BigDocs: An Open and Permissively-Licensed Dataset for Training Multimodal Models on Document and Code Tasks

Juan A. Rodriguez[1,2,3*] **Xiangru Jian**[1,4*] **Siba Smarak Panigrahi**[1,2*] **Tianyu Zhang**[1,2,5*]
Aarash Feizi[1,2,6*] **Abhay Puri**[1*] **Akshay Kalkunte**[1*] **François Savard**[1*]
Ahmed Masry[1,7†] **Shravan Nayak**[1,2,5†] **Rabiul Awal**[1,2,5†] **Mahsa Massoud**[1,2,6†]
Amirhossein Abaskohi[1,8†] **Zichao Li**[1,2,6†] **Suyuchen Wang**[2,5†] **Pierre-André Noël**[1†]
Mats Leon Richter[1†] **Saverio Vadacchino**[1] **Shubham Agarwal**[1,2] **Sanket Biswas** [9]
Sara Shanian[1] **Ying Zhang**[1] **Noah Bolger**[1] **Kurt MacDonald**[1] **Simon Fauvel**[1]
Sathwik Tejaswi[1] **Srinivas Sunkara**[1] **Joao Monteiro**[1] **Krishnamurthy DJ Dvijotham**[1]
Torsten Scholak[1] **Nicolas Chapados**[1] **Sepideh Kharagani**[1] **Sean Hughes**[1] **M. Özsu**[4]
Siva Reddy[1,2,6,10] **Marco Pedersoli**[1,3] **Yoshua Bengio**[2,5,10] **Christopher Pal**[1,2,10,11]
Issam Laradji[1,8] **Spandana Gella**[1] **Perouz Taslakian**[1] **David Vazquez**[1] **Sai Rajeswar**[1,2]

[1]ServiceNow [2]Mila [3]École de Technologie Supérieure [4]University of Waterloo
[5]Université de Montréal [6]McGill University [7]York University [8]University of British Columbia
[9]Universitat Autònoma de Barcelona [10]CIFAR AI Chair [11]Polytechnique Montréal

[*]First Author Equal contribution [†]Second Author Equal contribution

Abstract

Multimodal AI has the potential to significantly enhance document-understanding tasks, such as processing receipts, understanding workflows, extracting data from documents, and summarizing reports. Code generation tasks that require long-structured outputs can also be enhanced by multimodality. Despite this, their use in commercial applications is often limited due to limited access to relevant training data and restrictive licensing, which hinders open access. To address these limitations, we introduce BigDocs-7.5M, a high-quality, open-access dataset comprising 7.5 million multimodal documents across 30 tasks. We use an efficient data curation process to ensure our data is high quality and license-permissive. Our process emphasizes accountability, responsibility, and transparency through filtering rules, traceable metadata, and careful content analysis. Additionally, we introduce BigDocs-Bench, a benchmark suite with 10 novel tasks where we carefully create datasets that reflect real-world use cases involving reasoning over Graphical User Interfaces (GUI) and code generation from images. Our experiments show that training with BigDocs-Bench improves average performance up to 25.8% over closed-source GPT-4o in document reasoning and structured output tasks such as Screenshot2HTML or Image2Latex generation. Finally, human evaluations showed a preference for outputs from BigDocs-trained models over GPT-4o. This suggests that BigDocs can help both academics and the open-source community utilize and improve AI tools to enhance multimodal capabilities and document reasoning.

Workshop on Responsibly Building the Next Generation of Multimodal Foundational Models
38th Conference on Neural Information Processing Systems (NeurIPS 2024).

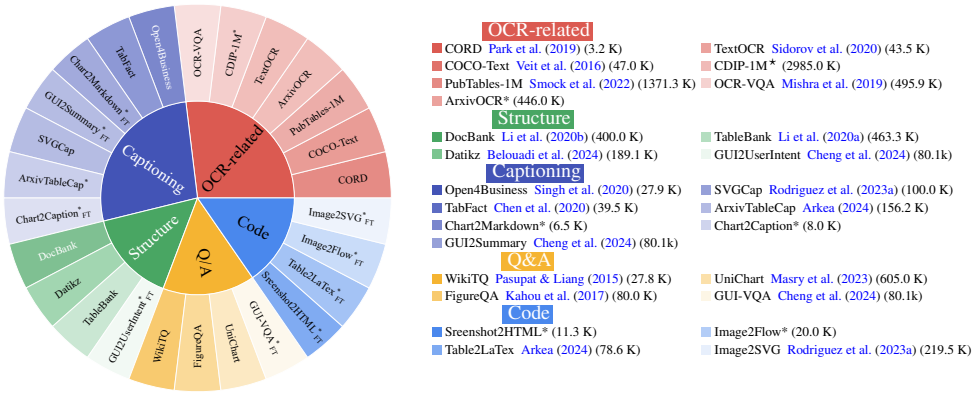


Figure 1: ***BigDocs: A Large-Scale Structured Continual Pretraining and Finetuning Dataset.*** *The inner circle represents the distribution of BigDocs, detailing the categories. The outer circle displays the specific datasets compiled to form 7 million image-text pairs. Datasets with * denotes our contribution.*

1 Introduction

Visually-Rich Document (VRD) data containing text and structured elements (such as charts, infographics, diagrams, sketches, tables, etc.) are essential cues for users to efficiently understand complex information holistically. To facilitate this understanding, foundation models for VRDs must process these structured documents and subsequently extract key insights, identify patterns, and generate concise summaries and responses from user requests with reasoning (Landeghem et al., 2023; Zhu et al., 2022). Recent advances in multimodal AI (Yang et al., 2023; Team et al., 2024b) have demonstrated impressive capabilities, including generating functional web pages (Zheng et al., 2024), automating document understanding workflows (Wang et al., 2023b), and extracting detailed information from documents to produce comprehensive reports (Borchmann, 2024). However, the datasets used to train these models remain closed-source, with undisclosed details and restrictive licensing, which hampers their broader adoption in research and the advancement of open-source model development. In contrast, current open-source models and datasets (Chen et al., 2023; Liu et al., 2023b, 2024) primarily focus on basic document understanding tasks, such as optical character recognition (OCR), e.g., DocStruct4M (Hu et al., 2024), or basic question-answering and mathematical problems, e.g., Cambrian-7M (Tong et al., 2024). These efforts do not sufficiently address the complexity of processing intricate visual documents or generating long-structured outputs, such as JSON and HTML, which are valuable in real-world applications.

In this work, we present BigDocs, a large-scale and open *dataset*, *benchmark suite*, and *models* specifically designed for user-facing document-related tasks. BigDocs aims to bridge existing gaps by enabling open-source models to meet the rising demand for sophisticated document understanding technologies. Comprising 7.5 million image-text pairs, BigDocs is carefully curated to support three core areas: (1) **Document Information Extraction**, which includes enhanced OCR for diverse document types, named entity recognition, layout analysis, and table detection; (2) **Document Understanding**, covering semantic comprehension tasks such as document classification, question answering, and analysis of diagrams; and (3) **Document Creation and Manipulation**, which involves converting visual data into structured formats like HTML, LaTeX, and JSON.

Our survey of 133 existing datasets revealed that 80% of them (i.e., around 100 datasets) have either non-permissive licenses (Jaume et al., 2019; Štěpán Šimsa et al., 2023) or no clear licensing information (Chaudhry et al., 2019; kleister Charity, 2021), creating barriers to reuse and transparency. In response, BigDocs prioritize datasets with permissive licenses (e.g., CC-BY-4.0, MIT) and document-related information, ultimately retaining 16 fully accessible datasets. To further support accessibility, we developed the BigDocs Toolkit, which offers modular tools for data preprocessing, filtering, and consolidation. Additionally, we introduced a unified metadata framework to enhance dataset traceability (e.g., properties, sources, licenses), including detailed documentation of transformations applied by us to the original data. We also conduct a data contamination analysis on downstream tasks data, showing that BigDocs-7.5M has lower contamination rates than previous datasets.

To further advance document intelligence, BigDocs-Bench offers 10 novel downstream tasks, each with four splits: train, validation, test, and hidden test (with 329k training samples, 11k validation

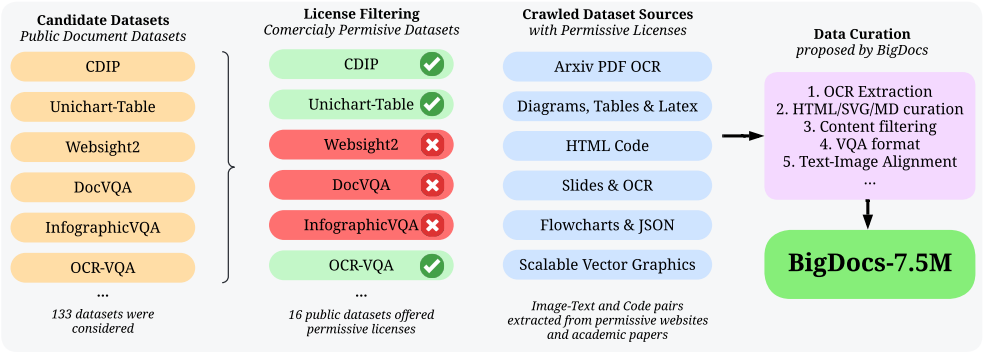


Figure 2: ***BigDocs-7.5M Dataset Curation.*** *The figure illustrates the extraction, filtering, and curation process of BigDocs-7.5M, which emphasizes maintaining permissive licensing. To build BigDocs-7.5M, we first gather publicly-available vision-language datasets, particularly those centered on document analysis, and apply a rigorous filtering process. We then augment these datasets with our own crawled data. Finally, we standardize all samples and tasks into a unified format to produce BigDocs-7.5M.*

samples, 10k testing samples). These tasks focus on structured output generation, including code formats such as HTML, LaTeX, SVG, and Markdown. Our experimental results demonstrate that models trained on the BigDocs suite outperform those trained on existing datasets like DocStruct4M (Hu et al., 2024) on standard document benchmarks. Additionally, automatic and human evaluations on the novel tasks introduced in BigDocs-Bench highlight the advanced capabilities of these models in generating long-format, structured outputs. User evaluations reveal a preference for our models' outputs 88% of the time over Phi3.5 Instruct and 63% over GPT-4.

Built with a commitment to accountability, responsibility, and transparency (ART) (Bommasani et al., 2023; Vogus & Llansóe, 2021), BigDocs will be open-sourced, including datasets, models, and documentation to foster responsible AI development. In summary, BigDocs contributions include:

1. **BigDocs-7.5M**, a large-scale, license-permissive dataset designed for continual pretraining (further training from a pretrained foundation model checkpoint) and downstream finetuning (e.g., to follow instructions or task formats) of multimodal models on document-related tasks. It includes traceable *metadata* and curated licensing drawing from document-rich *multiple* data sources, ensuring full public accessibility.
2. **BigDocs-Bench**, a set of 10 new benchmarks, including test datasets as well as corresponding innovative evaluation metrics for multimodal models to generate long-structured code outputs from images, including formats such as HTML, LaTeX, Markdown, and SVG.
3. **BigDocs Toolkit**, unified tools supporting open-source efforts. These tools allow efficient data curation, filtering, formatting, and preparation for training models generating structured outputs.
4. **BigDocs Models**: We conduct extensive experiments using four state-of-the-art public models, demonstrating the advantages of training with BigDocs over alternative datasets and enabling the models to learn novel tasks through our dataset suite.

2 BigDocs-7.5M

BigDocs-7.5M is a large-scale, license-permissive, and carefully curated dataset for visual document understanding designed to train foundational models across various document types and tasks. It consolidates public datasets and newly crawled data with permissive licenses by preprocessing, cleaning, and filtering them into a unified collection of 7.5 million image-text pairs. All curated datasets and related artifacts will be openly released to foster community collaboration (Bender & Friedman, 2018). The curation process is illustrated in Figure 2 and detailed below.

2.1 Dataset Curation Process

Existing Dataset Acquisition. The authors, along with domain experts and researchers, guided the collection strategy, assessing dataset relevance, quality, and diversity. We gathered 133 public vision-language datasets by searching academic repositories, open data platforms, and research

papers. The collection focused on tasks like image captioning (Chen et al., 2015; Sidorov et al., 2020), OCR (Park et al., 2019; Smock et al., 2022), visual question answering (Mishra et al., 2019; Mathew et al., 2021b), scene-text recognition (Veit et al., 2016; Singh et al., 2021), and document layout analysis (Li et al., 2020a,b), resulting in a diverse multimodal dataset repository.

Datasheets for Datasets. During data acquisition, we compiled detailed datasheets for each dataset, capturing metadata such as ownership, status, size, references, source type, annotations, licensing, and specific observations. We filtered datasets based on licensing compatibility and relevance to our document-related tasks, then extended the datasheets of selected datasets for better categorization. This extension organized datasets by attributes such as medium type (e.g., digital, scanned), document type (e.g., articles, infographics), sourcing method, text type (e.g., computer-generated, handwritten), structure, language, timeframe, and licensing. For licensing, we documented both the image licenses and annotation licenses separately, as these often differed and impacted the overall permissiveness and usability of each dataset. This structured approach aligned datasets with specific use cases (e.g., OCR, structured parsing) and grouped them for pretraining, finetuning, and evaluation, ensuring effective integration into our visual document understanding pipeline.

License Filtering. A key criterion for dataset selection was ensuring permissive licenses (e.g., CC-BY, MIT, Apache 2, CC0) for both images and annotations, suitable for open access and commercial use (more details on various of licenses in Appendix A.8). Datasets with non-permissive licenses, like DVQA (Kafle et al., 2018) (CC-BY-NC 4.0) or DocILE (Štěpán Šimsa et al., 2023) (non-commercial use), or with no license information, like DeepForm (Svetlichnaya, 2020), were excluded. Ultimately, we prioritized permissive licenses for both text and images, resulting in 20% being kept, while 7.5% moderately restrictive and 72.5% non-permissive were discarded. Some included datasets still have images under less clear terms, such as "Fair Use" (e.g., OCR-VQA), documented in the metadata.

2.2 BigDocs Toolkit: Data Preprocessing, Filtering, and Consolidation

The BigDocs Toolkit provides modular tools for preprocessing, filtering, contamination management, metadata management, and dataset loading. These components work in unison to streamline the integration of large-scale document datasets, ensuring quality and ease for efficient model training.

Datamaker Module. The BigDocs Toolkit offers a modular framework for dataset curation, focusing on standardization, quality control, and metadata management. Its core *DataMaker* class acts as a template for handlers that extract annotations and convert raw data into a standardized format. A universal function processes tasks like OCR, VQA, and code generation, ensuring consistency. Bounding boxes are standardized, and corrupted samples are filtered. The Toolkit also generates metadata to enhance transparency, covering licensing and processing details (see Appendix A.10).

Unified Metadata Framework. We propose a unified metadata framework for BigDocs to ensure transparency and traceability. This framework thoroughly examines each raw data source, extracts fine-grained license information, and documents transformations applied to the data (e.g., different sources may have distinct licenses). Each data sample includes a metadata attribute detailing its properties, licenses, sources, and transformations (see Appendix A.10 for an example in Figure 13 and structure details). To our knowledge, this is the first systematic approach to track metadata for visually rich documents, advancing transparency in multimodal dataset curation.

Assessing Contamination. The presence of downstream evaluation data in training datasets can significantly affect the accurate measurement of a model's effectiveness (Magar & Schwartz, 2022). BigDocs-7.5M contains samples from the training split of TabFact, WTQ, and TextVQA. However, any evaluation dataset may *a priori* overlap with training datasets through less direct dependencies. We favor a transparency strategy: Appendix A.4 reports our approach and estimation of this overlap.

2.2.1 The resulting BigDocs-7.5M

BigDocs-7.5M consists of 7.5M image-text pairs for training (4M unique images), 500k for validation (234k unique images), and 470k for testing (261k unique images). These data points aggregate four document-related tasks – OCR, structured parsing, captioning, and question-answering – enabling models to handle diverse document use cases (see Figure 1 and Appendix A and A.2 for more details). Low-quality data is filtered out, and metadata details our best-effort licensing compliance, making the dataset suitable for training foundational models, even in commercial applications.

Table 1: ***Statistics of the ten downstream tasks in BigDocs-Bench.*** *GPT2 tokenizer is used to produce token numbers of both queries and annotations (if any), where the format is (avg ± std).*

Downstream Task	# Training	# Validation	# Public Test	# Hidden Test	# Text Tokens
Screenshot2HTML	9,338	1,000	500	500	32,700 ± 53,105
Table2LaTeX	77,669	1,000	500	500	438 ± 540
Image2SVG	198,000	2,000	748	748	2,871 ± 1,728
Image2Flow$_{\text{(GraphViz)}}$	8,000	1,000	500	500	418 ± 124
Image2Flow $_{\text{(JSON)}}$	8,000	1,000	500	500	1,771 ± 601
Chart2Markdown	4,516	1,000	500	500	1,559 ± 4,442
Chart2Caption	5,412	1,300	650	650	94 ± 49
GUI2UserIntent	79,000	1,000	500	500	28 ± 4
GUI2Summary	79,000	1,000	500	500	132 ± 25
GUI-VQA	78,991	1,000	500	500	35 ± 24

Figure 3: ***8 of the new tasks introduced in BigDocs-Bench.*** *These tasks share a focus on understanding the underlying structure of visually rich documents, with many also requiring generating lengthy outputs, such as SVG and HTML code. More tasks are shown in Figure 9 and 10.*

3 Building BigDocs-Bench

In the previous section, we introduced BigDocs-7.5M, a unified and permissive dataset for training models on document understanding tasks. Building on this, we present BigDocs-Bench, a benchmark suite for evaluating downstream tasks that transform visual inputs into structured outputs, such as GUI2UserIntent (fine-grained reasoning), Image2Flow (structured output), Chart2Caption (understanding), and Screenshot2HTML (creative generation). BigDocs-Bench includes ten specialized tasks, with 329k training samples, 11k validation samples, 10k testing samples, and an additional hidden test set. Refer to Table 1 for task details and Figure 3 for examples.

3.1 BigDocs-Bench Tasks Suite

Screenshot2HTML: We introduce a benchmark for Screenshot2HTML conversion, with 10,838 real-world website screenshots paired with HTML code (Appendix A.5.2). Curated from diverse, text-heavy websites in the FineWeb corpus (Penedo et al., 2024), it contrasts with synthetic GPT-generated sites from prior work (Laurençon et al., 2024b). Using `Playwright`, we retrieved, rendered, and filtered sites for accessibility, content, and licensing. External assets (e.g., CSS, fonts) were inlined, JavaScript removed, and images replaced to focus on structure. Screenshot2HTML evaluates the accuracy of HTML generated from webpage screenshots, emphasizing layout fidelity and semantic correctness.

Table2LaTeX: We propose a benchmark for Table2LaTeX conversion, consisting of 79,669 table images paired with original LaTeX code and captions (details in Appendix A.5.3). The dataset was curated by crawling arXiv papers with permissible licenses and extracting tables from PDFs and TeX source files.* Instead of relying on imperfect PDF detection, tables were rendered from the LaTeX `.tex` code to ensure accurate visuals. For instance, an image of a table is paired with its corresponding LaTeX snippet code and caption. This benchmark supports precise table extraction and LaTeX generation evaluation from academic documents.

Image2SVG: We present a benchmark for Image2SVG conversion, curated from the existing SVG-Stack collection by StarVector (Rodriguez et al., 2023a) (details in Appendix A.5.4). The dataset includes 200k raster images paired with SVG code (e.g., flowchart image paired with SVG

*https://arxiv.org/

code replicating it) or descriptive text. We filtered for complex designs, using image entropy to exclude simple graphics, and ranked image-text pairs by CLIP Score (Hessel et al., 2022; Radford et al., 2019). This benchmark evaluates models on precise vector image reconstruction and scalability.

Image2Flow: We introduce two benchmarks, Image2Flow$_{\text{(GraphViz)}}$ and Image2Flow$_{\text{(JSON)}}$, mapping flowchart images to JSON or GraphViz code (Appendix A.5.1). The dataset includes 10,000 flowchart samples with GraphViz files generated using LLaMA 3.1 (Dubey et al., 2024) and JSON files detailing nodes and connections. Random colors and styles were applied for diversity. Unlike FlowchartQA (Tannert et al., 2023), which focuses on QA over preprocessed flowcharts, this benchmark tests models' ability to extract structure from raw images.

Chart2Markdown: This dataset assesses the models' capabilities in extracting data values from chart images. Formally, the model is given a chart image and asked to produce a data table of the underlying data table in markdown format. To create this dataset, we crawled recent chart images from the Statista website[†], focusing on charts from 2023 and 2024 that were not used in prior datasets like UniChart (Masry et al., 2023) and ChartQA (Masry et al., 2022). This ensures that the dataset reflects the most up-to-date facts and trends and overlaps less with existing datasets and benchmarks. We collected 6,516 chart images with their data tables and human-written summaries.

Chart2Caption: We introduce a benchmark for Chart2Caption conversion, aiming to generate textual summaries from chart images. The dataset includes 6,516 samples, with charts sourced from Kaggle by running public analytics notebooks and extracting charts and code. Summaries were generated using multimodal InterVL2-26B model (Chen et al., 2023, 2024) based on a custom prompt (see Appendix A.5.5) and augmented with human-written summaries from the Chart2Markdown task. This benchmark evaluates models' abilities to interpret and summarize visual data representations.

GUI2UserIntent: This benchmark interprets user intent from GUI interactions, identifying elements linked to clicked bounding boxes (details in Appendix A.5.7). The dataset, repurposed from SeeClick (Cheng et al., 2024), includes 80,000 website screenshots with bounding box coordinates and corresponding user intents sourced from Common Crawl to capture user interactions effectively.

GUI2Summary: The GUI2Summary task generates descriptions of website screenshots, focusing on web layouts. We synthesized 80,000 summaries (under 100 words each) using InternVL2-8B (Chen et al., 2023) in a zero-shot setting (details in Appendix A.5.8). Each summary provides an overview of the main content, referencing key visual elements, layout, and color schemes.

GUI-VQA: The GUI-VQA task answers questions about website screenshots, focusing on content and elements.We generated 80k QA pairs using sentences from GUI-to-Summary and prompting LLaMA 3.1-8b (Dubey et al., 2024) in a zero-shot setting (details in Appendix A.5.9).

3.2 Filtering and Quality

Filtering with BigDocs Toolkit. To ensure a high-quality, open-access dataset, we employed an NSFW detector alongside filtering tools to eliminate harmful content, corrupted images, misaligned annotations, and personally identifiable information (PII). The BigDocs Toolkit also facilitates distributed web crawling while adhering to ART principles, featuring robust filtering classes to curate safe URLs, remove NSFW content, and exclude PII. This multi-layered filtering process yields a clean, reliable dataset suitable for advanced document tasks. Finaly, the test set underwent *manual human verification*, with each sample reviewed to ensure overall quality, consistency, and accuracy.

4 Training Multimodal Models on BigDocs

We trained several state-of-the-art multimodal models of varying sizes and architectures on BigDocs to assess its effectiveness for document-based continual pretraining and downstream finetuning. For comparison, we also trained the models on DocStruct4M, the closest alternative in terms of scale and document-oriented tasks. Additionally, we experiment on the training set from BigDocs-Bench that requires generating long structure outputs, e.g., valid code outputs.

We follow two stages of training: continual pretraining (CPT) and downstream finetuning (FT). The CPT stage involves training on large domain-specific datasets, such as BigDocs and DocStruct4M,

[†] https://www.statista.com/

learning general tasks like OCR, layout understanding, and captioning. For FT smaller datasets, such as DocDownStream and BigDocs-Bench, to focus on specific tasks like question answering or generating HTML from images. A previous stage of pretraining (PT) is typically performed for general multimodal alignment. In our framework, we do not perform this stage and rely on publicly available checkpoints. While pretraining (PT) is typically performed for general multimodal alignment, we rely on public checkpoints instead. In CPT, we train the image encoder and connector to align image features with the LLM. For FT, both the connector and LLM remain unfrozen. See Figure 11 in Appendix A.6 for more details.

4.1 Experimental Setup

Baseline Models. We selected DocOwl1.5-8B (Hu et al., 2024), Qwen2VL-2B (Bai et al., 2023a), Phi-3.5-Vision-4B (Abdin et al., 2024), and LLaVa-NeXT-7B (Li et al., 2024) for our training experiments. These models were chosen due to their focus on document-related tasks (DocOwl1.5), their openness regarding checkpoints (Qwen2VL, DocOwl1.5), and their state-of-the-art performance and task generalization capabilities (LLaVa-NeXT, Phi-3.5).

Training Details. We conduct all experiments using 8 nodes of 8 H100 GPUs, using Fully Sharded Data Parallel (FSDP) for distributed training. All experiments use a batch size of 256 and a learning rate of 2e-5, with AdamW as the optimizer. More training details are provided in Appendix A.6.

Evaluation Benchmarks & Metrics. In addition to the newly introduced BigDocs-Bench, we assess performance on well-known document-oriented benchmarks, termed as **General Document Benchmarks**. We select these benchmarks for their relevance and diverse range of document tasks. DocVQA (Mathew et al., 2021b), InfoVQA (Mathew et al., 2021a), DeepForm (Svetlichnaya, 2020), KLC (Stanisławek et al., 2021) (Kleister Benchmark for Key information extraction), WTQ (Pasupat & Liang, 2015) (Wikipedia Tables), TabFact (Chen et al., 2020), ChartQA (Masry et al., 2022), TextVQA (Singh et al., 2019), MMMU (Yue et al., 2024), DUDE (Landeghem et al., 2023), SlideVQA (Tanaka et al., 2023), and TableVQA (Kim et al., 2024). We utilize (and extended) VLM Eval Kit (Duan et al., 2024).

In **BigDocs-Bench**, we employ the following evaluation methods based on each task's characteristics. For Screenshot2HTML, inspired by related works in this domain (Reis et al., 2004), we compute Tree Edit Distance (TE Dist.) between the Document Object Model (DOM) of ground truth and generation. For Table2LaTeX, we report TeXBLEU (Jung et al., 2024). For Image2SVG, we compare the cosine similarity between the DINOv2 (Oquab et al., 2023) representations of the ground-truth and generated SVG images (DINOScore). For the Image2Flow tasks, we propose the Length-Shape Triplet F1 (**LST F1**) score between ground-truth and generated flowcharts' edge sets. More specifically, each edge is represented as a (s, e, d) triplet, where e is the edge label, and s and t are the source and destination nodes' labels concatenated with their shapes, respectively. For Chart2Markdown, we adopt the RMSF1 metric for markdown tables (Liu et al., 2022; Masry et al., 2023, 2024). For summarization and VQA tasks, we report Rouge-L F1 score (Lin, 2004). For more details about the evaluation process, please refer to Appendix A.7.

Setup. For our CPT on DocOwl1.5-8B and Qwen2VL-2B, we used base weights, while on Phi-3.5-Vision-4B and Llava-NeXT-7B, we initialized from their instruction-tuned versions, since their base weights are not publicly available. We first trained each model on a CPT corpus, either DocStruct4M (Hu et al., 2024) or BigDocs-7.5M, for one epoch. Following this, we performed further alignment (finetuning) using DocDownStream to enhance the model's ability to follow instructions. For each selected model, we also evaluated the author-provided base model and the instruction-tuned version (separate from the base checkpoint, if available) as baselines and reported the performance on general document benchmarks.

We also provide a comprehensive evaluation on the proposed BigDocs-Bench. We conducted an off-the-shelf performance analysis on BigDocs-7.5M using models such as GPT4 (Achiam et al., 2023), Claude (Anthropic, 2024), Gemini Pro (Team et al., 2024a) and Qwen2VL-72B (Wang et al., 2024a) and Idefics2 (Laurençon et al., 2024a). In addition, we also evaluated the previously selected models on their instruction versions, including DocOwl1.5-8B, Qwen2VL-2B, Phi-3.5-Vision-4B, and Llava-NeXT-7B. To incorporate the new capabilities introduced in BigDocs-Bench, we further finetuned these models after BigDocs CPT using the training set from BigDocs-Bench (see Table 3). For each model, we only evaluate the instruction-tuned version (where available) as baselines, reporting their respective performance.

Table 2: ***General Document Benchmarks***. *Models trained on {BigDocs-7.5M+DocDownstream} perform competitively across multimodal document benchmarks. We compare them to base checkpoints, instruction-tuned models, and those trained on {DocStruct4M+DocDownstream}. BigDocs models show consistent performance.*

Model	DocVQA VAL	InfoVQA VAL	DeepForm TEST	KLC TEST	WTQ TEST	TabFact TEST	ChartQA TEST	TextVQA VAL	MMMU VAL	DudeMini TEST	SlideVQA-M TEST	TableVQA TEST	Avg. Score
DocOwl1.5-8B (instruct)	80.73	49.94	68.84	37.99	38.87	79.67	68.56	68.91	33.67	34.64	31.62	52.60	53.84
DocOwl1.5-8B (base)	2.07	1.84	0.00	0.00	0.00	0.00	0.00	0.00	24.44	19.07	3.30	13.63	5.36
DocOwl1.5-8B (base) + DocStruct4M	75.99	46.88	62.77	35.21	32.86	71.56	**68.36**	65.08	**33.67**	29.00	27.03	46.27	49.56
DocOwl1.5-8B (base) + BigDocs (Ours)	**78.70**	**47.62**	**64.39**	**36.93**	**35.69**	**72.65**	65.80	**67.30**	32.33	**32.55**	**29.60**	**49.03**	**51.05**
Qwen2-VL-2B (instruct)	89.16	64.11	32.38	25.18	38.20	57.21	73.40	79.90	42.00	45.23	46.50	43.07	53.03
Qwen2-VL-2B (base)	7.26	0.78	0.00	0.00	0.00	0.00	0.00	1.14	34.89	28.43	14.55	0.00	7.25
Qwen2-VL-2B (base) + DocStruct4M	**59.53**	**32.00**	**53.98**	**36.38**	28.48	64.24	54.44	55.89	34.89	**28.78**	**22.68**	46.53	43.15
Qwen2-VL-2B (base) + BigDocs (Ours)	57.23	31.88	49.31	34.39	**31.61**	**64.75**	**68.60**	**61.01**	**35.67**	27.19	17.46	**47.53**	**43.89**
Phi3.5-Vision-4B (instruct)	86.00	56.20	10.47	7.49	17.18	30.43	82.16	73.12	46.00	37.20	30.93	70.70	45.66
Phi3.5-Vision-4B + DocStruct4M	86.76	68.90	70.12	**37.83**	**51.30**	**82.12**	79.76	68.60	44.11	35.52	31.90	**69.17**	60.51
Phi3.5-Vision-4B + BigDocs (Ours)	**87.05**	**70.05**	**70.97**	37.45	51.21	81.24	**81.56**	**68.72**	**45.00**	**36.15**	**32.47**	67.77	**60.80**
LLaVA-NeXT-7B (instruct)	63.51	30.90	1.30	5.35	20.06	52.83	52.12	65.10	38.89	17.94	7.46	32.87	32.36
LLaVA-NeXT-7B + DocStruct4M	**60.95**	**26.14**	39.78	28.34	25.90	67.72	**61.20**	**52.25**	**25.78**	21.70	15.33	27.03	37.68
LLaVA-NeXT-7B + BigDocs (Ours)	57.13	24.47	**46.38**	**31.09**	**27.06**	**72.58**	54.72	49.06	17.78	**22.88**	**16.07**	**33.13**	**37.70**

Table 3: ***Comparison of model performance in BigDocs-Bench.*** *BigDocs models trained on {BigDocs-7.5M+DocDownstream+BigDocs-Bench train-split}, which combine CPT and FT, outperform all baselines in tasks requiring long-format code generation, particularly in flow generation, GUI reasoning, and image-to-LaTeX generation, surpassing even state-of-the-art closed models.*

Model	Chart2MD RMSF1	Chart2Cap. ROUGEL FT	Image2Flow (GraphViz) LST F1	Image2Flow (JSON) LST F1	GUI2Sum. ROUGEL FT	GUI2Intent ROUGE-L F1	Image2SVG DINO Score	Screenshot2HTML DOM MEDIC	Table2Latex TeXBleu	GUI-VQA ROUGE-L F1	Avg. Score
Open Models											
DocOwl-1.5-8B	0.05	10.08	0.00	0.00	10.54	13.88	2.41	3.44	75.07	23.81	15.70
Qwen2-VL-2B	41.17	15.19	0.00	0.00	23.98	17.70	23.18	6.46	74.83	19.17	20.00
Phi3.5-V-4B	60.64	16.29	0.00	0.00	27.80	10.81	25.35	4.25	74.22	31.72	20.75
LLaVA-NeXT-7B	22.00	13.24	0.00	0.00	21.99	2.24	20.53	4.09	73.37	8.63	15.93
Idefics2-8B	0.00	7.68	0.00	0.00	8.75	5.06	22.58	3.50	74.24	16.92	15.09
Closed Models											
GPT-4o 20240806	14.94	21.46	6.95	0.00	27.12	9.40	60.34	10.33	74.65	18.25	24.66
Claude-3.5 Sonnet	0.77	13.26	1.66	0.00	26.45	4.87	25.46	9.70	74.44	8.61	18.31
GeminiPro-1.5	8.09	19.13	3.94	0.20	25.54	9.89	15.21	7.43	75.22	15.38	19.23
Qwen2-VL-72B	13.22	12.45	3.65	0.70	18.80	19.80	54.43	10.03	74.51	20.07	23.07
BigDocs Models (ours)											
DocOwl-1.5-8B + BigDocs	74.43	24.15	42.16	48.54	45.55	**89.15**	33.66	3.64	81.28	43.46	44.35
Qwen2-VL-2B + BigDocs	72.25	26.05	41.61	52.11	42.59	71.65	-	9.20	78.54	33.97	42.80
LLaVA-NeXT-7B+ BigDocs	72.78	24.05	59.66	71.49	46.14	79.55	**60.63**	10.40	80.79	40.67	50.43
Phi3.5-v-4B + BigDocs	**84.01**	**28.89**	**63.07**	**71.86**	**47.32**	86.91	34.65	**12.05**	**81.94**	**44.81**	**50.46**

4.2 Quantitative Results

Results on Existing Document Downstream Tasks. Table 2 presents the models performance across general document benchmarks. Base models perform poorly, mainly due to their inability to follow user instructions, like answering questions. Phi3.5 Vision, originally optimized for reasoning tasks, achieves an average score of 60.80% when finetuned on BigDocs, enhancing its ability to handle complex document-based tasks. For Qwen2-VL, an interesting pattern emerges: while the instruction-tuned version excels on tasks reported in its technical report (e.g., DocVQA, InfoVQA, ChartQA, MMMU), the BigDocs-trained model surpasses it on new tasks like TabFact, DeepForm, KLC, and TableVQA, suggesting that Qwen2-VL's instruction-tuning may rely on complex prompt engineering and task-specific optimizations. *We find that performing additional continual pretraining and finetuning on instruction-tuned models does not significantly degrade performance*. Moreover, **we observe substantial improvements on previously underperforming tasks**. As shown in Table 2, Phi3.5 and LLaVA-Next show marked gains on DeepForm, KLC, WTQ, and SlideVQA. However, LLaVA's performance declined on MMMU, indicating the need for more multiple-choice question data. This reinforces our argument that transparency in training datasets is essential for proper evaluation. Finally, across all models, BigDocs training yields higher average scores compared to training on DocStruct4M, even with lower contamination rates on most benchmarks, as we highlighted in Section 2. These findings indicate that **BigDocs supports better generalization and robustness without complex task-specific optimizations while also being license-permissive.**

Results on BigDocs-Bench Tasks. Table 3 shows results for our proposed downstream tasks, evaluating models' ability to generate lengthy structured and valid code outputs, and reasoning from GUIs. Overall, BigDocs models consistently outperform both open and closed models on most

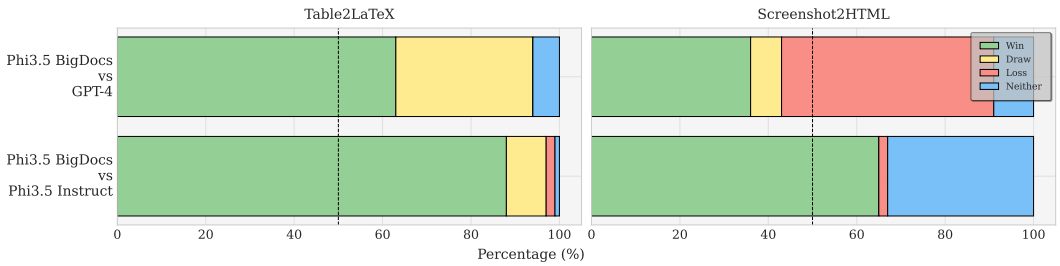


Figure 4: ***Human evaluation results*** *comparing Phi3.5 BigDocs-Bench against Phi3.5 Instruct and GPT-4o on two tasks: Table2LaTex (Left) and Screenshot2HTML (Right).*

tasks, particularly in Flow and GUI tasks such as GUI2UserIntent, GUI2summary, GUI-VQA, and Image2Flow, revealing areas where existing models fall short. The performance gap is narrower on tasks like Screenshot2HTML, Image2SVG and Chart2Caption, suggesting these tasks have been explored in the literature but not extensively enough. Phi3.5-V4B + BigDocs stands out as the top performer in 8 tasks out of 10, with an average score of 50.46. However, its performance on Image2SVG (25.98 points behind LLaVA-NeXT-7B + BigDocs) indicates less exposure to SVG data in its instruction tuning. We find that the model can generate valid code outputs in different formats, including SVG, HTML, JSON, or Latex, when conditioned on images.

4.3 Human Evaluations and Qualitative Results

We conducted a human evaluation comparing the performance of **Phi-BigDocs**, **Phi-Instruct**, and **GPT-4o** on **Screenshot2HTML** and **Table2LaTeX** tasks. Twenty-eight evaluators participated, providing 1,900 annotations. For Table2LaTeX, evaluators assessed if the LaTeX table matched the input table. For Screenshot2HTML, they evaluated the visual similarity between the rendered HTML and the screenshot. Evaluators chose between "Win" (one output was superior), "Neither" (both were poor), or "Both" (outputs were equally good). We collected 1,900 annotations in total. See Appendix A.9 for more details on the evaluation platform.

From human evaluation results in Figure 4, for the Table2LaTeX task, Phi3.5 BigDocs wins 88% of the time against Phi3.5 Instruct and achieves a 63% win rate, with a 31% draw rate, against GPT-4o. These results highlight our model's ability to accurately preserve the table's structure, including lines, borders, and margins, whereas GPT-4o often struggles to maintain consistent formatting despite capturing content accurately. In the Screenshot2HTML task, Phi3.5 BigDocs achieves a 65% win rate against Phi3.5 Instruct and performs competitively against GPT-4o, with a 36% win rate and 7% draw rate, demonstrating its strong capability to reproduce visual elements faithfully.

Qualitative Results. We provide qualitative results in Appendix A.11. Figure 8 presents outputs from experiments with the Phi-3.5-Vision-4B model on BigDocs-Bench for tasks like Chart2Markdown, Table2LaTeX, and Image2SVG. The model delivers visually consistent outputs and generates valid code across formats, adhering to task instructions. Table 6 compares sample outputs between Phi-3.5-Vision models and GPT-4o, highlighting the strong performance of our BigDocs trained version in captioning and VQA tasks.

5 Conclusion

We introduce BigDocs-7.5M, a large-scale, license-permissive dataset for training multimodal models on document and code-related tasks. Along with a comprehensive suite of tools and data analysis, we present BigDocs-Bench, featuring 10 downstream tasks that assess a model's ability to generate long-format code outputs from images. These tasks serve as practical benchmarks for real-world applications. Our experiments show that models trained on BigDocs outperform those trained on existing datasets. Furthermore, training on the BigDocs-Bench train split endows the resulting models with new capabilities and significantly enhances their ability to generate long, structured outputs. All BigDocs artifacts will be freely available under permissive licenses.

Limitations Our work presents a pioneering license-permissive dataset for multimodal document understanding, achieving strong performance across tasks. However, there are limitations: (1) Suboptimal performance on some public benchmarks, indicating a need to refine the data mixture and explore additional sources. (2) Limited context length, as models are trained with a maximum of 8192 tokens, restricting performance on tasks with long, structured outputs like HTML and SVG. (3) Uncertainty in the commercial viability of base models, as their pretraining data lacks transparency.

Ethics Statement Our work is centered around responsible and transparent curation of datasets for multimodal document understanding models. While we have made extensive efforts to filter harmful content from our dataset, we cannot fully guarantee that the models will not generate offensive language. Additionally, we have taken significant steps to remove personally identifiable information (PII) from the compiled datasets to protect user privacy. However, we cannot ensure that the generated code will be free of malicious snippets, and developers are encouraged to implement protection protocols to safeguard against potential risks. Finally, all human evaluation studies were conducted by collaborator researchers, and no PII was collected during this process.

Reproducibility Statement We are committed to ensuring the reproducibility of our work by providing all necessary details and resources. All artifacts, including code, datasets, model weights, data sheets, and metadata, will be publicly released. Furthermore, we have fully documented all hyperparameters, experimental setups, and evaluation metrics to allow for accurate replication of our results. For human evaluation, we provide clear instructions and describe the environment used for comparison to ensure transparency and consistency.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024.

Credit Mutuel Arkea. Aftdb dataset, 2024. URL https://huggingface.co/datasets/cmarkea/aftdb. Hugging Face Dataset.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv: 2309.16609*, 2023a.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023b.

Jonas Belouadi, Anne Lauscher, and Steffen Eger. Automatikz: Text-guided synthesis of scientific vector graphics with tikz. In *IEEE Conference Computer Vision Pattern Recognition*, 2023.

Jonas Belouadi, Simone Paolo Ponzetto, and Steffen Eger. Detikzify: Synthesizing graphics programs for scientific figures and sketches with tikz, 2024.

Emily M. Bender and Batya Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 2018.

Rishi Bommasani, Kevin Klyman, Shayne Longpre, Sayash Kapoor, Nestor Maslej, Betty Xiong, Daniel Zhang, and Percy Liang. The foundation model transparency index. *arXiv preprint arXiv:2310.12941*, 2023.

Rishi Bommasani, Kevin Klyman, Sayash Kapoor, Shayne Longpre, Betty Xiong, Nestor Maslej, and Percy Liang. The foundation model transparency index v1.1. *arXiv preprint arXiv:2310.12941*, 2024.

Łukasz Borchmann. Notes on applicability of gpt-4 to document understanding. *arXiv preprint arXiv:2405.18433*, 2024.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *IEEE Conference Computer Vision Pattern Recognition*, 2021.

Ritwick Chaudhry, Sumit Shekhar, Utkarsh Gupta, Pranav Maneriker, Prann Bansal, and Ajay Joshi. Leaf-qa: Locate, encode & attend for figure question answering. In *Winter Conference on Applications of Computer Vision*, 2019.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference Learning Representations*, 2020.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference Learning Representations*, 2024.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, Dahua Lin, and Kai Chen. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, 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, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Wikimedia Foundation. Wikimedia downloads, 2024. URL https://dumps.wikimedia.org.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander J. Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alexandros G. Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. In *Adv. Neural Information Processing Systems*, 2023.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92, 2021.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *arXiv preprint arXiv:2403.12895*, 2024.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *International Conference on Document Analysis and RecognitionWorkshop*, 2019.

Kyudan Jung, Nam-Joon Kim, Hyongon Ryu, Sieun Hyeon, Seung jun Lee, and Hyeok jae Lee. Texbleu: Automatic metric for evaluate latex format. *arXiv preprint arXiv: 2409.06639*, 2024.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *IEEE Conference Computer Vision Pattern Recognition*, 2018.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning. In *International Conference Learning Representations*, 2017.

Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. Tablevqa-bench: A visual question answering benchmark on multiple table domains. *arXiv preprint arXiv:2404.19205*, 2024.

kleister Charity. kleister charity dataset, 2021. URL https://github.com/applicaai/kleister-charity.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, et al. The stack: 3 tb of permissively licensed source code. *Transactions Machine Learn Research*, 2022.

Jordy Van Landeghem, Rubén Tito, Łukasz Borchmann, Michał Pietruszka, Paweł Józiak, Rafał Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Ackaert, Ernest Valveny, Matthew Blaschko, Sien Moens, and Tomasz Stanisławek. Document understanding dataset and evaluation (dude). In *International Conference Computer Vision*, 2023.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024a.

Hugo Laurençon, Léo Tronchon, and Victor Sanh. Unlocking the conversion of web screenshots into html code with the websight dataset. *arXiv preprint arXiv:2403.09029*, 2024b.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. Obelics: An open web-scale filtered dataset of interleaved image-text documents, 2023.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv: 2405.02246*, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv: 2408.03326*, 2024.

Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. Tablebank: A benchmark dataset for table detection and recognition. *arXiv preprint arXiv:1903.01949*, 2020a.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. In *International Conference on Computational Linguistics*, 2020b.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Starcoder: may the source be with you! *Transactions Machine Learn Research*, 2023.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, 2004.

Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. Deplot: One-shot visual language reasoning by plot-to-table translation. *arXiv preprint arXiv:2212.10505*, 2022.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Adv. Neural Information Processing Systems*, 2023b.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Tengchao Lv, Yupan Huang, Jingye Chen, Lei Cui, Shuming Ma, Yaoyao Chang, Shaohan Huang, Wenhui Wang, Li Dong, Weiyao Luo, et al. Kosmos-2.5: A multimodal literate model. *arXiv preprint arXiv:2309.11419*, 2023.

Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Dublin, Ireland, May 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.acl-short.18.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. UniChart: A universal vision-language pretrained model for chart comprehension and reasoning. In *Empirical Methods in Natural Language Processing*, 2023.

Ahmed Masry, Mehrad Shahmohammadi, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. ChartInstruct: Instruction tuning for chart comprehension and reasoning. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V Jawahar. Infographicvqa. In *Winter Conference on Applications of Computer Vision*, 2021a.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images. In *Winter Conference on Applications of Computer Vision*, 2021b.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *International Conference on Document Analysis and Recognition*, 2019.

OpenAI. GPT-4V(ision) System Card. https://cdn.openai.com/papers/GPTV_System_Card.pdf, September 2023. Accessed: 2023-11-05.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *Adv. Neural Information Processing Systems*, 2011.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. Cord: A consolidated receipt dataset for post-ocr parsing. In *Adv. Neural Information Processing Systems Workshop*, 2019.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Annual Meeting of the Association for Computational Linguistics*, 2015.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference Machine Learning*, 2021.

N Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

D. C. Reis, P. B. Golgher, A. S. Silva, and A. F. Laender. Automatic web news extraction using tree edit distance. In *International Conference on World Wide Web*, 2004.

Juan A Rodriguez, Shubham Agarwal, Issam H Laradji, Pau Rodriguez, David Vazquez, Christopher Pal, and Marco Pedersoli. Starvector: Generating scalable vector graphics code from images. *arXiv preprint arXiv:2312.11556*, 2023a.

Juan A Rodriguez, David Vazquez, Issam Laradji, Marco Pedersoli, and Pau Rodriguez. Figgen: Text to scientific figure generation. In *International Conference Learning Representations*, 2023b.

Juan A Rodriguez, David Vazquez, Issam Laradji, Marco Pedersoli, and Pau Rodriguez. Ocr-vqgan: Taming text-within-image generation. In *Winter Conference on Applications of Computer Vision*, 2023c.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Annual Meeting of the Association for Computational Linguistics*, 2018.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioningwith reading comprehension. In *European Conference Computer Vision*, 2020.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *IEEE Conference Computer Vision Pattern Recognition*, 2019.

Amanpreet Singh, Niranjan Balasubramanian, and A B. Open4business(o4b): An open access dataset for summarizing business documents. *arXiv preprint arXiv:2011.07636*, 2020.

Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *IEEE Conference Computer Vision Pattern Recognition*, 2021.

Brandon Smock, Rohith Pesala, and Robin Abraham. Pubtables-1m: Towards comprehensive table extraction from unstructured documents. In *IEEE Conference Computer Vision Pattern Recognition*, 2022.

Ian Soboro. Complex document information processing (cdip) dataset. https://doi.org/10.18434/mds2-2531, 2022. Accessed: 2024-06-20.

Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. Kleister: key information extraction datasets involving long documents with complex layouts. In *International Conference on Document Analysis and Recognition*, 2021.

S Svetlichnaya. Deepform: Understand structured documents at scale, 2020.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: A dataset for document visual question answering on multiple images. In *Association for the Advancement of Artificial Intelligence*, 2023.

Simon Tannert, Marcelo G. Feighelstein, Jasmina Bogojeska, Joseph Shtok, Assaf Arbelle, Peter W. J. Staar, Anika Schumann, Jonas Kuhn, and Leonid Karlinsky. FlowchartQA: The first large-scale benchmark for reasoning over flowcharts. In Piush Aggarwal, \"Ozge Ala\ccam, Carina Silberer, Sina Zarrie\ss, and Torsten Zesch (eds.), *Proceedings of the 1st Workshop on Linguistic Insights from and for Multimodal Language Processing*, pp. 34–46, Ingolstadt, Germany, September 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.limo-1.5.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, and Others. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2024a.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, and Others. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024b.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.

Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016.

Caitlin Vogus and Emma Llansóe. Making transparency meaningful: A framework for policymakers. *Center for Democracy and Technology*, 2021.

Štěpán Šimsa, Milan Šulc, Michal Uřičář, Yash Patel, Ahmed Hamdi, Matěj Kocián, Matyáš Skalický, Jiří Matas, Antoine Doucet, Mickaël Coustaty, and Dimosthenis Karatzas. Docile benchmark for document information localization and extraction. In *International Conference on Document Analysis and Recognition*, 2023.

Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. Screen2words: Automatic mobile ui summarization with multimodal learning. In *ACM Symposium on User Interface Software and Technology*, 2021.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. Docllm: A layout-aware generative language model for multimodal document understanding. *arXiv preprint arXiv: 2401.00908*, 2023a.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Lvbench: An extreme long video understanding benchmark, 2024b.

Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. Vrdu: A benchmark for visually-rich document understanding. In *International Conference on Knowledge Discovery & Data Mining*, 2023b.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 2023.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *IEEE Conference Computer Vision Pattern Recognition*, 2024.

Tianyu Zhang, Suyuchen Wang, Lu Li, Ge Zhang, Perouz Taslakian, Sai Rajeswar, Jie Fu, Bang Liu, and Yoshua Bengio. Vcr: Visual caption restoration. *arXiv preprint arXiv: 2406.06462*, 2024.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024.

Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat-Seng Chua. Towards complex document understanding by discrete reasoning. In *ACM International Conference on Multimedia*, 2022.

A Appendix

A.1 Related Work

Multimodal Datasets. General-purpose vision-language datasets like COCO Caption (Chen et al., 2015) and SBUCaption (Ordonez et al., 2011) primarily feature photographic content, lacking visually-rich document (VRD) data (Sharma et al., 2018; Changpinyo et al., 2021). In contrast, our focus is on text-heavy datasets including PDFs, tables, and invoices (Veit et al., 2016; Mishra et al., 2019; Singh et al., 2021; Li et al., 2020b,a; Soboro, 2022), which are crucial for tasks like information extraction and parsing (Masry et al., 2022; Rodriguez et al., 2023c,b). While datasets like DocStruct4M (Hu et al., 2024) and Cambrian7M (Tong et al., 2024) partially address these needs, they often lack permissive licenses or are not open-source. Kosmos-2.5 (Lv et al., 2023) focuses on building a large document dataset; however, the authors did not make it public. BigDocs fills these gaps by providing 7.5M permissively licensed image-text pairs from 16 academic datasets and other open platforms, supporting diverse document understanding tasks.

Responsible Data and Licensing. Enterprise models like GPT-4 (OpenAI, 2023) and Claude (Anthropic, 2024) are often closed-source, lacking transparency in training data (Bommasani et al., 2024). Foundational works (Gebru et al., 2021; Bender & Friedman, 2018) emphasize the importance of open access and transparency in dataset documentation. Previous works promoting open-access such as StarCoder (Li et al., 2023), The Stack (Kocetkov et al., 2022), FineWeb (Penedo et al., 2024), and LLaMA (Dubey et al., 2024), have addressed this by releasing data or models for language models pretraining. Our work builds on these efforts by creating a curated, well-documented resource for open access (Laurençon et al., 2023), addressing licensing complexities ranging from permissive licenses (e.g., Apache 2.0, MIT (Hu et al., 2024; Gadre et al., 2023)) to restrictive ones like CC BY-SA (Foundation, 2024; Zhang et al., 2024) and CC BY-NC-SA (Wang et al., 2024b). Compound datasets, like Cambrian-7M (Tong et al., 2024), face mixed licensing issues, while some datasets, such as DocVQA (Mathew et al., 2021b), have different licenses for images and annotations. In developing BigDocs, we prioritized permissive licensing in data curation and contributed to new datasets that adhere to open-access principles, ensuring transparency at all stages of development.

Multimodal Document Understanding Models. Recent advancements have introduced several general-purpose multimodal models (Liu et al., 2023b,a, 2024; Bai et al., 2023b; Laurençon et al., 2024; Tong et al., 2024). For example, LLaVA (Liu et al., 2023b) integrates a vision encoder with a language model, with later versions enhancing reasoning and OCR capabilities (Liu et al., 2023a, 2024). Qwen2-VL (Bai et al., 2023b) processes images at native resolutions, while Phi 3.5 Vision (Abdin et al., 2024) offers a lightweight model for reasoning tasks. Specialized models for visually-rich documents, like DocOwl1.5 (Hu et al., 2024) and DocLLM (Wang et al., 2023a), have also gained traction, particularly for commercial applications. However, the datasets for training these models are often not publicly available or have restrictive licenses (Hu et al., 2024). Our work, BigDocs, addresses this by consolidating existing permissive datasets to support the development of reproducible, commercially viable document understanding models.

A.2 BigDocs-7.5M Tasks

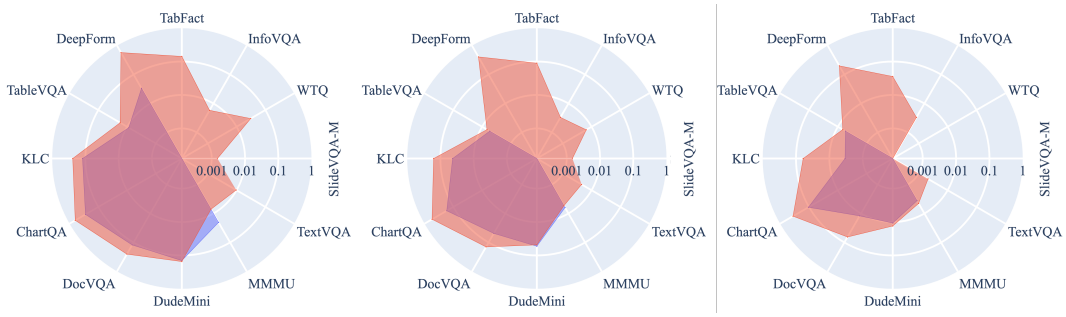
1. **OCR-Related Tasks**: These tasks involve converting images of text (e.g., scanned documents) into machine-readable formats, including transformations such as bounding-box-to-text and text-to-bounding-box (text localization as described in Hu et al. (2024)). Models learn to recognize and map textual information within images.
2. **Structured Parsing and Extraction**: This task focuses on extracting and transforming structured data from documents, like parsing tables, forms, and charts into formats such as JSON or Markdown. It includes handling documents with complex visual layouts and sometimes incorporates bounding box information for individual elements.
3. **Captioning and Summarization**: This task requires models to generate captions or summaries for visual or textual content, such as figures, charts, or document sections. The models provide concise descriptions or overviews, enhancing document comprehension.

4. **Question-Answering (QA)**: QA tasks involve responding to questions posed over structured or visual data (e.g., tables, figures). This includes multi-turn QA, where models address a series of related questions to improve their reasoning and comprehension.

A.3 Datasets included in BigDocs-7.5M

The following datasets are utilized in our work, supporting the four core tasks mentioned in Section 2.2.1. We reference the task supported by each dataset via the index in Section A.2. Note that datasets with * are those fully or partially curated by us (details explained in the description of each of them).

1. **TabFact** (Chen et al., 2020): [Task included: (2), (4)] TabFact is used for question-answering tasks, where models check whether a given statement is supported or refuted by a table. It also involves structured parsing, helping models extract and process structured table data into formats like Markdown.
2. **Open4Business (O4B)** (Singh et al., 2020): [Task included: (2), (3)] O4B is a dataset focused on business-related documents and is processed for structured parsing and extraction, as well as captioning and summarization, allowing models to retrieve key insights from documents and generate summaries or descriptions.
3. **WikiTQ** (Pasupat & Liang, 2015): [Task included: (2), (4)] WikiTableQuestions is used for question-answering tasks over tables, where models answer questions based on table data, and structured parsing for converting table data into Markdown.
4. **CORD** (Park et al., 2019): [Task included: (1), (2)] CORD is a dataset for parsing receipts, used for both OCR-related tasks and structured parsing and extraction, helping models interpret structured financial data from document layouts. This latter task requires extracting entities and providing their text, category, and location as a JSON output.
5. **UniChart** (Masry et al., 2023): [Task included: (2), (4)] UniChart is used for structured parsing and extraction to extract structured information from chart-like tables, converting complex visual layouts into formats like JSON or Markdown. And also for question-answering tasks, where models answer questions related to the chart content.
6. **TextOCR** (Sidorov et al., 2020): [Task included: (1)] TextOCR is processed for OCR-related tasks, enabling models to perform bounding-box-to-text transformations on scene text from images.
7. **COCO-Text** (Veit et al., 2016): [Task included: (1)] COCO-Text is used for OCR-related tasks, helping models extract and recognize text from real-world images with natural scene text.
8. **CDIP-1M*** (Soboro, 2022): [Task included: (1), (2)] CDIP-1M is processed for OCR-related tasks and structured parsing, focusing on extracting text and structure from large-scale scanned document collections. We used an in-house OCR engine to get the text (i.e. annotations) from its 11M documents from the source. Like the text localization task in DocStruct4M Hu et al. (2024), we generate word-, line- and block-level bounding-box-to-text and text-to-bounding-box QA pairs. We subsample zones randomly, and based on OCR confidence, a large fraction of the images are pretty noisy. In addition, for the block level, we generate structured parsing QA pairs where text lines and their location need to be given as JSON by the model.
9. **PubTables-1M** (Smock et al., 2022): [Task included: (1), (2)] PubTables-1M is a large dataset of tables from scientific papers. It is used for **OCR-related tasks** to extract information in tables. It is also processed for **structured parsing and extraction**, allowing models to handle scientific tables and convert them into markdown.
10. **FigureQA** (Kahou et al., 2017): [Task included: (4)] FigureQA is focused on question-answering tasks, where models answer questions based on charts and figures, improving reasoning over visual and tabular data.
11. **DocBank** (Li et al., 2020b): [Task included: (2), (4)] DocBank is utilized for structured parsing and extraction and question-answering tasks, enabling models to interact with scholarly documents and extract structured data from layouts.



(a) *Threshold* 0.96 **(b)** *Threshold* 0.97 **(c)** *Threshold* 0.98

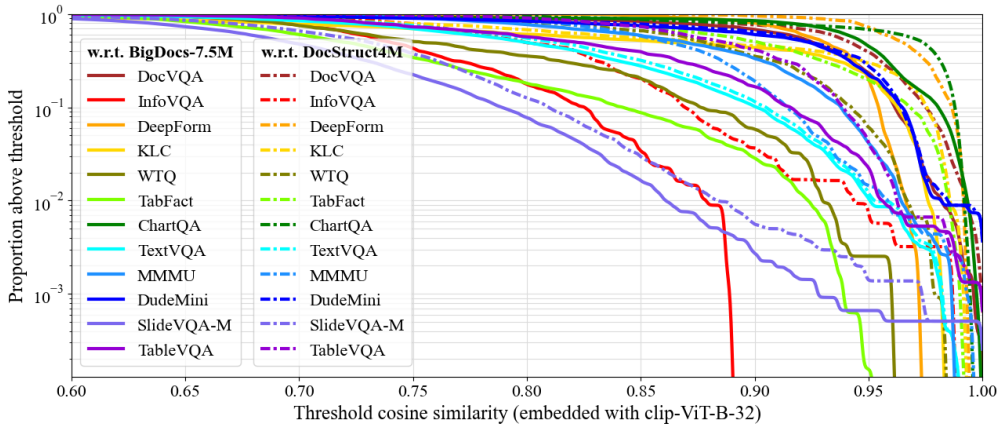
Figure 5: ***Assessing data contamination (smaller is better).*** *The radial axis (log scale) indicates the proportion of images from the evaluation dataset that exhibit similarity to a training sample beyond a given threshold according to CLIP. Human evaluations indicate that most instances captured at a threshold of* 0.98 *are problematic, and most problematic samples are identified at a threshold of* 0.96*. Except for MMMU and DudeMini, BigDocs-7.5M (blue/darkred) is less contaminated compared to DocStruct4M (red).*

12. **TableBank** (Li et al., 2020a): [Task included: (2)] TableBank is a large dataset used for structured parsing and extraction, helping models parse table structures from both Word and LaTeX documents into structured formats.
13. **OCRVQA** (Mishra et al., 2019): [Task included: (1),(4)] OCRVQA focuses on OCR-related tasks and question-answering tasks over OCR-extracted text, where models answer questions based on text and visual data from real-world scenes.
14. **Datikz** (Belouadi et al., 2023): [Task included: (2), (3)] Datikz is processed for captioning and summarization as well as structured parsing, enabling models to describe and interpret complex diagrams and generate structured data from them.
15. **ArxivOCR***: [Task included: (1)] The ArxivOCR dataset contains OCR-scanned academic papers and is used for OCR-related tasks, where models perform bounding-box-to-text transformations, improving accessibility and structure for scholarly articles. This dataset is curated by us. We filter out the papers from Arxiv that have permissive licenses, i.e. CC-BY 4.0 and CC0 in this case. Then we use in-house OCR engines to produce OCR results on the pages from the papers collected.
16. **ArxivTableCap***: [Task included: (3)] The ArxivTable dataset focuses on generating captions for the tables and figures extracted from Arxiv papers, helping models describe the content and context of tables in academic papers. Among the 156.2k samples, 70k of them are from AFTdb Arkea (2024). We perform the filtering to make sure all the selected ones are in papers with permissive licenses, i.e. CC-BY 4.0 and CC0 in this case.
17. **SVGCap Dataset** (Rodriguez et al., 2023a): [Task included: (3)] The SVG Dataset supports captioning and summarization tasks, where models generate descriptive captions for SVG content, summarizing the structure and elements of vector graphics.

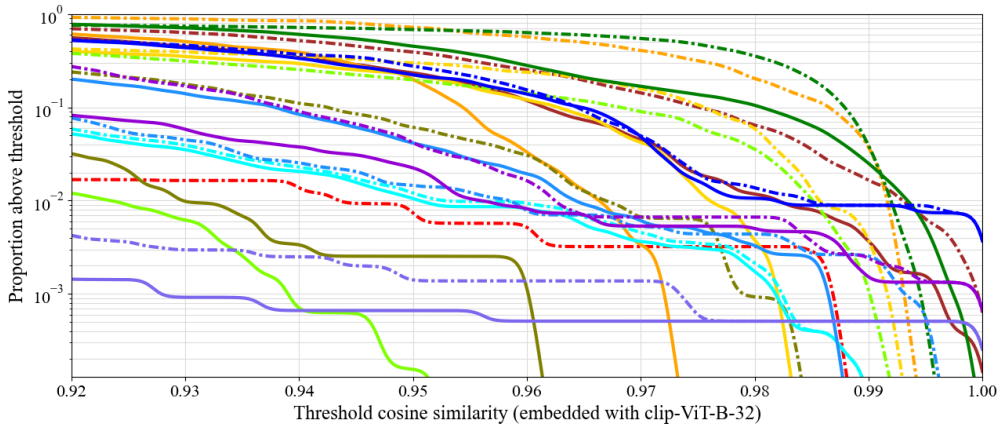
A.4 Assessing Contamination

BigDocs-7.5M's direct dependency on TabFact, WTQ, and TextVQA is restricted to their training splits, and DocStruct4M reports a similar dependency on DocVQA, InfoVQA, DeepForm, KLC, ChartQA, TabFact, WTQ, and TextVQA. However, overlaps between either of these training sets and evaluation datasets may emerge through indirect means (e.g., the same source material was involved).

Our primary "automatic" approach to assess contamination consists of embedding all images from the reference dataset (i.e., BigDocs-7.5M or DocStruct4M) using a pretrained CLIP (Radford et al., 2021), namely `clip-ViT-B-32` from `sentence-transformers` (Reimers, 2019). We similarly embed each image from an evaluation dataset and retrieve the reference image with the highest cosine similarity: the closer this measure is to 1.0, the more likely it is that the evaluation image is part of the reference dataset. Figures 5 and 6, as well as table 4 all report these values.



(a) *InfoVQA, SlideVQA-M, and TabFact overlap very little with BigDocs-7.5M (train).*

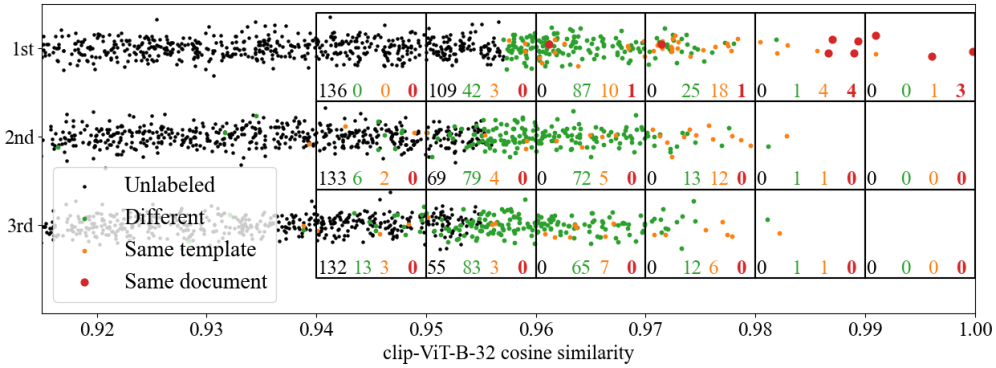


(b) *Zoom on the above (same legend).*

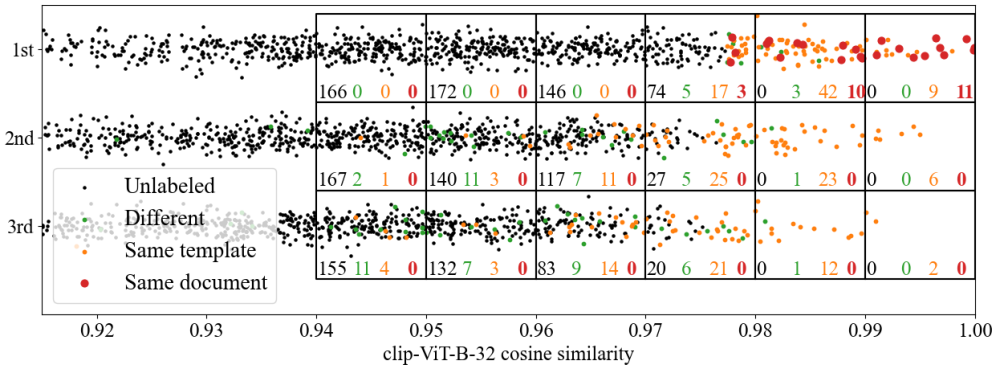
Figure 6: ***Cumulative distribution by cosine similarity.*** *Each curve shows the proportion of samples in an evaluation dataset for which there exists at least one sample in the reference dataset with cosine similarity higher than the specified threshold. Lower values are better; higher cosine similarity thresholds are more pertinent. Based on human evaluations, samples with cosine similarity < 0.96 are unlikely to be problematic, whereas those > 0.98 are most likely problematic. Except for MMMU, and DudeMini, BigDocs-7.5M appears less contaminated than DocStruct4M.*

Table 4: ***Detailed Results on Contamination Experiments, related to Figure 6.*** *Lower values are better; higher cosine similarity thresholds are more pertinent. Except for MMMU and DudeMini, BigDocs-7.5M appears to be less contaminated by these metrics than DocStruct4M. Figure 5's data comes from this table.*

Cosine Similarity Threshold	Reference Dataset	DocVQA	InfoVQA	DeepForm	KLC	WTQ	TabFact	ChartQA	TextVQA	MMMU	DudeMini	SlideVQA-M	TableVQA
0.99	DocStruct4M	0.016	0.0	0.037	0.0033	0.0	0.00039	0.034	0.0	0.0026	0.0089	0.0	0.0027
	BigDocs-7.5M	**0.0036**	0.0	**0.0**	**0.0**	0.0	**0.0**	**0.025**	0.0	**0.0**	0.0089	0.0	**0.0**
0.98	DocStruct4M	0.065	0.0032	0.20	0.061	0.0	0.036	0.36	0.0020	0.0044	0.013	0.0	0.0067
	BigDocs-7.5M	**0.012**	**0.0**	**0.0**	**0.0033**	0.0	**0.0**	**0.10**	**0.0**	**0.0035**	**0.011**	0.0	**0.0053**
0.97	DocStruct4M	0.14	0.0032	0.41	0.16	0.0064	0.089	0.55	0.0044	**0.0053**	**0.048**	0.0014	0.0067
	BigDocs-7.5M	**0.049**	**0.0**	**0.0**	**0.043**	**0.0**	**0.0**	**0.17**	**0.0**	0.0061	0.053	**0.0**	**0.0053**
0.96	DocStruct4M	0.26	0.0057	0.57	0.24	0.031	0.14	0.63	0.0096	**0.0070**	0.15	0.0014	0.017
	BigDocs-7.5M	**0.12**	**0.0**	**0.033**	**0.12**	**0.0**	**0.0**	**0.28**	**0.0**	0.021	**0.14**	**0.0**	**0.0087**
0.95	DocStruct4M	0.39	0.0057	0.72	0.30	0.062	0.20	0.68	0.012	**0.015**	0.28	0.0014	0.04
	BigDocs-7.5M	**0.24**	**0.0**	**0.21**	**0.22**	**0.0**	**0.0**	**0.45**	**0.0**	0.045	**0.22**	**0.0012**	**0.023**
Number of samples		5349	2801	1500	4872	4343	12722	2500	5000	1140	5275	19600	1500
Number of unique images		1284	500	300	608	421	1693	1509	3166	1100	609	3596	751



(a) *With respect to BigDocs-7.5M; top 200 samples human-labeled.*



(b) *With respect to DocStruct4M; top 100 samples human-labeled*

Figure 7: ***Human evaluation of DocVQA's overlap.*** *Among the 1284 unique images in DocVQA's samples, we use the same cosine similarity method as in figure 6 to identify the samples that are the most similar to samples in the corresponding training set. Although we prioritize using only the closest match, we also retrieve the second and third closest matches after deduplication (i.e., if the next closest match is identical to the previous match, skip it). A human is then tasked to label the top matches as either "different", "same template" or "same document" (see text for definitions). Counts are provided for the most relevant* 0.01*-wide cosine similarity intervals. Most samples below* 0.96 *are "different", and most samples above* 0.98 *are not. From these numbers, we expect less than* 10% *of the samples with cosine similarity below* 0.97 *to be "same template", and less than* 1% *of the samples with cosine similarity below the same threshold to be "same document". All identified "same document" are found at the first rank. There is only one instance where the first rank is "different" but a "same template" is identified at higher rank.*

Except when the cosine similarity is exactly 1.0 – indicating an exact match – interpretations of these scores must be grounded in human evaluations. However, assessing whether two images are "the same" can be a non-trivial task, even for human eyes.

Consider the following edge cases:

- receipts for recurring orders emitted on different days;
- the same form filed by different people;
- different versions of the same form at the same company; and
- different full-page table from the same appendix of the same report.

Technically, all these cases involve pairs of different documents. However, one could argue that training on one such samples confers an unfair advantage to a model evaluated on the other sample. For this reason, we annotate such instances as *same template* whenever we encounter them.

Conversely, the "actual" same document can appear as quite diversified images. Indeed, stamps, watermarks, censorship, and/or annotations may be apposed *a posteriori* to a document, in addition to scaling, crops, rotations, and fax-induced artifacts. In all these cases, the original intent to

communicate the same information matters: different copies of the same memorandum, scanned separately, are here treated as the *same document*.

With these definitions in hand, a human is tasked with labeling the samples that are most likely to be problematic in the DocVQA evaluation dataset. Figure 7 reports the results, calibrating how we interpret the cosine similarities for other evaluation datasets.

Note that this labeling procedure does not alter the composition of BigDocs-7.5M: we leave all samples, problematic or not, in the dataset. Instead, we release the annotations, which may help the community develop an intuition of the overlaps that may not have been identified yet, or even enable better detection strategies in the future.

A.5 Description of Downstream Tasks proposed in BigDocs-Bench

The following is a formal description of the downstream tasks we aim to solve using the proposed BigDocs-Bench dataset.

A.5.1 Image2Flow

The task at hand is an image-to-flow conversion, where the input is an image of a flowchart, and the output is the corresponding information in JSON format. This JSON includes the nodes and edges that represent the flowchart's structure.

Formally, given an image I representing a flowchart, the goal is to extract a graph $G = (V, E)$ where V is the set of nodes, and E is the set of directed edges between these nodes. The output JSON(G) contains two main components: (1) a list of nodes V with their corresponding attributes such as node ID, label, shape, and description, and (2) a list of edges E with their source node ID, destination node ID, and label (if any).

The dataset for this task consists of 10,000 samples. Each sample includes an image of a flowchart, its corresponding Graphviz file, and a JSON representation. On average, each flowchart contains 12.51 nodes, with a variance depending on the random generation process described below.

The dataset was generated using the following pipeline: First, a random number between 5 and 15 was generated to determine the number of nodes in each flowchart. Based on the total number of nodes, a distribution was assigned to different types of nodes, such as conditions (diamond shape), statements (rectangular nodes), and others (parallelogram, circle). A random flow direction was then selected from the options: BT (bottom to top), TB (top to bottom), LR (left to right), and RL (right to left). Using the LLaMA 3.1 model, we used the prompt in Table A.5.1 to generate a Graphviz file.

Prompt used with Llama 3.1 to generate Graphviz data

```
Create a directed flowchart graph in DOT format with the following
specifications:
- The direction of the graph should be {direction}.
- Total number of nodes: {total_nodes}.
- Nodes distribution: {nodes}.
- The edges should connect the nodes in a way that makes sense:
* Each node should have at least one outgoing edge.
* For 'diamond' nodes, there should be at least two outgoing edges. The
graph can include any colors or additional styling. Generate a valid
and coherent graph. The graph should represent an enterprise-level
workflow like the steps of an incident resolution workflow. The
enterprise-level workflow labels are so important. Just give me the
graph in Graphviz format no need for any python code or any description
about the graph.
```

Random colors and styles were applied to remove model biases toward specific stylings. The generated Graphviz files (.gv) were then converted to PNG images and JSON files. The JSON files contain two main components: Each node has an ID, label, shape, and a description of the shape (e.g.,

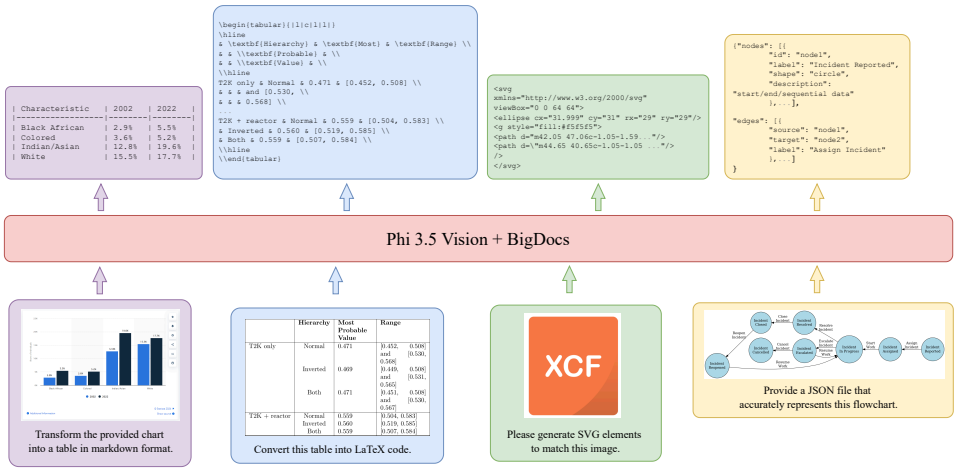


Figure 8: ***Qualitative Results.*** *Generations of our Phi3.5-Vision model on the presented document downstream tasks, as part of the test set of BigDocs-Bench. We show samples of Chart2Markdown, Table2LaTex, Image2SVG, and Image2Flow (JSON). A single model trained on BigDocs datasets can generate image-conditioned code in different coding languages while providing valid outputs.*

diamonds represent conditions). Each edge contains the source and destination node IDs and a label if present (the label represents the text on the edge).

A.5.2 Screenshot2HTML

The Screenshot-to-HTML conversion task involves transforming an image of a website's layout, such as a screenshot, into the corresponding HTML code that accurately reconstructs the structure and content of the original website. This process enables the generation of a functional website solely from its visual representation, facilitating applications in web design automation, accessibility enhancements, and rapid prototyping.

Formally, given an input image I depicting the visual layout of a website, the objective is to generate the HTML structure H that includes essential web elements such as headers, paragraphs, images, links, forms, and navigation bars. The resulting HTML code HTML(H) should not only replicate the visual appearance of the original website but also ensure semantic correctness and structural integrity, enabling the recreated website to be interactive and accessible.

Data Collection and Filtering Process. The Image-to-HTML dataset was curated through an automated pipeline designed to ensure the quality and diversity of website layouts. Utilizing the `Playwright` library, the system asynchronously retrieves and renders websites from a comprehensive list of URLs (Penedo et al., 2024). Each website undergoes a series of checks to verify its accessibility, compliance with `robots.txt` directives, and predominance of English content. Additionally, content appropriateness is assessed by filtering out websites containing NSFW language.

External CSS and JavaScript resources are inlined to maintain consistency and reduce dependencies, and unnecessary or oversized scripts are removed. Images are replaced with placeholders, and background images are eliminated to focus on structural elements. The viewport is adjusted to capture only the visible portion of each page, enhancing the clarity of the layout representations.

Websites are further evaluated based on performance and structural metrics, including load time, page size, number of JavaScript and CSS files, DOM depth, and total number of HTML elements. Technologies and frameworks used by each website are identified to exclude those utilizing disallowed technologies, e.g., allowing a website to render without JavaScript. We also removed comments and prettified the HTML for standardization. Only websites that meet all predefined criteria are included in the final dataset.

Dataset Statistics. The resulting dataset comprises 11,000 website samples, each with a high-resolution screenshot and its corresponding HTML representation. On average, each layout contains 20.3 HTML elements, reflecting a diverse range of website designs. This diversity provides a robust

foundation for training and evaluating image-to-HTML conversion models, ensuring the dataset is both comprehensive and representative of various web structures.

A.5.3 Table2LaTex

The task involves identifying and associating tables in academic PDFs with their corresponding LaTeX code and captions. The aim is to precisely match each table image with the LaTeX source used to generate it and the relevant caption, ensuring accurate alignment between the visual content and its textual description.

More specifically, given a table image I, the LaTeX code C used to render the table, and the caption T_{caption}, the goal is to create a dataset Dataset$(I, C, T_{\text{caption}})$ that establishes a reliable association between the tables, their LaTeX source, and their descriptive captions.

We crawled publicly available, license-compliant arXiv papers to create this dataset, collecting both their PDFs and associated TeX source files. We began by parsing the TeX files to extract the LaTeX code for tables and their captions. Next, we used the PDF parsing library `PyMuPDF` to locate tables within the PDFs. However, this table detection process proved to be imperfect, as the algorithm frequently misidentified content with parallel lines as tables, leading to false positives.

To address this challenge, we adopted an alternative approach. Instead of cropping tables directly from the PDFs—where false positives were common—we chose to render the parsed LaTeX table code to generate accurate table images. This method ensured that the images faithfully represented the original table formatting, reducing detection errors and improving the reliability of the dataset. As a result, we created a high-quality dataset comprising over 95,000 table images, each paired with its corresponding LaTeX code and caption, providing a valuable resource for further research into table structures in academic papers.

A.5.4 Image2SVG & Text2SVG

Scalable Vector Graphics (SVG) provide a precise alternative to pixel-based images, capable of representing diagrams, icons, plots, and graphic designs with superior detail and scalability. Unlike raster images, SVGs can be scaled to any resolution without losing quality. In this work, we introduce the task of image-to-SVG generation, where the goal is to process an input image and generate SVG code that closely resembles the image upon rendering. This task requires advanced parsing capabilities for textual and numerical data and an understanding of various shapes, such as squares and arrows, commonly found in diagrams. Given an input image I, the model outputs SVG code C that visually replicates the image. Additionally, the task can extend to scenarios where a textual description T serves as input, yielding SVG code that aligns with the described content.

We curate a dataset of images, SVG codes, and texts sourced from the SVG-Stack dataset introduced by StarVector (Rodriguez et al., 2023a). The curation process involves filtering to ensure high-quality samples. First, we filter based on image entropy to select images with complex designs and intricate details, excluding simpler icons or shapes. Second, we compute the CLIP Score (Hessel et al., 2022; Radford et al., 2019) for image-text pairs and retain the top 100k examples to build our curated SVG dataset. The SVG-Stack dataset adheres to permissive licensing standards, originating from TheStack (Kocetkov et al., 2022), carefully designed for open and permissive use.

A.5.5 Chart2Markdown

This dataset is a novel contribution of our work, designed to assess the models' capabilities in extracting data values from chart images. In this task, the model is given a chart image I and asked to produce a data table T of the underlying data table in markdown format.

To create this dataset, we crawled recent chart images from the Statista website [‡], focusing on charts from 2023 and 2024 that were not used in prior datasets like UniChart Masry et al. (2023) and ChartQA Masry et al. (2022). This ensures that the dataset reflects the most up-to-date facts and trends and overlaps less with existing datasets and benchmarks. We collected 6,516 chart images, their corresponding data tables, and human-written summaries.

[‡]https://www.statista.com/

A.5.6 Chart2Caption

The task at hand is a chart-to-caption conversion. The input consists of an image of a chart along with the code used to generate it and the dataset's name and description. The output is a textual caption of the important insights and information conveyed by the chart.

Formally, given a chart image I, the code C used to generate the chart, and the dataset information D, the goal is to produce a caption Caption(I, C, D) that highlights key insights and information represented in the chart.

The dataset for this task consists of 1,496 pairs of chart images and the corresponding code that generated these charts. The data was collected by selecting various Kaggle public datasets and their associated data analytics notebooks. We executed these notebooks locally and parsed their outputs to generate the chart-image and code pairs.

To generate the captions, we used the prompt below with the provided chart image, code, dataset name, and description. The caption was generated by using InterVL2-26B (Chen et al., 2023, 2024). This process allows us to leverage the model's capabilities to generate meaningful captions based on the provided context of the chart, code, and dataset description.

Prompt used with InternVL2-26B to generate chart summaries

```
You are a powerful data analyst.  This is a notebook with name
"{dataset_name}".  In the description of this dataset, it's told that:
"{dataset_description}".  You are seeing a plot from this notebook.
Here is the code that was used to generate this plot:
{code}
Now as a data analyst, summarize the important insights and information
about the chart.
```

A.5.7 GUI2UserIntent

The GUI2UserIntent task tests the abilities of grounding on GUI. Concretely, given the bounding box coordinate clicked by the user, the goal is to identify the text element the user intends to interact with. While this is similar to the GUI grounding task introduced in Cheng et al. (2024), which predicts the bounding box based on a user query, it differs in its focus on interpreting user clicks. The datasets are curated by repurposing the pretraining dataset for SeeClick (Cheng et al., 2024). The original dataset includes clickable text elements and their bounding box coordinates in website screenshots. We directly extracted them to curate the GUI-to-UserIntent dataset.

A.5.8 GUI2Summary

The GUI2Summary task is similar to the Chart-to-Summary task in NovelCharts; however, instead of charts, the input images I are website screenshots. Unlike existing UI summary datasets like Screen2Words (Wang et al., 2021) that focus on short, phrase-level summaries, our GUI-to-Summary dataset provides paragraph-level summaries. These summaries are comprehensive, providing a brief overview of the main content, referencing key visual and textual elements in the screenshots, and additional aspects like layout and color schemes. To synthesize data for the GUI-to-Summary task, we used InternVL2-8B (Chen et al., 2023), prompting it with website screenshots to create concise descriptions of the main content and layout. Appendix A.5.8 shows the specific prompt used for data generation.

Prompt used with InternVL2-8B to generate website screenshot summaries

```
Summarize this website in less than 100 words.  Cover the main content,
layout, color, and other style elements.
```

A.5.9 GUI-VQA

In the GUI-VQA task, the model answers questions about website screenshots, covering the overall content and specific elements like buttons, text boxes, and menu items. This task requires recognizing

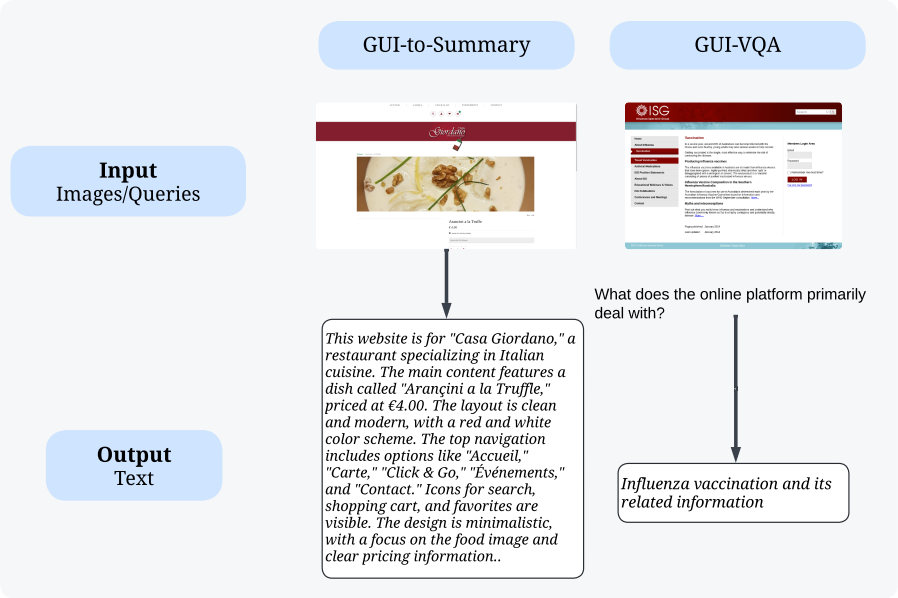


Figure 9: ***GUI-to-Summary and GUI-VQA introduced in BigDocs-Bench.***

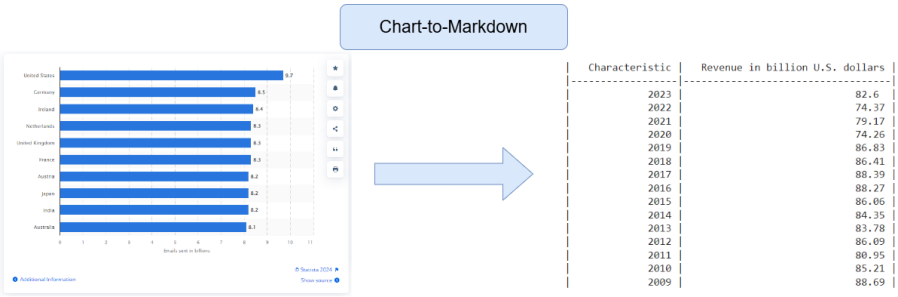


Figure 10: ***Chart-to-Markdown task introduced in BigDocs-Bench.***

key components within the GUI and understanding their functionalities and interdependencies. We synthesized data for GUI-VQA using the GUI-to-Summary dataset. Specifically, we sampled a sentence from each summary and prompted LLaMA 3.1-8b (Dubey et al., 2024) to convert it into a QA pair. The prompt is shown in Appendix A.5.9.

Prompt used with InternVL2-8B to convert a summary to a QA pair.

```
You will be provided with a sentence.  Your task is to convert it into a
question-answer pair.  The question should focus on factual information
and avoid subjective inquiries.  Do not generate Yes/No questions.
Structure the response in the format:
Q: {question}
A: {answer}
```

A.6 Details on Training Vision-Language Models on BigDocs

Figure 11 illustrates our training pipeline for evaluating the quality of BigDocs and its impact compared to other datasets. The process has three stages: (1) Pretraining, which focuses on general modality alignment; (2) Continual pretraining, where models are further aligned to a specific domain, such as documents, using general tasks like OCR and captioning; and (3) Finetuning, which uses smaller datasets to prepare models for specific downstream tasks, like document processing.

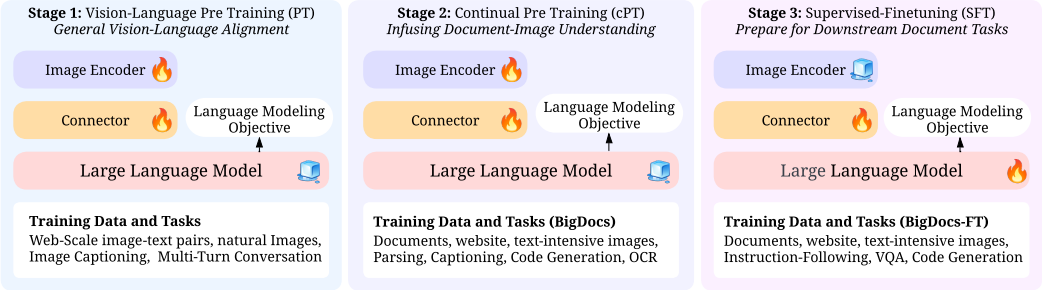


Figure 11: ***Training Stages for BigDocs.*** *Three-Stage Training Pipeline for multimodal Document Understanding with BigDocs. Our approach consists of: (1) General Vision-Language Pretraining, (2) Document-Specific Continual Pretraining, and (3) Supervised Finetuning for Document Tasks. We evaluate the impact of BigDocs by using checkpoints of models after Stage 1, as provided by their original authors, and comparing them with models that undergo all three stages. Performance is assessed based on standard document tasks and novel tasks, including HTML/SVG code generation, flowchart generation/parsing, and LaTeX interpretation.*

We did not perform stage 1 in this paper, instead relying on checkpoints after pretraining. Additionally, we conducted some experiments using instruction-tuned checkpoints. Throughout all three stages, we employ a generative loss objective, specifically a categorical cross-entropy loss, to predict the next token given the context. The goal of BigDocs-7.5M is to enhance the model's understanding and alignment with document-specific structures. At the same time, BigDocs-Bench focuses on training models to handle tasks that require processing images and converting them into structured code representations. These outputs often follow strict validity constraints to ensure they are syntactically correct. Additionally, the BigDocs-Bench test sets introduce novel benchmarks comprising five company-related downstream tasks, further validating the model's ability to generalize to real-world document-based applications.

Training Details. We conduct all our experiments on a cluster of 64 H100 80GB GPUs leveraging the Transformers library (Wolf et al., 2019) for model development. Using Accelerate (Gugger et al., 2022) and Fully Sharded Data Parallel (FSDP) (Zhao et al., 2023), we distribute training over all GPUs in our cluster. We wrap the transformer blocks in both the encoder and the decoder into separate FSDP units with activation checkpointing to achieve a data parallel of 64 without running out of memory. FlashAttention-2 (Dao, 2024) gives a significant speedup on our training runs. For reproducibility, we provide the hyperparameters and training details of our experiments. All experiments maintain a constant total batch size of 256 and a learning rate of 2e-5. We utilize the AdamW optimizer with the following parameters: a cosine learning rate scheduler, 60 warm-up steps, a beta1 coefficient of 0.95, a beta2 coefficient of 0.999, a weight decay of 1e-6, and an epsilon value of 1e-8. For the CPT experiments, models are trained for 1 epoch, while the finetuning experiments on the DocDownstream dataset and BigDocs-Bench are conducted over 3 epochs.

A.7 Details on BigDocs-Bench's Evaluation

In this section, we dive into more details of the evaluation process for BigDocs-Bench, including the preprocessing procedure before the evaluation and the implementation details of the metrics.

A.7.1 Preprocessing

Before conducting the evaluation, we perform the following preprocessing step for both the ground truth and generation texts:

1. **Image2Flow (GraphViz)**: If the text contains markdown-style fenced code blocks, we use a regular expression to extract the contents within the first code block; otherwise, we treat the entire text as GraphViz code and remove anything before `digraph` or `graph` if either exists. The flowchart triplets are extracted with the `pydot` library.
2. **Image2Flow (JSON)**: If the text contains markdown-style fenced code blocks, we use a regular expression to extract the contents within the first code block; otherwise, we remove all contents outside the first "{" and last "}" in the text.

3. **Screenshot2HTML**: If the text contains markdown-style fenced code blocks, we use a regular expression to extract the contents within the first code block; otherwise, we treat the entire text as the output HTML code. We first normalize the HTML code with `htmlmin.minify()`, which removes the comments in HTML and condenses the attributes to their most miniature possible representations. Then, we create the DOM tree with `BeautifulSoup4`.
4. **Table2LaTeX**: If the text contains markdown-style fenced code blocks, we use a regular expression to extract the contents within the first code block; otherwise, we treat the entire text as the output LaTeX code. We use a series of regular expressions to normalize the contents. This procedure removes the comments, excessive whitespaces, and commands such as `\label`, `\cite`, `\citep`, `\citet`, `\ref`, `\eqref`, and `\pageref`.
5. **Image2SVG & Text2SVG**: If the text contains markdown-style fenced code blocks, we use a regular expression to extract the contents within the first code block; otherwise, we treat the entire text as the output SVG code. We parse the SVG code and generate a PNG image with the `cairosvg` library.
6. **Chart2Markdown**: If the text contains markdown-style fenced code blocks, we use a regular expression to extract the contents within the first code block; otherwise, we treat the entire text as the output Markdown code. We use regular expressions to remove the code's comments, links, and excessive whitespaces.
7. **Chart2Caption, GUI2UserIntent, GUI2Summary, & GUI-VQA**: We directly evaluate the generated texts against the ground truth.

A.7.2 Metrics

Here, we provide brief explanations of BigDocs-Bench's evaluation metrics.

1. **Flowchart Triplet F1**: Designed for Image2Flow tasks, this metric evaluates the accuracy of node relationships in flowchart codes, focusing on the correctness of edge triplets (s, e, d) extracted from GraphViz or JSON representations. Here, e denotes the edge label (set to `None` if unlabeled), while s and d represent the source and destination nodes, formatted as "label#shape". The Triplet F1 score is calculated by comparing the generated triplet list against the ground truth, disregarding the ordering of nodes and edges to emphasize relational accuracy.
2. **HTML DOM Tree Edit Distance**: Applicable to Screenshot2HTML tasks, this metric measures the similarity between generated and ground-truth DOM trees using the Tree Edit Distance. Leveraging `BeautifulSoup4` with `lxml` parsing and the `zss` library, the distance is normalized by the larger node count of the compared trees. Invalid GraphViz or JSON generations receive a Triplet F1 score of 0.
3. **TeXBLEU** (Jung et al., 2024)[§]: Utilized for the Table2LaTeX task, TeXBLEU employs a LaTeX-trained tokenizer and a finetuned embedding model with positional encoding. Unlike traditional BLEU, TeXBLEU assesses similarity based on n-gram token precision, demonstrating a higher correlation with human evaluations of LaTeX math expressions compared to BLEU, SacreBLEU, and Rouge (Jung et al., 2024).
4. **RMSF1** (Liu et al., 2022)[¶]: Used for the Chart2Markdown task, RMS F1 treats tables as mappings from headers to values, measuring textual and numeric similarity through normalized Levenshtein distance and relative distance. This approach ensures robustness against row and column permutations and transpositions, making it well-suited for evaluating flexible Markdown table structures.
5. **DINOv2Score**: Employed in the Image2SVG task, this metric calculates the cosine similarity between representations of the ground-truth and generated images using DINOv2 (Oquab et al., 2023), which better captures image similarity than comparable models [‖]. Invalid SVG generations are assigned a DINOv2Score of 0.

[§]Available at https://github.com/KyuDan1/TeXBLEU.
[¶]Available at https://github.com/google-research/google-research/tree/master/deplot.
[‖]https://medium.com/aimonks/clip-vs-dinov2-in-image-similarity-6fa5aa7ed8c6

6. **Rouge-L F1** (Lin, 2004): Applied to summarization and VQA tasks, Rouge-L F1 measures the longest common subsequence between the generated and reference texts. We compute this score using the implementation provided by `torchmetrics`.

A.8 Making BigDocs License-Permissible

We dedicated significant effort to acquiring a license-permissible dataset suitable for training models for commercial purposes in document images. We aim to create a large-scale dataset that supports various tasks relevant to companies while adhering to accountability, responsibility, and transparency principles. To achieve this, we thoroughly investigated all public datasets concerning their licenses, evaluating both the sources of the images and their annotations. Our complete analysis, summarized in Table 5, enabled us to identify and retain only the sources that meet permissive licensing criteria.

Dataset licensing frameworks are crucial in determining how data can be used, shared, and modified, generally falling into two categories: *permissive* and *restrictive* licenses. *Permissive licenses* offer the most freedom, allowing for broad usage and modification of the data. For instance, the **CC0** license places the data in the public domain, enabling unrestricted use, modification, and distribution. The **MIT License** permits both commercial and non-commercial applications, provided the license terms are retained in any distribution. In contrast, the **Apache 2.0** License extends these freedoms with an additional grant of patent rights.

In contrast, *restrictive licenses* imposes certain limitations on data usage. The CC BY license requires users to provide attribution to the original creator. In contrast, the **CC BY-SA** license demands that any derivative works also carry the same licensing terms. More restrictive options, like the **CC BY-NC** and **CC BY-ND** licenses, prohibit commercial use and modifications, respectively. In cases where the licensing terms are *unclear*, it is prudent to exercise caution in using the data, as misinterpretation can lead to legal risks.

Additionally, the concept of **Fair Use** allows for limited use of copyrighted material without explicit permission, particularly for purposes such as research, criticism, or commentary. However, Fair Use does not equate to unrestricted permission, and its applicability can vary, necessitating careful consideration when applied to dataset usage in research contexts.

A.9 Human Evaluation

We developed a web application using Flask, Javascript, and HTML/CSS where users are presented with pairs of outputs from two models and are asked to judge which model has better output for a given input image. See Figure 12 for a snapshot of the platform.

A.10 Details of Unified Metadata Framework

We propose a unified metadata framework for the BigDocs dataset to ensure transparency and traceability. This framework is organized into three primary keys: `license`, `origin`, and an optional `features` section. By adopting this structure, we provide a standardized and flexible system that enhances the dataset's usability, ensuring clarity for research and commercial applications.

Along with the dataset, we include two separate files named *`tracking_instructions.json`* and *`tracking_transformations.json`*, which are dictionaries that store information standards across large subsets of samples, reducing redundancy and promoting consistency in metadata.

License Metadata The `license` key is mandatory and provides detailed information regarding the licensing terms of both the images and annotations. It is structured as a dictionary with three sub-keys:

- `image_license` (mandatory): Specifies the license under which the image is provided.
- `annotation_license` (mandatory): Indicates the license for the annotations or labels associated with the image (e.g., text, bounding boxes, tables).
- `license_note` (optional): Includes additional licensing information or clarifications, such as specific usage conditions or exceptions. This allows for clear communication of any nuances in licensing terms.

Table 5: ***Datasheet of candidate datasets.*** *For transparency purposes, we provide a detailed description of over 100 datasets considered in curating our BigDocs dataset. This table presents a systematic analysis of public datasets, including information on medium types, source documents, text structures, languages, years, annotation types and methods, and licensing for both data and annotations. We also share sample counts across different modalities and splits. Some fields may be blank due to unavailable information. This comprehensive overview enables assessment of each dataset's characteristics and potential contributions to BigDocs.*

Dataset Name	Data						Annotations		Licenses			Total Size		Units	
Acronym	Medium	Source Document Type	Data Sourcing	Text Type	Text Structure	Text Languages	Annotations Type	Annotation method	Images	Annotations	Permissive	Documents	Annotations	Documents	Annotations
unichart-table	Photo	Article	Dataset	Computer Generated	Structures - Charts	English	Layout	Automated	MIT	MIT	Good to use	304997	304997	Image - Figure	Full page annotation
unichart-qa	Photo	Article	Dataset	Computer Generated	Structures - Charts	English	Q&A (Question and Answer)	Synthetic	MIT	MIT	Good to use	170679	300000	Image - Figure	Element - Q&A pairs
DocBank	Digital, Word, Latex	Article - Scientific paper	Repository - ArXiV	Computer Generated	Text with Structures	English	OCR, Layout	Weak Supervision	Apache 2	Apache 2	Good to use	500000	500000	Page	Full page annotation
TableBank	Digital, Word, Latex	Article - Scientific paper	Crawling, Repository - ArXiV	Computer Generated	Text with Structures	English, Chinese, Japanese, Arabic, Other	OCR, Table Detection/Extraction	Weak Supervision	Unknown license	Apache 2	Borderline	424045	562697	Multi-page	Element - Table structure
DocVQA	Scanned, Digital	Legal, Biussines	Repository - Government	Handwritten, Typewritten, Computer Generated	Structures, Text with Structures	English	OCR, Q&A (Question and Answer)	Annotators - Crowdsourced	Fair Use	MIT	Borderline	12767	50000	Multi-page	Element - Q&A pairs
InfographicVQA	Digital	Infographics	Crawling, Repository	Computer Generated	Infographics	English	Q&A (Question and Answer)	Annotators - In house	Unknown license	CC BY	Borderline	5485	30035	Image	Element - Q&A pairs
TQA	Digital	Book - Textbook, Infographics	Repository	Computer Generated	Text with Structures	English	Q&A (Question and Answer)	Weak Supervision	CC BY-NC 3.0		Not good to use	1076	26260	Multi-page	Element - Q&A pairs
HierText	Photo	Natural Scene	Dataset	Computer Generated	Natural Image	English	OCR, Layout	Weak Supervision	CC BY 2.0	CC BY-SA	Good to use	11639	1208128	Image	Word
RecipeQA	Digital - Website	Book - Manuals	Repository	Computer Generated	Text with Images	English	Q&A (Question and Answer)	Weak Supervision	Various Licenses	Various Licenses	Various	19779	36786	Multi-page	Element - Q&A pairs
ST-VQA	Photo	Natural Scene	Dataset	Computer Generated	Natural Image	English	Q&A (Question and Answer)	Annotators - Crowdsourced		Unknown license	Not sure	22020	30471	Image	Element - Q&A pairs
FigureQA	Digital	Synthetic	Synthetic	Computer Generated	Structures - Figures	English	Q&A (Question and Answer)	Synthetic	MIT	MIT	Good to use	140000	1800000	Image	Element - Q&A pairs
OCR-VQA	Scanned	Infographics	Dataset	Computer Generated	Infographics - Covers	English	Q&A (Question and Answer)	Weak Supervision	Fair Use	Apache 2	Borderline	207572	1000000	Image	Element - Q&A pairs
DVQA	Digital	Synthetic	Synthetic	Computer Generated	Structures - Charts	English	Q&A (Question and Answer)	Weak Supervision	CC BY-NC 4.0	CC BY-NC 4.0	Not good to use	300k	3487194	Image	Element - Q&A pairs
Bunny-v1.0-data	Photo	Natural Scene		Computer Generated	Natural Image		Q&A - Dialogs					2000000			
EXAMS-V	Digital	Academic	Repository - Government	Computer Generated	Text with Structures	English, Chinese, French, German, Italian, Arabic, Polish, Hungarian, Bulgarian, Croatian, Serbian	Q&A (Question and Answer)	Weak Supervision	CC BY-SA 4.0	Apache 2	Good to use	20932	20932	Multi-page	Element - Answers
DocILE	Digital, Scanned	Biusines	Repository - Government	Computer Generated	Structures	English	KIE (Key Information Extraction)	Annotators - Crowdsourced	Custom	Custom	Not good to use	106608	106608	Multi-page	Full page annotation
LEAF-QA	Digital	Synthetic	Repository - Government	Computer Generated	Structures - Charts	English	Q&A (Question and Answer)	Weak Supervision	Unknown license	Unknown license	Not sure	240k	1.8M	Image - Figure	Element - Q&A pairs
WikiTQ	Digital	Article - Wikipedia	Repository - Wikipedia	Computer Generated	Structures - Tables	English	Q&A (Question and Answer)	Annotators - Crowdsourced		CC BY-SA 4.0	Good to use	2108	22033	Image - Table	Element - Q&A pairs
VisualMRC	Digital - Website	Article	Crawling	Computer Generated	Text with Structures	English	Q&A - Refered to a ROI	Annotators - Crowdsourced	Unknown license	Custom	Not good to use	10000	30562	Image	Element - Q&A pairs
Open4Business	Digital	Report - financial	Repository - Government	Computer Generated	Text with Structures	English	Summary	Weak Supervision	CC BY 3.0, CC BY 4.0	CC0	Good to use	17458	17458	Multi-page	Element - Summary
TabFact	Digital	Article - Wikipedia	Repository - Wikipedia	Computer Generated	Structures - Tables	English	Q&A - True/False sentences	Annotators - Crowdsourced	CC BY-SA 4.0	CC BY 4.0	Good to use	16575	117854	Image - Table	Element - Sentence
FinTabNet.c	Digital	Report - Financial	Repository - Government	Computer Generated	Structures - Tables	English	OCR, TD, TSE	Weak Supervision	CDLA-Permissive	CDLA-Permissive	Good to use	112887	112887	Image - Table	Element - Table structure
SVIT	Photo	Natural Scene	Dataset	Computer Generated	Natural Image	English	Q&A (Question and Answer)	LLM generated - No permissive like GPT	Various Licenses	CC BY 4.0	Various	108077	4240311	Image	Element - Q&A pairs
Kleister Charity (KLC)	Digital	Report	Repository - Government	Computer Generated	Text with Structures	English	NER (Name-Entity Recognition)		Unknown license	Unknown license	Not sure	2778	2778	Multi-page	Full page annotation
Kleister-NDA	Digital - PDF								Unknown license	Unknown license	Not sure				
CORD	Scanned	Busuines - Receipts	Crowdsourced	Computer Generated	Structures - Forms	English	OCR, Layout, KIE	Annotators - Crowdsourced	CC BY 4.0	CC BY 4.0	Good to use	1000	1000	Image	Full page annotation
FUNSD	Scanned	Busuines	Dataset	Typewritten, Computer Generated	Structures - Forms	English	OCR, Layout, KIE	Annotators - Crowdsourced		Custom	Not good to use	199	199	Image	Full page annotation
DeepForm	Digital - PDF	Administrative	Repository - Government - FCC-PIF	Computer Generated	Text with Structures	English	KIE (Key Information Extraction), OCR	Annotators - Volunteers		MIT	Not sure	20000	20000	Multi-page	Full page annotation
PWC											Not sure	2000			
COCOText	Photo	Natural Scene	Dataset	Computer Generated	Natural Image	English	Caption - Bounding box	Annotators - Crowdsourced	Unknown license	CC BY 4.0	Borderline	63686	173589	Image	Word
TextOCR	Photo	Natural Scene	Dataset	Computer Generated	Natural Image	English		Annotators - In house	CC BY 2.0	CC BY 4.0	Good to use	28134	903069	Image	Word
PubTables-1M-TSR-TFA	Digital - PDF	Article - Scientific paper	Repository - Pubmed	Computer Generated	Structures - Tables	English	OCR, TD, TSE	Automated		MIT	Good to use	948000	-	Multi-page	
PubTables-1M-TD	Digital - PDF	Article - Scientific paper	Repository - Pubmed	Computer Generated	Structures - Tables	English	OCR, TD, TSE	Automated		MIT	Good to use	948000	-	Multi-page	
SROIE	Scanned	Busuines - Receipts	Dataset	Computer Generated	Structures - Forms	English	OCR, Layout	Annotators	Unknown license	MIT	Not sure	1000	-	Image	Full page annotation
VRDU-Registration-Form	Scanned	Administrative	Repository - Government	Computer Generated	Structures - Forms	English	OCR, Layout	Annotators - In house	Unknown license	Unknown license	Not sure	1915	-	Multi-page	
VRDU ad-buy	Digital - PDF	Administrative	Repository - Government	Computer Generated	Structures - Forms	English	OCR, Layout	Annotators - In house	Unknown license	Unknown license	Not sure	641	-	Multi-page	
Websight1	Digital - Website	Synthetic	Synthetic	Computer Generated	Text with Images	English		LLM generated - Permisive like Llama	CC BY 4.0	CC BY 4.0	Good to use	823000	-	Image - Screenshot	Code
Websight2	Digital - Website	Synthetic	Synthetic	Computer Generated	Text with Images	English		LLM generated - Permisive like Llama	CC BY 4.0	CC BY 4.0	Good to use	2000000	-	Image - Screenshot	Code
TextVQA	Photo	Natural Scene	Dataset	Computer Generated	Natural Image	English	OCR, Q&A (Question and Answer)	Annotators - In house	CC BY 2.0	CC BY 4.0	Good to use	28408	45336	Image	Element - Q&A pairs
Paper2Fig100k	Digital	Article - Scientific paper	Repository - ArXiV	Computer Generated	Structures - Figures	English	Caption	Automated	Various Licenses	CC BY 4.0	Borderline	102453	102453	Image	Element - Caption
CommonCatalog	Photo	Natural Scene	Repository	No Text			Caption	LLM generated - Permisive like Llama	Various Licenses	Various Licenses	Good to use	26232417	34.3M	Image	Element - Caption
RVL-CDIP	Scanned	Article, Report, Legal documents...	Repository - Government	Handwritten, Typewritten, Computer Generated	Text with Structures	English	OCR, Class	Automated			Good to use	399999	399999	Image	Full page annotation
IIT CDIP Colection	Scanned	Article, Report, Legal documents...	Repository - Government	Handwritten, Typewritten, Computer Generated	Text with Structures	English	OCR, Class	Automated			Good to use	11000000	11000000	Image	Full page annotation
OCR-IDL	Scanned	Article, Report, Legal documents...	Repository - Government	Handwritten, Typewritten, Computer Generated	Text with Structures	English	OCR, Class	Automated		WTFPL	Good to use	26621635	26621635	Page	Full page annotation
PDFA	Digital - PDF	Various	Crawling	Various	Text with Structures	English	OCR	Automated	CommonCrawl	CommonCrawl	Not sure				
ShareGPT4V	Photo	Natural Scene	Dataset	No Text			Caption	LLM generated - No permissive like GPT	Various Licenses	CC BY-NC 4.0	Not good to use	100000	100000	Image	Element - Caption
ShareGPT4V-PT	Photo	Natural Scene	Dataset	No Text			Caption	LLM generated - No permissive like GPT	Various Licenses	CC BY-NC 4.0	Not good to use	1200000	1200000	Image	Element - Caption
LLaVA-Instruct-150K	Photo	Natural Scene	Dataset	No Text			Q&A - Dialogs	LLM generated - No permissive like GPT		CC BY 4.0	Not good to use	158000	158000	Image	Element - Q&A pairs
LLaVA-CC3M-Pretrain-595K	Photo	Natural Scene	Dataset	No Text			Q&A - Dialogs	LLM generated - No permissive like GPT		CC BY 4.0	Borderline	595000	595000	Image	Element - Q&A pairs
LAION-400M	Photo	Natural Scene	Crawling	No Text		English	Caption	Automated	CommonCrawl	CC BY 4.0	Various	413000000	413000000	Image	Element - Caption
LAION-5B	Photo	Natural Scene	Crawling	No Text		Multilingual	Caption	Automated	CommonCrawl	CC BY 4.0	Various	5850000000	5850000000	Image	Element - Caption
LAION-COCO	Photo	Natural Scene	Dataset	No Text		English	Caption	LLM generated - Permisive like Llama	CommonCrawl	CC BY 4.0	Various	600000000	600000000	Image	Element - Caption
LAION-Aesthetic	Photo	Natural Scene	Dataset	No Text		English	Caption	LLM generated - Permisive like Llama	CommonCrawl	CC BY 4.0	Various	1200000000	1200000000	Image	Element - Caption
LVIS Instruct 4V	Photo	Natural Scene	Dataset	No Text		English	Q&A - Dialogs	LLM generated - No permissive like GPT		MIT	Borderline	100000	200000	Image	Element - Q&A pairs
Visual Genome	Photo	Natural Scene	Dataset	No Text		English	Q&A - Refered to a ROI	Annotators - Crowdsourced	Various Licenses	CC BY 4.0	Good to use	108077	1773258	Image	Element - Q&A pairs
OK-VQA	Photo	Natural Scene	Dataset	No Text		English	Q&A (Question and Answer)	Annotators - Crowdsourced	Various Licenses	Unknown license	Not sure	14031	14055	Image	Element - Q&A pairs
A-OKVQA	Photo	Natural Scene	Dataset	No Text		English	Q&A (Question and Answer)	Annotators - Crowdsourced	Various Licenses	Unknown license	Not sure	23700	24900	Image	Element - Q&A pairs
VisDial	Photo	Natural Scene	Dataset	No Text		English	Q&A - Dialogs	Annotators - Crowdsourced	Various Licenses	Unknown license	Not sure	133060	1336600	Image	Element - Q&A pairs
Hateful Memes	Digital	Infographics	Crowdsourced	Computer Generated	Text with Images	English	Class	Annotators - Crowdsourced	Custom	Custom	Not sure	10000	10000	Image	Element - Class
Tablet-LLaVA	Digital	Article	Dataset	Computer Generated	Structures - Tables	English	Q&A (Question and Answer)	LLM generated - No permissive like GPT	Various Licenses	Unknown license	Not good to use	97000	150000	Image - Table	Element - Q&A pairs

Origin Metadata The `origin` key is mandatory and serves as a traceability mechanism, allowing users to track each data sample back to its source. This key is also structured as a dictionary and includes the following sub-keys:

- `data_id` (mandatory): A unique identifier for each data sample (e.g., file name, index, or ID in the source dataset).
- `tracking_instruction_id` (mandatory): A reference to an entry in the *`tracking_instructions.json`* file, specifying how to use the metadata fields to trace the data sample back to its source.
- `tracking_transformation_id` (optional): A reference to an entry in the *`tracking_transformations.json`* file, providing a high-level description of any transformations or modifications applied to the original sample.
- `dataset_url` (optional): The URL of the original dataset from which the sample was obtained, if applicable.
- `image_path` (optional): The local path to the image within the raw dataset files, if available. This key is helpful when the dataset is stored in raw form.

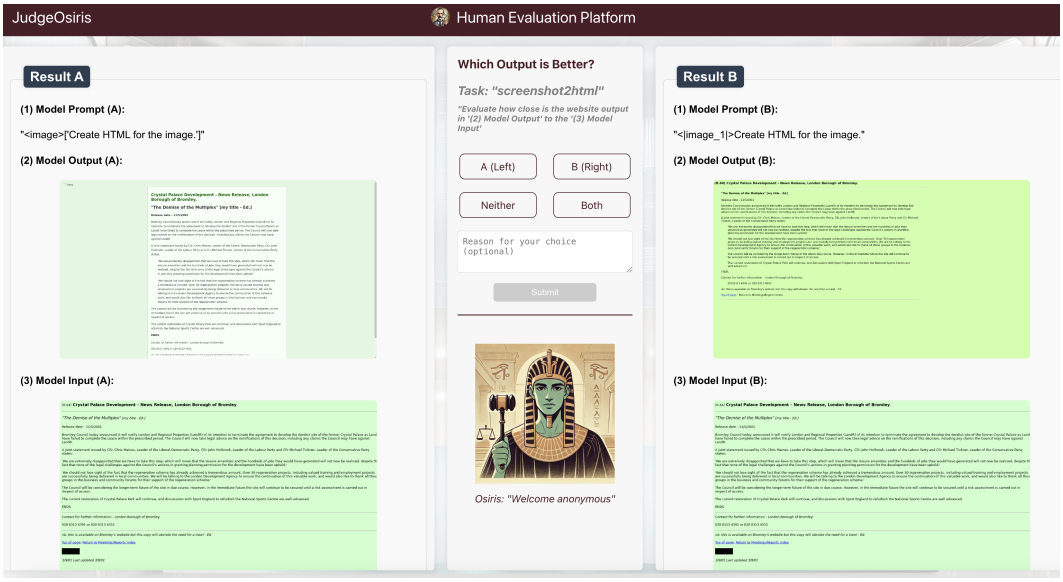


Figure 12: ***Human Evaluation Platform.*** *A user is presented with two model outputs and asked to select the one that is more accurate in relation to the original model input.*

Proposed metadata of a data sample from TabFact dataset (Chen et al., 2020)

metadata: features: {image_height: 1584, image_width: 1224}, license: {annotation_license: CC BY 4.0, image_license: CC BY-SA 4.0}, origin: {table_path: data/all_csv/2-15401676-3.html.csv, data_id: 2-15401676-3.html.csv, dataset_url: https://github.com/wenhuchen/Table-Fact-Checking.git, tracking_instruction_id: tabfact, tracking_transformation_id: tabfact_1}

tracking_instructions.json: tabfact: "Clone repo from dataset_url, locate CSV via table_path, render image."

tracking_transformations.json:

ID	Description
tabfact_1	Use table title as image caption.
tabfact_2	Concatenate annotation facts as summary.
tabfact_3	Load table using csv, convert to markdown.

Figure 13: ***Metadata framework*** *from our BigDocs Toolkit for a sample in the BigDocs 7.5M dataset. It includes details about image properties, licenses, data sources, and transformation steps, such as using the table title as an image caption.*

- `image_url` (optional): The URL from which the image was initially downloaded.
- `Misc.` (optional): Other information related to the origin of the data entry.

Features Metadata The `features` key provides detailed characteristics of the data sample. This can be useful for downstream tasks that require specific information about the sample, like the image size and type of annotations.

- `image_height`: The height of the image, in pixels.
- `image_width`: The width of the image, in pixels.

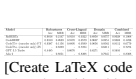
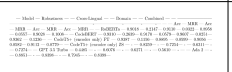
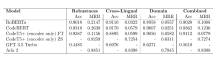
Common Metadata Files The *`tracking_instructions.json`* and *`tracking_transformations.json`* files are designed to store information common to many samples, minimizing redundancy and ensuring consistency. Each file is structured as a dictionary:

- `tracking_instructions.json`: Contains instructions on how to use the metadata fields (such as `data_id`, `dataset_url`, etc.) to trace the data sample back to its source. Each entry is uniquely identified and can be referenced in the individual sample metadata by `tracking_instruction_id`.

- `tracking_transformations.json`: Contains descriptions of any transformations, processing steps, or additions to the original samples. This includes synthetic generation processes, data augmentation steps, or other modifications. Each transformation is uniquely identified and can be referenced in the individual sample metadata by `tracking_transformation_id`.

A.11 Qualitative Results

Table 6: *Task Structure and Samples with Model Outputs for Phi Models and GPT-4o*

Task	Query	Phi-3.5-vision-instruct	Phi-3.5-vision-instruct-BigDocs	GPT-4o
GUI VQA	 [What opens when the user clicks [550, 0, 600, 80]?]	[The image does not provide information on what opens when the user clicks the specified coordinates.]	[View the news]	[By clicking on the coordinates [620, 50, 650, 70], the user will navigate to the "news" section of the website.]
Chart Captioning	 [I need a caption for the chart in the image, please]	[The image displays a pie chart with various segments, each representing a different range of values and their corresponding percentages. The segments are color-coded and labeled with both the value range and the percentage they represent. The chart is sourced from Statista 2024]	[In 2022, the most common salary range for employees in the United Kingdom was between 100,000 and 149,999 British pounds, with 19.3 percent of employees falling into this bracket. Employees earning between 200,000 and 249,999 British pounds made up the second most common salary range, at 15.8 percent.]	[Distribution of Income Levels: A Pie Chart Showing the Percentage of Households by Income Bracket in 2024]
Latex Table Generation	 [Create LaTeX code to produce this table.]			
HTML Generation	 [Could you create Hypertext Markup Language code from the screenshot?]			

A.12 Glossary & Task Definitions

We facilitate a list of concepts introduced and utilized throughout our research. Below is a list of these concepts and tasks, along with their definitions, including descriptions, inputs, and outputs.

1. **Text with Structures**
   - **Description**: Analyzing documents combining text with elements like headers, lists, and paragraphs to understand the overall structure and organization.
   - **Input**: Documents containing structured text elements such as headers, lists, and paragraphs.
   - **Output**: A structured representation of the document's content.
2. **Structures - Figures**
   - **Description**: Identifying and interpreting figures, diagrams, and other visual elements in documents to extract the information they represent.

- **Input**: Documents with embedded figures, diagrams, and visual elements.
- **Output**: A detailed understanding or extraction of the visual content.

3. **Infographics - Covers**
   - **Description**: Examining infographics and cover pages that use a mix of text and graphics to communicate information visually.
   - **Input**: Infographics and cover pages containing text and visual elements.
   - **Output**: A comprehensive analysis of the combined visual and textual information.
4. **Structures - Tables**
   - **Description**: Finding and understanding tables in documents, focusing on accurately capturing the data and relationships they present.
   - **Input**: Documents containing tabular data.
   - **Output**: A structured dataset or representation of the tabular information.
5. **Structures - Forms**
   - **Description**: Recognizing and processing forms within documents, extracting data from fields and labels typically found in structured forms.
   - **Input**: Documents with structured forms, including fields and labels.
   - **Output**: An organized collection of the extracted form data.
6. **Natural Image**
   - **Description**: Analyzing photographs or natural images in documents, interpreting their content in the context of the surrounding text.
   - **Input**: Documents containing photographs or natural images.
   - **Output**: An interpreted or categorized representation of the image content.