and were displayed side by side along with the render of the document page. We also open source our evaluation tool to support qualitative inspection of this visually-rich data.

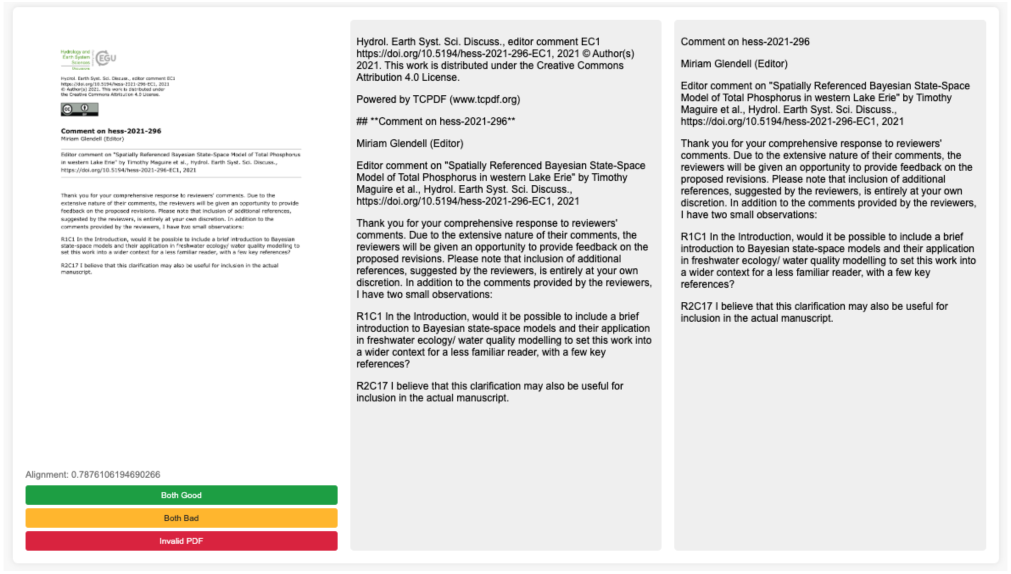
4 Deploying olmOCR

4.1 Inference Pipeline

To efficiently convert millions of documents, we develop the OLMOCR pipeline using SGLang (Zheng et al., 2024) as the inference engine. The pipeline batches documents into work items of around 500 pages each. Each work item is then queued to run on a worker with access to a GPU for inference. Optionally, workers can coordinate using a shared cloud bucket[6], allowing for batch jobs that scale from single nodes to hundreds of nodes without the need for complicated queue management.

To balance maintaining high GPU utilization while also ensuring work items are completed quickly, each worker queues up inference for all PDF pages in a work item simultaneously, and then waits until the SGLang

[6]We use Amazon Simple Storage Service (S3), but other cloud providers or network storage solutions could be easily used.


Alignment: 0.7876106194690266

Hydrol. Earth Syst. Sci. Discuss., editor comment EC1
https://doi.org/10.5194/hess-2021-296-EC1, 2021 © Author(s) 2021. This work is distributed under the Creative Commons Attribution 4.0 License.

Powered by TCPDF (www.tcpdf.org)

Comment on hess-2021-296

Miriam Glendell (Editor)

Editor comment on "Spatially Referenced Bayesian State-Space Model of Total Phosphorus in western Lake Erie" by Timothy Maguire et al., Hydrol. Earth Syst. Sci. Discuss., https://doi.org/10.5194/hess-2021-296-EC1, 2021

Thank you for your comprehensive response to reviewers' comments. Due to the extensive nature of their comments, the reviewers will be given an opportunity to provide feedback on the proposed revisions. Please note that inclusion of additional references, suggested by the reviewers, is entirely at your own discretion. In addition to the comments provided by the reviewers, I have two small observations:

R1C1 In the Introduction, would it be possible to include a brief introduction to Bayesian state-space models and their application in freshwater ecology/ water quality modelling to set this work into a wider context for a less familiar reader, with a few key references?

R2C17 I believe that this clarification may also be useful for inclusion in the actual manuscript.

Comment on hess-2021-296

Miriam Glendell (Editor)

Editor comment on "Spatially Referenced Bayesian State-Space Model of Total Phosphorus in western Lake Erie" by Timothy Maguire et al., Hydrol. Earth Syst. Sci. Discuss., https://doi.org/10.5194/hess-2021-296-EC1, 2021

Thank you for your comprehensive response to reviewers' comments. Due to the extensive nature of their comments, the reviewers will be given an opportunity to provide feedback on the proposed revisions. Please note that inclusion of additional references, suggested by the reviewers, is entirely at your own discretion. In addition to the comments provided by the reviewers, I have two small observations:

R1C1 In the Introduction, would it be possible to include a brief introduction to Bayesian state-space models and their application in freshwater ecology/ water quality modelling to set this work into a wider context for a less familiar reader, with a few key references?

R2C17 I believe that this clarification may also be useful for inclusion in the actual manuscript.

Figure 5 Example of side-by-side evaluation tool used during development. The software used to create these comparisons is released as open-source software as part of OLMOCR.

Model	Hardware	Tokens/sec	Pages/USD	Cost per million pages
GPT-4o	API	-	80	$12,480
GPT-4o	Batch	-	160	$6,240
marker	API	-	800	$1,250
MinerU	L40s	238	1678	$596
OLMOCR	A100 80GB	1,487	**3,700**	$270
OLMOCR	L40S	906	**5,200**	**$190**
OLMOCR	H100 80GB	3,050	**5,200**	**$190**

Table 3 Inference cost comparison against other OCR methods. A100 80GB Estimated at $1.89 per hour, L40S Estimated at $0.79 per hour, H100 80GB Estimated at $2.69 per hour. Assuming 20% retries.

server has no more pending requests before proceeding to another work item in the queue.

We summarize our efforts by comparing operational costs of OLMOCR against other API and local models in Table 3. Overall, we find OLMOCR to be significantly more efficient than other pipelines. It is over 32 times cheaper than GPT-4o in batch mode; compared to other purposed-built pipelines and models, OLMOCR is over 6 times cheaper than MinerU, and $1/3^{rd}$ of the cost of marker.

4.2 Increasing Robustness

We implement several heuristics to improve reliability of OLMOCR without compromising its throughput.

Prompt format During inference, we use the same abbreviated prompt described in Section §3.2. This keeps the test time examples looking the same as what the model was trained on. If the additional tokens generated by DOCUMENT-ANCHORING cause the overall prompt to exceed 8,192 tokens, then we continue regenerating the DOCUMENT-ANCHORING tokens with exponentially lower character limits until the overall prompt is of acceptable length.

Retries Unlike when we created `olmOCR-mix-0225`, we do not enforce a specific JSON schema during inference on our fine-tuned model. This is for two reasons: first, we find that open source tools designed to force decode a sequence into a particular schema are unreliable, and that enforcing a schema which is even slightly off from what the model expects can cause generations to go out-of-domain or collapse into repetitions. Second, and most importantly, we note that, since the model was extensively fine-tuned on the structured output, it reliably adheres to the required schema without constraints. For the rare cases when JSON parsing fails, we simply retry generating from the same input sequence.

Rotations The output JSON schema includes fields for `is_rotation_valid` and `rotation_correction`. During inference, olmOCR pipeline reads these two fields and if `is_rotation_valid` is set to `true` it will rotate the page by the amount specified in `rotation_correction` and reprocess the page.

Decoding In developing olmOCR, the most common failure we experience is outputs degenerating into endless repetitions of the same token, line, or paragraph. This failure is caught automatically when the model's output either exceeds the maximum context length, or does not validate against our JSON schema. We find that increasing generation temperature from $\tau = 0.1$ up to $\tau = 0.8$ reduces the likelihood of repetitions occurring. Further, we modify olmOCR to reprocess failed pages up to N times, falling back to a plain text-based PDF extraction if the pipeline repeatedly fails. This last mitigation is aided by the fact that document-anchoring randomly samples which anchors to include in the prompt; thus, resampling can sometimes help the page process correctly by removing potentially problematic meta tokens.

We note that one one limitation of this approach is that, if retries occur often, the total generation throughput could be significantly reduced. Further, letting generations repeat up to maximum sequence length uses significant memory within SGLang. In future work, we plan to detect repeated generations sooner than at the maximum context length limit, and abort promptly.

5 Evaluating olmOCR

We conduct three evaluations for olmOCR. First, we study how faithful olmOCR is to its teacher model (Section §5.1). Then, we compare olmOCR to other PDF text extraction systems through pairwise comparisons of their outputs (Section §5.2). Finally, we quantify the usefulness olmOCR for language modeling by continued pretraining on an OLMo 2 checkpoint (OLMo et al., 2024) on content extracted and linearized with our toolkit (Section §5.3).

5.1 Alignment with Teacher Model

To compare the output of `olmOCR-7B-0225-preview` to the GPT-4o silver data in `olmOCR-mix-0225`, we build a document similarity metric which splits a document into words, uses Hirschberg's algorithm to align those words, and counts what proportion match.

We report alignment scores in Table 4. Overall, we find that `olmOCR-7B-0225-preview` has good alignment, 0.875 on average, with its teacher model. To calibrate this result, we also report GPT-4o self-alignment score of 0.954, which is simply from calling the model again; imperfect alignment here is due to resampling differences. In fact, we find that our model actually better mimics the content extraction and linearization of GPT-4o than its smaller counterpart GPT-4o mini.

When partitioning scores in low, medium, and high alignment buckets (Table 5), we find that most documents parsed with olmOCR have medium to high alignment with GPT-4o. Increasing temperature unsurprisingly leads to a wider distribution of alignment scores, as noted by the increase of low matches for $\tau = 0.8$.

5.2 Intrinsic Human Evaluation

Experimental setup To compare olmOCR against other common OCR methods, we collected pairwise human judgments of plain text produced by the three top ML-based PDF linearization tools—Marker, MinerU, and GOT-OCR 2.0—and calculating ELO ratings.

To create our evaluation set, we sample 2,017 new PDFs from the same distribution as used to create `olmOCR-mix-0225` and run each PDF through olmOCR and the linearization tools mentioned above. All

Model	Temperature τ	Alignment
GPT-4o (self-alignment)	*0.1*	*0.954*
GPT-4o mini	0.1	0.833
olmOCR	0.8	0.859
olmOCR	0.1	**0.875**

Table 4 Page-weighted alignment between GPT-4o, GPT-4o mini, and our fine-tuned model. We find that `olmOCR-7B-0225-preview` is more consistent with respect to its teacher than GPT-4o mini. Note that GPT-4o does not achieves a perfect alignment against itself due to the probabilistic nature of autoregressive decoding.

Name	Low match	Medium match	High match
GPT-4o (self alignment)	*38*	*218*	*965*
GPT-4o mini	214	478	529
olmOCR (τ = 0.1)	158	363	700
olmOCR (τ = 0.8)	195	390	636

Table 5 Match-up between olmOCR and different models compared to the `olmOCR-mix-0225` dataset. Low match indicates $<$ 70% alignment, Medium match is 70-95% alignment, High match is $>$95% alignment.

other linearization tools were installed from either PyPI or Github according to their publicly available instructions as of January 14th, 2025. GOT-OCR 2.0 was configured in "format" mode, but otherwise all comparisons were done against default settings.

We then sampled 2,000 comparison pairs (same PDF, different tool). We asked 11 data researchers and engineers at Ai2 to assess which output was the higher quality representation of the original PDF, focusing on reading order, comprehensiveness of content and representation of structured information. The user interface used is similar to that in Figure 5. Exact participant instructions are listed in Appendix B.

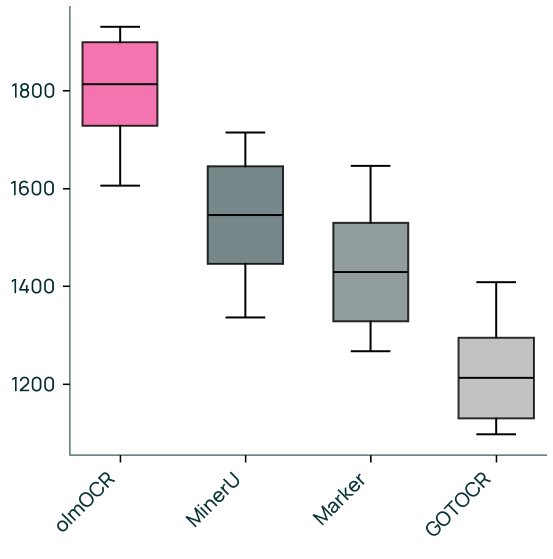


Figure 6 ELO ranking of olmOCR vs other popular PDF content extraction tools.

Evaluation results We collected a total of 452 judgments where a participant expressed a preference between two models (the remaining 1,548 pairs were either skipped for being too similar, or marked as invalid). On average, this is 75 judgments per pair of tools. We calculate ELO ratings starting from a base of 1500 and report the average of 100 simulations to avoid ordering effects in ELO calculations; for 95% confidence intervals, we use bootstrapping with 5000 resamples.

We visualize our results in Figure 6. olmOCR achieves an ELO score over 1800, far exceeding all other PDF linearization tools.

5.3 Downstream Evaluation

Experimental setup To assess the impact of improved PDF linearization, we experiment using the pre-mid-training checkpoint of `OLMo-2-1124-7B` and perform mid-training (that is, continued pretraining while linearly decaying learning rate to zero) using content extracted from the same PDFs but with different linearization tools. This procedure for assessing the quality of new data sources was introduced as domain upsampling in Blakeney et al. (2024), annealing in Grattafiori et al. (2024), and micro-annealing in OLMo et al. (2024).

For our baseline, we use PDF extracted tokens from `peS2o` (Soldaini and Lo, 2023), a collection of over 58B tokens extracted from academic PDFs which were derived using Grobid in S2ORC (Lo et al., 2020) and further cleaned with heuristics for language modeling. To represent olmOCR, we identify the same documents used in peS2o, acquire their source PDFs from the upstream S2ORC pipeline, and reprocess them using olmOCR. For both of these, we perform 50B tokens worth of mid-training.

PeS2o version	Average	MMLU	ARC_C	DROP	HSwag	NQ	WinoG
Grobid + rules (Soldaini and Lo, 2023)	53.9	**61.1**	75.0	42.3	57.4	**29.4**	**58.3**
olmOCR	**55.2**	**61.1**	**76.4**	**43.7**	**62.6**	29.1	58.0

Table 6 Comparison on OLMo et al. (2024) downstream evaluation tasks of `OLMo-2-7B-1124` on 50B of original peS2o tokens vs 50B tokens from the same source PDFs but processed with olmOCR.

Evaluation results We use the same downstream evaluation setup as in OLMo et al. (2024); results are in Table 6. Overall, we see an improvement in +1.3 percentage points on average across a number of core benchmark tasks, which most performance improvements in ARC Challenge and DROP.

6 Conclusion

We introduce olmOCR, an open-source toolkit for converting PDF documents into clean plain text. Our approach combines document-anchoring, a novel prompting technique that leverages available metadata in born-digital PDFs, with a fine-tuned 7B parameter vision language model to achieve results competitive with closed commercial solutions at a fraction of the cost. Our efficient inference pipeline which ships as part of this release contains everything needed to start converting anything from single documents to million-page archives of PDFs. We hope olmOCR's ability to efficiently process millions of documents will help unlock new sources of training data for language models, particularly from high-quality PDF documents that are currently underrepresented in existing datasets that rely heavily solely on crawled web pages.

Acknowledgments

This work would not be possible without the support of our colleagues at Ai2. We thank:

- Jon Borchardt, Byron Bischoff, Aaron Sarnat, Huy Tran, Sam Skjonsberg, Eric Marsh, and Chris Newell for help setting up the live demo,
- Taira Anderson, Sruthi Sreeram for program management support,
- Will Smith and Crystal Nam for legal guidance,
- and Michael Schmitz, Caitlin Wittlif and Carissa Schoenick for various indirect support.

We also thank Benjamin Charles Germain Lee for helpful feedback and suggestions around evaluation and potential use cases.

References

GROBID. https://github.com/kermitt2/grobid, 2008–2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv [cs.CV]*, February 2025.

Cody Blakeney, Mansheej Paul, Brett W. Larsen, Sean Owen, and Jonathan Frankle. Does your data spark joy? performance gains from domain upsampling at the end of training, 2024. https://arxiv.org/abs/2406.03476.

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural optical understanding for academic documents, 2023. https://arxiv.org/abs/2308.13418.

Patrick Emond. Lingua-py: Natural language detection for python, 2025. https://github.com/pemistahl/lingua-py. Accessed: 2025-01-06.

Google. Explore document processing capabilities with the gemini API. https://web.archive.org/web/20250224064040/https://ai.google.dev/gemini-api/docs/document-processing?lang=python, 2025. Accessed: 2025-2-23.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathurx, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian

Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. https://arxiv.org/abs/2407.21783.

John Gruber. Markdown. https://daringfireball.net/projects/markdown/, 2004. Accessed: 2025-2-23.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. https://arxiv.org/abs/2106.09685.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, 2021. https://api.semanticscholar.org/CorpusID:250924870.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. DataComp-LM: In search of the next generation of training sets for language models. *arXiv [cs.LG]*, June 2024.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The semantic scholar open research corpus. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.447. https://aclanthology.org/2020.acl-main.447/.

Kyle Lo, Zejiang Shen, Benjamin Newman, Joseph Chang, Russell Authur, Erin Bransom, Stefan Candra, Yoganand Chandrasekhar, Regan Huff, Bailey Kuehl, Amanpreet Singh, Chris Wilhelm, Angele Zamarron, Marti A. Hearst, Daniel Weld, Doug Downey, and Luca Soldaini. PaperMage: A unified toolkit for processing, representing, and manipulating visually-rich scientific documents. In Yansong Feng and Els Lefever, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 495–507, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-demo.45. https://aclanthology.org/2023.emnlp-demo.45/.

Kyle Lo, Joseph Chee Chang, Andrew Head, Jonathan Bragg, Amy X. Zhang, Cassidy Trier, Chloe Anastasiades, Tal August, Russell Authur, Danielle Bragg, Erin Bransom, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Yen-Sung Chen, Evie Yu-Yen Cheng, Yvonne Chou, Doug Downey, Rob Evans, Raymond Fok, Fangzhou Hu, Regan Huff, Dongyeop Kang, Tae Soo Kim, Rodney Kinney, Aniket Kittur, Hyeonsu B. Kang, Egor Klevak, Bailey Kuehl, Michael J. Langan, Matt Latzke, Jaron Lochner, Kelsey MacMillan, Eric Marsh, Tyler Murray, Aakanksha Naik, Ngoc-Uyen Nguyen, Srishti Palani, Soya Park, Caroline Paulic, Napol Rachatasumrit, Smita Rao, Paul Sayre, Zejiang Shen, Pao Siangliulue, Luca Soldaini, Huy Tran, Madeleine van Zuylen, Lucy Lu Wang, Christopher Wilhelm, Caroline Wu, Jiangjiang Yang, Angele Zamarron, Marti A. Hearst, and Daniel S. Weld. The semantic reader project. *Commun. ACM*, 67(10):50–61, September 2024. ISSN 0001-0782. doi: 10.1145/3659096. https://doi.org/10.1145/3659096.

S Mori, C Y Suen, and K Yamamoto. Historical review of OCR research and development. *Proceedings of the IEEE. Institute of Electrical and Electronics Engineers*, 80(7):1029–1058, July 1992. ISSN 0018-9219,1558-2256. doi: 10.1109/5.156468.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James Validad Miranda, Jacob Daniel Morrison, Tyler C. Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Chris Wilhelm, Michael Wilson, Luke S. Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hanna Hajishirzi. 2 OLMo 2 Furious. *arXiv preprint*, 2024. https://api.semanticscholar.org/CorpusID:275213098.

Vik Paruchuri. Marker: Convert pdf to markdown + json quickly with high accuracy, 2025. https://github.com/VikParuchuri/marker. Version 1.4.0.

PDF Association staff. Pdf in 2016: Broader, deeper, richer. *PDF Association*, December 2015. https://pdfa.org/pdf-in-2016-broader-deeper-richer/.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra-Aimée Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *ArXiv*, abs/2306.01116, 2023. https://api.semanticscholar.org/CorpusID:259063761.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. https://arxiv.org/abs/2406.17557.

Poppler. Poppler: A pdf rendering library. https://poppler.freedesktop.org/, 2005–2025. Accessed: 2025-01-06.

PyPDF. Pypdf: A pure-python pdf library. https://github.com/py-pdf/pypdf, 2012–2025. Accessed: 2025-01-06.

Zejiang Shen, Kyle Lo, Lucy Lu Wang, Bailey Kuehl, Daniel S. Weld, and Doug Downey. VILA: Improving structured content extraction from scientific PDFs using visual layout groups. *Transactions of the Association for Computational Linguistics*, 10:376–392, 2022. doi: 10.1162/tacl_a_00466. https://aclanthology.org/2022.tacl-1.22/.

Ray W Smith. History of the tesseract OCR engine: what worked and what didn't. In Richard Zanibbi and Bertrand Coüasnon, editors, *Document Recognition and Retrieval XX*, volume 8658, page 865802. SPIE, February 2013. doi: 10.1117/12.2010051.

Luca Soldaini and Kyle Lo. peS2o (Pretraining Efficiently on S2ORC) Dataset. Technical report, Allen Institute for AI, 2023. ODC-By, https://github.com/allenai/pes2o.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research, 2024. https://arxiv.org/abs/2402.00159.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. Mineru: An open-source solution for precise document content extraction, 2024a. https://arxiv.org/abs/2409.18839.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024b. https://arxiv.org/abs/2409.12191.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, et al. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. *arXiv preprint arXiv:2409.01704*, 2024.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs, 2024. https://arxiv.org/abs/2312.07104.

A Appendix

`olmOCR-mix-0225` construction prompt for gpt-4o

The prompt below was used to create the silver dataset that was then used to fine-tune our model. The `{base_text}` is replaced with the output of DOCUMENT-ANCHORING.

```
Below is the image of one page of a PDF document, as well as some raw textual content that
    was previously extracted for it that includes position information for each image and
    block of text (The origin [0x0] of the coordinates is in the lower left corner of the
    image).
Just return the plain text representation of this document as if you were reading it
    naturally.
Turn equations into a LaTeX representation, and tables into markdown format. Remove the
    headers and footers, but keep references and footnotes.
Read any natural handwriting.
This is likely one page out of several in the document, so be sure to preserve any sentences
     that come from the previous page, or continue onto the next page, exactly as they are.
If there is no text at all that you think you should read, you can output null.
Do not hallucinate.
RAW_TEXT_START
{base_text}
RAW_TEXT_END
```

JSON Schema

```
"json_schema": {
            "name": "page_response",
            "schema": {
                "type": "object",
                "properties": {
                    "primary_language": {
                        "type": ["string", "null"],
                        "description": "The primary language of the text using two-letter
                            codes or null if there is no text at all that you think you
                            should read.",
                    },
                    "is_rotation_valid": {
                        "type": "boolean",
                        "description": "Is this page oriented correctly for reading? Answer
                            only considering the textual content, do not factor in the
                            rotation of any charts, tables, drawings, or figures.",
                    },
                    "rotation_correction": {
                        "type": "integer",
                        "description": "Indicates the degree of clockwise rotation needed if
                             the page is not oriented correctly.",
                        "enum": [0, 90, 180, 270],
                        "default": 0,
                    },
                    "is_table": {
                        "type": "boolean",
                        "description": "Indicates if the majority of the page content is in
                            tabular format.",
                    },
                    "is_diagram": {
                        "type": "boolean",
                        "description": "Indicates if the majority of the page content is a
                            visual diagram.",
                    },
                    "natural_text": {
                        "type": ["string", "null"],
                        "description": "The natural text content extracted from the page.",
                    },
                },
                "additionalProperties": False,
                "required": [
                    "primary_language",
                    "is_rotation_valid",
                    "rotation_correction",
                    "is_table",
                    "is_diagram",
                    "natural_text",
                ],
            },
            "strict": True,
        },
```

`olmOCR-7B-0225-preview` prompt

The prompt below is used to prompt the fine-tuned model, using the same rule to replace `{base_text}` with the output of DOCUMENT-ANCHORING.

```
Below is the image of one page of a document, as well as some raw textual content that was
    previously extracted for it.
Just return the plain text representation of this document as if you were reading it
    naturally.
Do not hallucinate.
RAW_TEXT_START
{base_text}
RAW_TEXT_END
```

`olmOCR-mix-0225` Classification Prompt

The prompt and structured schema below was used to classify a sample of documents from `olmOCR-mix-0225` as reported in Table 2.

```
This is an image of a document page, please classify it into one of the following categories
     that best overall summarizes its nature: academic, legal, brochure, slideshow, table,
    diagram, or other. Also determine the primary language of the document and your
    confidence in the classification (0-1).
```

```
class DocumentCategory(str, Enum):
    ACADEMIC = "academic"
    LEGAL = "legal"
    BROCHURE = "brochure"
    SLIDESHOW = "slideshow"
    TABLE = "table"
    DIAGRAM = "diagram"
    OTHER = "other"

class DocumentClassification(BaseModel):
    category: DocumentCategory
    language: str
    confidence: float
```

B ELO Evaluation Instructions

In Section 5.2, we asked participants to compare the output of various common OCR tools against OLMOCR. Participants were given the instructions below, and presented with a document page, and the output of two random tools. They could then select which output was better, or select "Both Good", "Both Bad", or "Invalid PDF" any of which would not count the comparison in the ELO ranking.

Instructions to participants

```
Compare the text in the two fields, and select which one better represents the contents of
    the document.

REMINDER: This is not about "the most faithful OCR", but "this OCR output seems really
    useful for training LMs"

- Does the text capture all of the meaningful content in the document in a natural order?
- Are the words correct (no weird incorrect words or split words)
- Is the whitepsace sensical?
- Is the useless header/footer content removed?
- Do the tables/equations look okay?

There is not a strict preference between Markdown and LaTeX, most importantly you should
    evaluate it on the text content, not which method was used to format it.

If you are not sure, or the document is in a language other than english, you can skip that
    entry, or mark "both good" "both bad", "invalid pdf".
```

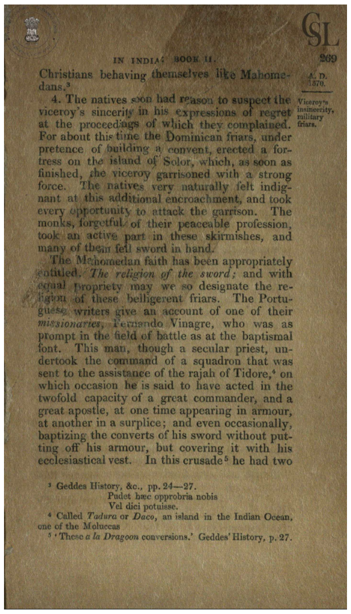
ELO data

Table 7 Pairwise Win/Loss Statistics Between Models

Model Pair	Wins	Win Rate (%)
olmOCR vs. Marker	49/31	**61.3**
olmOCR vs. GOTOCR	41/29	**58.6**
olmOCR vs. MinerU	55/22	**71.4**
Marker vs. MinerU	53/26	67.1
Marker vs. GOTOCR	45/26	63.4
GOTOCR vs. MinerU	38/37	50.7
Total	452	

C Example output

Below are some sample outputs on particularly challenging data. olmOCR, MinerU, GOT-OCR 2.0 and Marker run with default settings.



CSL

IN INDIA: BOOK II. 269

Christians behaving themselves like Mahomedans.[3]

A. D. 1570.

Viceroy's insincerity, military friars.

4. The natives soon had reason to suspect the viceroy's sincerity in his expressions of regret at the proceedings of which they complained. For about this time the Dominican friars, under pretence of building a convent, erected a fortress on the island of Solor, which, as soon as finished, the viceroy garrisoned with a strong force. The natives very naturally felt indignant at this additional encroachment, and took every opportunity to attack the garrison. The monks, forgetful of their peaceable profession, took an active part in these skirmishes, and many of them fell sword in hand.

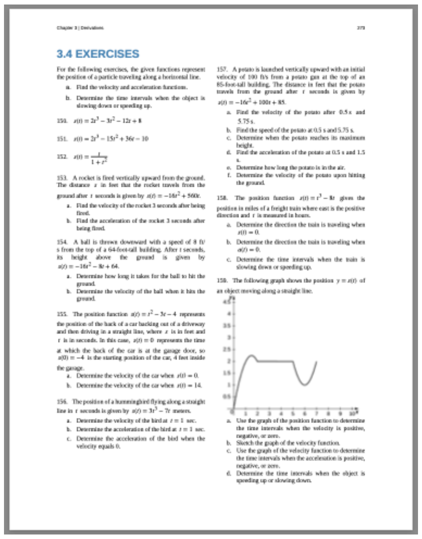
The Mahomedan faith has been appropriately entitled, *The religion of the sword;* and with equal propriety may we so designate the religion of these belligerent friars. The Portuguese writers give an account of one of their *missionaries*, Fernando Vinagre, who was as prompt in the field of battle as at the baptismal font. This man, though a secular priest, undertook the command of a squadron that was sent to the assistance of the rajah of Tidore,[4] on which occasion he is said to have acted in the twofold capacity of a great commander, and a great apostle, at one time appearing in armour, at another in a surplice; and even occasionally, baptizing the converts of his sword without putting off his armour, but covering it with his ecclesiastical vest. In this crusade[5] he had two

[3] Geddes History, &c., pp. 24—27.
Pudet hæc opprobria nobis
Vel dici potuisse.

[4] Called *Tadura* or *Daco*, an island in the Indian Ocean, one of the Moluccas

[5] 'These *a la Dragoon* conversions.' Geddes' History, p. 27.

olmOCR	MinerU	GOT-OCR 2.0	Marker
Christians behaving themselves like Mahomedans. 4. The natives soon had reason to suspect the viceroy's sincerity in his expressions of regret at the proceedings of which they complained. For about this time the Dominican friars, under pretence of building a convent, erected a fortress on the island of Solor, which, as soon as finished, the viceroy garrisoned with a strong force. The natives very naturally felt indignant at this additional encroachment, and took every opportunity to attack the garrison. The monks, forgetful of their peaceable profession, took an active part in these skirmishes, and many of them fell sword in hand. The Mahomedan faith has been appropriately entitled, The religion of the sword; and with equal propriety may we so designate the religion of these belligerent friars. The Portuguese writers give an account of one of their missionaries, Fernando Vinagre, who was as prompt in the field of battle as at the baptismal font. This man, though a secular priest, undertook the command of a squadron that was sent to the assistance of the rajah of Tidore, on which occasion he is said to have acted in the twofold capacity of a great commander, and a great apostle, at one time appearing in armour, at another in a surplice; and even occasionally, baptizing the converts of his sword without putting off his armour, but covering it with his ecclesiastical vest. In this crusade he had two	ININDIASY BOOKU Christians bchaving.themselves like Mahome dans.3 4.The natives soon had reason to suspect ihe viceroy's sincerity in his expressions of regret at the proceedings of which they complained. For about this time the Dominican friars,under pretenceof building a convent,erected a for tress on the island of Solorwhich,as soon as finishedthe viceroy garrisoned with a strong force. The natives very naturally felt indig nant at this additional encroachment, and took every pportunity to attack the garrison.The monks,forgetful of their peaceable profession took an active part in these skirmishes, and many of tbein feil sword in hand. TheMahornedan faithhas been appropriately ntitled.The religion of the swordand with equal propriety may we so designate the region of these belligerent friars.The Portugueswriters give an account of one of their missionarzes,femando Vinagre,who was as prompt in the field of battle as at the baptismal font. This man, though a secular priest, undertook the command of a squadron that was sent to the assistance of the rajah of Tidore,4 on which occasion he is said to have acted in the twofold capacity of a great commander, and a great apostle, at one time appearing in armour, at another in a surplice;and even occasionally baptizing the converts of his sword without put ting off his armour, but covering it with his ecclesiastical vest.In this crusadehe had two	IN INDIA: BOOK U 269 Christians behaving themselves like Mahome-1670. 4. The natives son had reason to suspect the Viceroy' s vice roy' s sincerity in his expressions of regret in s in e eri ty at the proceedings of which they complained. fl it ars. For about this time the Dominican f mars, under pre ten ce of building a convent, erected a for- tress on the island of Sol or, which, as soon as finished, the vice roy garrisoned with a strong force. The natives very naturally felt indig- nant at this additional encroachment, and took every opportunity to attack the garrison. The monks, forgetful of their peaceable profession, took an active part in these skirmishes, and many of the n fell sword in hand. The Mh on med an faith has been appropriately entitled. The religion of the sword; and with e ral Tropriety may we so designate the re- gian of these belligerent friars. The Port u- gue s writers give an account of one of their mission are s, Fer endo Vina gre, who was as prompt in the fe ld of battle as at the baptismal font. This man, though a secular priest, un- der took the command of a squadron that was sent to the assistance of the rajah of Tidore, on which occasion he is said to have acted in the twofold capacity of a great commander, and a great apostle, at one time appearing in armour, at another in a surplice; and even occasionally, baptizing the converts of his sword without put- ting off his armour, but covering it with his ecclesiastical vest. In this crusade he had two 3 Ged des History, & c. , pp. 24-27. P ude th aec opp rob ria nobis Vel die ipo tui sse. Called Tadur u or Daco, an island in the Indian Ocean, one of the Mol ucc as These a laDra goon conversions. Ged des History, p. 27.	## **IN INDIA *** BOOK TI. S69 Christians behaving themselves like Ma borne- a. dans.3 .“5/0- *t>.* The natives soon had reason to suspect the viceroy, viceroy's sincerity in his expressions of regret at the proceedings of which they complained. “n.“' For about this time the Dominican friars, under pretence of building a. convent, erected a fortress on the island of Sol or, which, as soon as finished, the viceroy garrisoned with a strong force. The natives' very naturally felt indig-S nant at this additional encroachment, and took every opportunity to attack the garrison. The monks, forgetful/ of their peaceable profession, took an active part in these skirmishes, and many of tbg.tr fell sword in hand. The i'lfinomedan faith has been appropriately entitled., 'The religion of the sword',; and with equal propriety may we so designate the re- . i'gv.m of these belligerent friars. The Portugu writers give an account of one of their 'missionaries,' Fernando Vinagre, who was as prompt in the field of battle as at the baptismal font. This man, though a secular priest, undertook the command of a squadron that was I sent to the assistance of the rajah of Tidore,4 on which occasion he is said to have acted in the twofold capacity of a great commander, and a great apostle, at one time appearing in armour, ; at another in a surplice; and even occasionally, baptizing the converts of his sword without putting off his armour, but covering it with his ecclesiastical vest. In this crusade5 he had two 3 Geddes History, &c., pp. 24—27. Pudet hæc opprobria nobis Vel dici potuisse. 4 Called 'T a d u ra' or 'D a c o,' an island in the Indian Ocean, one of the Moluccas 5 'These 'a la D ra g o o n' conversions.' Geddes' History, p. 27.



Chapter 3 | Derivatives 273

3.4 EXERCISES

For the following exercises, the given functions represent the position of a particle traveling along a horizontal line.

a. Find the velocity and acceleration functions.

b. Determine the time intervals when the object is slowing down or speeding up.

150. $s(t) = 2t^3 - 3t^2 - 12t + 8$

151. $s(t) = 2t^3 - 15t^2 + 36t - 10$

152. $s(t) = \frac{t}{1+t^2}$

153. A rocket is fired vertically upward from the ground. The distance s in feet that the rocket travels from the ground after t seconds is given by $s(t) = -16t^2 + 560t$.

a. Find the velocity of the rocket 3 seconds after being fired.

b. Find the acceleration of the rocket 3 seconds after being fired.

154. A ball is thrown downward with a speed of 8 ft/s from the top of a 64-foot-tall building. After t seconds, its height above the ground is given by $s(t) = -16t^2 - 8t + 64$.

a. Determine how long it takes for the ball to hit the ground.

b. Determine the velocity of the ball when it hits the ground.

155. The position function $s(t) = t^2 - 3t - 4$ represents the position of the back of a car backing out of a driveway and then driving in a straight line, where s is in feet and t is in seconds. In this case, $s(t) = 0$ represents the time at which the back of the car is at the garage door, so $s(0) = -4$ is the starting position of the car, 4 feet inside the garage.

a. Determine the velocity of the car when $s(t) = 0$.

b. Determine the velocity of the car when $s(t) = 14$.

156. The position of a hummingbird flying along a straight line in t seconds is given by $s(t) = 3t^3 - 7t$ meters.

a. Determine the velocity of the bird at $t = 1$ sec.

b. Determine the acceleration of the bird at $t = 1$ sec.

c. Determine the acceleration of the bird when the velocity equals 0.

157. A potato is launched vertically upward with an initial velocity of 100 ft/s from a potato gun at the top of an 85-foot-tall building. The distance in feet that the potato travels from the ground after t seconds is given by $s(t) = -16t^2 + 100t + 85$.

a. Find the velocity of the potato after 0.5 s and 5.75 s.

b. Find the speed of the potato at 0.5 s and 5.75 s.

c. Determine when the potato reaches its maximum height.

d. Find the acceleration of the potato at 0.5 s and 1.5 s.

e. Determine how long the potato is in the air.

f. Determine the velocity of the potato upon hitting the ground.

158. The position function $s(t) = t^3 - 8t$ gives the position in miles of a freight train where east is the positive direction and t is measured in hours.

a. Determine the direction the train is traveling when $s(t) = 0$.

b. Determine the direction the train is traveling when $a(t) = 0$.

c. Determine the time intervals when the train is slowing down or speeding up.

158. The following graph shows the position $y = s(t)$ of an object moving along a straight line.

a. Use the graph of the position function to determine the time intervals when the velocity is positive, negative, or zero.

b. Sketch the graph of the velocity function.

c. Use the graph of the velocity function to determine the time intervals when the acceleration is positive, negative, or zero.

d. Determine the time intervals when the object is speeding up or slowing down.

olmOCR

3.4 EXERCISES
For the following exercises, the given functions represent the position of a particle traveling along a horizontal line.
a. Find the velocity and acceleration functions.
b. Determine the time intervals when the object is slowing down or speeding up.
150. $s(t) = 2t^3 - 3t^2 - 12t + 8$
151. $s(t) = 2t^3 - 15t^2 + 36t - 10$
152. $s(t) = \frac{t}{1+t^2}$
153. A rocket is fired vertically upward from the ground. The distance s in feet that the rocket travels from the ground after t seconds is given by $s(t) = -16t^2 + 560t$.
a. Find the velocity of the rocket 3 seconds after being fired.
b. Find the acceleration of the rocket 3 seconds after being fired.
154. A ball is thrown downward with a speed of 8 ft/s from the top of a 64-foot-tall building. After t seconds, its height above the ground is given by $s(t) = -16t^2 - 8t + 64$.
a. Determine how long it takes for the ball to hit the ground.
b. Determine the velocity of the ball when it hits the ground.
155. The position function $s(t) = t^2 - 3t - 4$ represents the position of the back of a car backing out of a driveway and then driving in a straight line, where s is in feet and t is in seconds. In this case, $s(t) = 0$ represents the time at which the back of the car is at the garage door, so $s(0) = -4$ is the starting position of the car, 4 feet inside the garage.
a. Determine the velocity of the car when $s(t) = 0$.
b. Determine the velocity of the car when $s(t) = 14$.
156. The position of a hummingbird flying along a straight line in t seconds is given by $s(t) = 3t^3 - 7t$ meters.
a. Determine the velocity of the bird at $t = 1$ sec.
b. Determine the acceleration of the bird at $t = 1$ sec.
c. Determine the acceleration of the bird when the velocity equals 0.
157. A potato is launched vertically upward with an initial velocity of 100 ft/s from a potato gun at the top of an 85-foot-tall building. The distance in feet that the potato travels from the ground after t seconds is given by $s(t) = -16t^2 + 100t + 85$. . . .

MinerU

# 3.4 EXERCISES
For the following exercises, the given functions represent the position of a particle traveling along a horizontal line.
a. Find the velocity and acceleration functions. b. Determine the time intervals when the object is slowing down or speeding up.
150. $s(t) = 2t^3 - 3t^2 - 12t + 8$ 151. $s(t) = 2t^3 - 15t^2 + 36t - 10$ 152. $s(t) = \frac{t}{1+t^2}$
153. A rocket is fired vertically upward from the ground. The distance s in feet that the rocket travels from the ground after t seconds is given by $s(t) = -16t^2 + 560t$, .
a. Find the velocity of the rocket 3 seconds after being fired. b. Find the acceleration of the rocket 3 seconds after being fired.
154. A ball is thrown downward with a speed of 8 ft/ s from the top of a 64-foot-tall building. After t seconds, its height above the ground is given by $s(t) = -16t^2 - 8t + 64$. .
a. Determine how long it takes for the ball to hit the ground. b. Determine the velocity of the ball when it hits the ground.
155. The position function $s(t) = t^2 - 3t - 4$ represents the position of the back of a car backing out of a driveway and then driving in a straight line, where s is in feet and t is in seconds. In this case, $s(t) = 0$ represents the time at which the back of the car is at the garage door, so $s(0) = -4$ is the starting position of the car, 4 feet inside the garage.
a. Determine the velocity of the car when $s(t) = 0$. b. Determine the velocity of the car when $s(t) = 14$.
156. The position of a hummingbird flying along a straight line in t seconds is given by $s(t) = 3t^3 - 7t$ meters.
a. Determine the velocity of the bird at $t = 1$ sec. b. Determine the acceleration of the bird at $t = 1$ sec. c. Determine the acceleration of the bird when the velocity equals 0.
157. A potato is launched vertically upward with an initial velocity of 100 ft/s from a potato gun at the top of an 85-foot-tall building. The distance in feet that the potato travels from the ground after t seconds is given by $s(t) = -16t^2 + 100t + 85$. .
. . .

GOT-OCR 2.0

Chapter 3 | Derivatives 273 3.4 EXERCISES For the following exercises, the given functions represent the position of a particle traveling along a horizontal line. a. Find the velocity and acceleration functions. b. Determine the time intervals when the object is slowing down or speeding up. 150. s(t) = 2t3 −3t2 −12t + 8 151. s(t) = 2t3 −15t2 + 36t −10 152. s(t) = t 1 + t2 153. A rocket is fired vertically upward from the ground. The distance s in feet that the rocket travels from the ground after t seconds is given by s(t) = −16t2 + 560t. a. Find the velocity of the rocket 3 seconds after being fired. b. Find the acceleration of the rocket 3 seconds after being fired. 154. A ball is thrown downward with a speed of 8 ft/ s from the top of a 64-foot-tall building. After t seconds, its height above the ground is given by s(t) = −16t2 −8t + 64. a. Determine how long it takes for the ball to hit the ground. b. Determine the velocity of the ball when it hits the ground. 155. The position function s(t) = t2 −3t −4 represents the position of the back of a car backing out of a driveway and then driving in a straight line, where s is in feet and t is in seconds. In this case, s(t) = 0 represents the time at which the back of the car is at the garage door, so s(0) = −4 is the starting position of the car, 4 feet inside the garage. a. Determine the velocity of the car when s(t) = 0. b. Determine the velocity of the car when s(t) = 14. 156. The position of a hummingbird flying along a straight line in t seconds is given by s(t) = 3t3 −7t 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4 4 4 2 2 2 2 2 2 2 2 2 0 1 1 1 1 1 1 1 1 3 4 4 4 4 4 4 4 4 3 4 4 4 4 4 4 4 4 2 3 3 3 3 3 3 3 3 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 0 2 2 2 2 2 2 2 2 2 5 5 5 5 5 5 5 5 5 1 1 1 1 1 1 1 1 1 5 5 5 5 5 5 5 5 5 0 0 0 0 0 0 0 0 0 5 5 5 5 5 5 5 5 5 3 3 3 3 3 3 3 3 3 5 5 5 5 5 5 5 5 5 2 2 2 2 2 2 2 2 2 a. Use the graph of the position function to determine the time intervals when the velocity is positive, negative, or zero. b. Sketch the graph of the velocity function. c. Use the graph of the velocity function to determine the time intervals when the acceleration is positive, negative, or zero. d. Determine the time intervals when the object is speeding up or slowing down. . . .

Marker

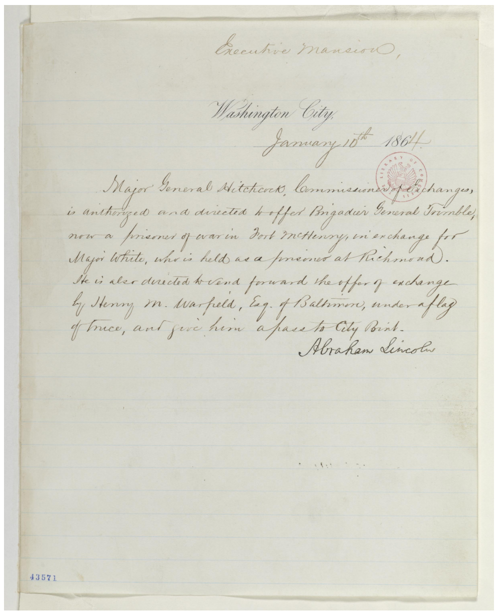
## **3.4 EXERCISES**
For the following exercises, the given functions represent the position of a particle traveling along a horizontal line.
- a. Find the velocity and acceleration functions. - b. Determine the time intervals when the object is slowing down or speeding up.

150. $s(t) = 2t^3 - 3t^2 - 12t + 8$

151. $s(t) = 2t^3 - 15t^2 + 36t - 10t$

152. $s(t) = \frac{t}{1+t^2}$

153. A rocket is fired vertically upward from the ground. The distance *s* in feet that the rocket travels from the ground after *t* seconds is given by *s*(*t*) = −16*t* 2 + 560*t*.
- a. Find the velocity of the rocket 3 seconds after being fired. - b. Find the acceleration of the rocket 3 seconds after being fired.
154. A ball is thrown downward with a speed of 8 ft/ s from the top of a 64-foot-tall building. After *t* seconds, its height above the ground is given by *s*(*t*) = −16*t* 2 − 8*t* + 64.
- a. Determine how long it takes for the ball to hit the ground. - b. Determine the velocity of the ball when it hits the ground.
155. The position function *s*(*t*) = *t* 2 − 3*t* − 4 represents the position of the back of a car backing out of a driveway and then driving in a straight line, where *s* is in feet and *t* is in seconds. In this case, *s*(*t*) = 0 represents the time at which the back of the car is at the garage door, so *s*(0) = −4 is the starting position of the car, 4 feet inside the garage.
- a. Determine the velocity of the car when *s*(*t*) = 0. - b. Determine the velocity of the car when *s*(*t*) = 14.
156. The position of a hummingbird flying along a straight line in *t* seconds is given by *s*(*t*) = 3*t* 3 − 7*t* meters.
- a. Determine the velocity of the bird at *t* = 1 sec. - b. Determine the acceleration of the bird at *t* = 1 sec. - c. Determine the acceleration of the bird when the velocity equals 0.
. . .


Executive Mansion,

Washington City,
January 10th 1864.

Major General Hitchcock, Commissioner of Exchanges, is authorized and directed to offer Brigadier General Trimble, now a prisoner of war in Fort McHenry, in exchange for Major White, who is held as a prisoner at Richmond. He is also directed to send forward the offer of exchange by Henry M. Warfield, Esq. of Baltimore, under a flag of truce, and give him a pass to City Point.

Abraham Lincoln

43571

olmOCR	MinerU	GOT-OCR 2.0	Marker
Executive Mansion, Washington City, January 15th, 1864 Major General Hitchcock, Commissioner of Exchanges, is authorized and directed to offer Brigadier General Trimble, now a prisoner of war in Fort McHenry, in exchange for Major White, who is held as a prisoner at Richmond. He is also directed to send forward the offer of exchange by Henry M. Warfield, Esq. of Baltimore, under a flag of truce, and give him a pass to City Point. Abraham Lincoln	*No text produced.*	43571 Bachington City January 10th 1864. Major General Architect, Commissioner of aivachangera is authorized and directed by ffeed Bngader General Trelmble, new a firemen of war in Fert nchery in exchange for Mayor White, who held a a firemen at Hillmannd. He is aker conducted by end forward the offer of exchange by Henry in. Warfield, Lag. of Balthmore, under a flag of three, and five him afaies to City Bink. Abraham Lincoln	necuhve Mansion Vastington amany layor Seneral Hitchcocks Commissioner of Cachanges, is anthonged and directed to offer Bingadier General Trin prisoner of war in Fort Inctienny, in exchange now w Major White, who is held as a preises at Richmond Ite is also directed to vand forwards the offer of exchange by Stenny in. Warfield, Eag. of Baltimore, under aflag 11 mice, and give him apass to tity Point. Abrakan Sincolus