InstructDoc: A Dataset for Zero-Shot Generalization of Visual Document Understanding with Instructions

Ryota Tanaka[1,2], **Taichi Iki**[1], **Kyosuke Nishida**[1], **Kuniko Saito**[1], **Jun Suzuki**[2]
[1]NTT Human Informatics Laboratories, NTT Corporation
[2]Tohoku University
{ryota.tanaka, taichi.iki, kyosuke.nishida, kuniko.saito}@ntt.com, jun.suzuki@tohoku.ac.jp

Abstract

We study the problem of completing various visual document understanding (VDU) tasks, e.g., question answering and information extraction, on real-world documents through human-written instructions. To this end, we propose InstructDoc, the first large-scale collection of 30 publicly available VDU datasets, each with diverse instructions in a unified format, which covers a wide range of 12 tasks and includes open document types/formats. Furthermore, to enhance the generalization performance on VDU tasks, we design a new instruction-based document reading and understanding model, InstructDr, that connects document images, image encoders, and large language models (LLMs) through a trainable bridging module. Experiments demonstrate that InstructDr can effectively adapt to new VDU datasets, tasks, and domains via given instructions and outperforms existing multimodal LLMs and ChatGPT without specific training.

Introduction

Building document artificial intelligence (Document AI) capable of reading and comprehending real-world documents, including webpages, office documents, mobile UIs, etc., has been a long-cherished goal. Toward this goal, numerous works on visual document understanding (VDU) have addressed a wide range of tasks, such as document question answering (QA) (Mathew, Karatzas, and Jawahar 2021) and information extraction (Jaume, Ekenel, and Thiran 2019). Document data contain both textual and visual objects, with content spread structurally across various locations depending on diverse document types and formats. To address this complexity, previous works have proposed models that aim to improve interactions among text/layout/visual modalities (Xu et al. 2021; Appalaraju et al. 2021). However, the diversity of documents and tasks poses a challenge in developing a unified model that can comprehend intricate relationships between text and visual objects across a wide range of document types, formats, and tasks.

To improve the generalizability and adaptivity of unseen vision-language tasks, visual instruction tuning (Xu, Shen, and Huang 2023; Liu et al. 2023a) has been introduced. This approach involves training multimodal large language models (MLLMs) on a collection of images, task inputs, and instructions. However, according to (Liu et al. 2023b), most of the previous visual instruction tuning datasets have primarily focused on understanding visual (non-textual) objects in scene images and existing models struggle with accomplishing tasks that require visual document understanding abilities. While recent works (Zhang et al. 2023; Ye et al. 2023a) attempt to deal with the issue, they still exhibit limitations when generalizing to unseen tasks and documents.

In this paper, we propose **InstructDoc**[1], the first large-scale visual instruction tuning dataset that covers a wide range of VDU tasks and datasets (12 diverse tasks created from 30 openly available datasets). Each dataset has diverse instructions annotated by experts, following a unified instruction schema, composed of user's *intent* and *answer style*, for VDU tasks. As shown in Figure 1, InstructDoc requires a rich set of abilities, including understanding document layout, visual representations of texts, and relation extraction of objects (e.g., graphs and charts) over open document types/formats with handcrafted instructions.

Furthermore, to enhance the generalization performance on VDU tasks, we present a **Instruct**ion-based **D**ocument **r**eading and understanding model, InstructDr, which unifies the visual, text, and layout modalities of documents by bridging the gap between a vision encoder and a large language model (LLM) through a new bridging module called Document-former. The Document-former converts documents into a useful feature for the LLM. Experiments show that InstructDr achieves the highest zero-shot performance among existing MLLMs and outperforms ChatGPT on a wide range of VDU datasets with instructions.

Related Work

Visual document understanding. Visual documents are ubiquitous and used in diverse applications, including QA on business documents (Mathew, Karatzas, and Jawahar 2021), information extraction on receipts (Jaume, Ekenel, and Thiran 2019), and classification over large document collections (Harley, Ufkes, and Derpanis 2015). Due to this diversity, previous works have generally been domain/task-specific, lacking the sharing of underlying data, model architectures, and objectives (Xu et al. 2020; Appalaraju

[1]Our dataset and codes are publicly available at https://github.com/nttmdlab-nlp/InstructDoc

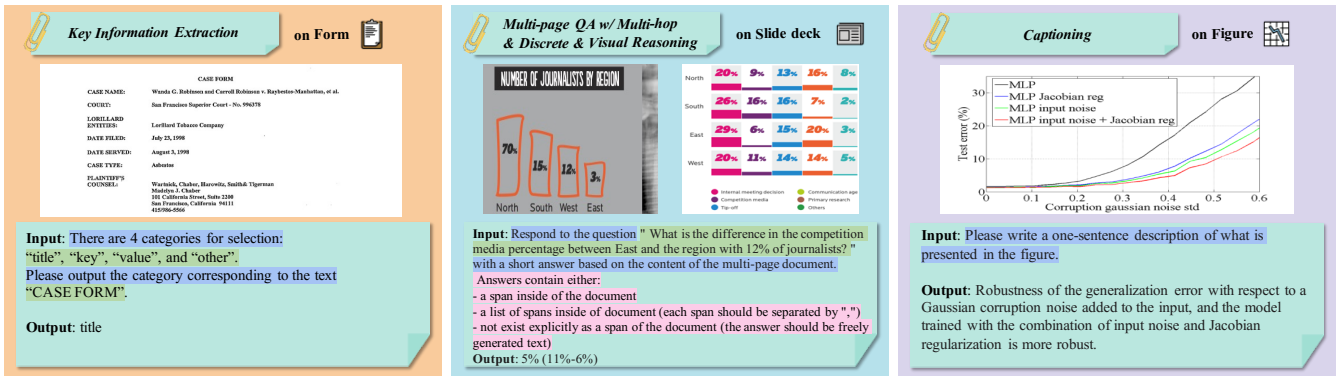


Figure 1: Examples of InstructDoc dataset. The input defines *intent* (blue), *query and options* (green), and *answer style* (pink). *query and options* and outputs are from original datasets.

et al. 2021; Huang et al. 2022). Although pixel-based methods (Kim et al. 2022; Lee et al. 2023) can simplify architectures, these methods have high computational costs (due to the encoding of high-resolution images) and can have degraded performance on new tasks. We leverage the reasoning abilities of LLMs and perform all VDU tasks in a unified sequence-to-sequence format with instructions, resulting in improved generalization performance.

Instruction-following language models. Training LLMs with instructions on various NLP tasks has proven effective in improving zero-shot performance of unseen tasks (Wei et al. 2021; Iyer et al. 2022). Flan (Wei et al. 2021; Longpre et al. 2023), PromptSource (Bach et al. 2022), and Natural Instructions (Mishra et al. 2022) collected instructions and datasets for a variety of general NLP tasks, such as machine reading comprehension and summarization tasks on plain-text documents. In contrast, we tackle the challenge of understanding real-world documents organized in non-plain text formats (e.g., HTML and PDF).

Visual instruction tuning. Researchers have recently explored the application of LLMs to vision-language tasks by distilling the output of LLMs (Liu et al. 2023a; Zhu et al. 2023; Ye et al. 2023b) or training with handcrafted instructions (Xu, Shen, and Huang 2023; Dai et al. 2023). However, as pointed out in (Liu et al. 2023b), these models struggle with tasks requiring document understanding abilities because they do not assume that text might be contained in images during instruction tuning. To mitigate this issue, LLaVAR (Zhang et al. 2023) and LLMDoc (Ye et al. 2023a) fine-tune MLLMs with instruction tuning on document images. However, these approaches have trouble understanding diverse real-world documents because (i) the datasets provide a few document and task types, hindering zero-shot generalization; and (ii) the models simply encode documents via vision encoders and cannot explicitly learn document meta-information (e.g., document layout). In contrast, the InstructDoc covers diverse VDU tasks and open document types/formats, and InstructDr learns rich representations of the underlying structure of documents with instructions.

InstructDoc Dataset

Problem Formulation

All of the tasks in InstructDoc are simply defined as: given an instruction T and a document image I, a model outputs an answer A. Each task is composed of one or more datasets, where the dataset $\mathcal{D}$ is associated with the set of K instructions $\mathcal{T}^{\mathcal{D}} = \{T_1^{\mathcal{D}}, ..., T_K^{\mathcal{D}}\}$ and contains N instances $\{(\mathcal{T}^{\mathcal{D}}, I_j, A_j)\}_{j=1}^{N}$. Here, we randomly select the instruction from $\mathcal{T}^{\mathcal{D}}$ for every instance. Note that we allow the utilization of external OCR engines to derive the answer in our setting, as in the previous VDU benchmark (Borchmann et al. 2021). Our goal is to enable the model to perform a wide range of VDU tasks with instructions rather than improving the accuracy of text recognition (Zhang et al. 2023).

We mainly evaluate the models' ability to perform zero-shot learning scenarios. Specifically, we fine-tune a model on a collection of instruction tasks and evaluate it on unseen datasets defined three types: (i) $\textbf{Test}_{\textbf{Cross-Dataset}}$: datasets not used during training, but whose tasks exist in training set; (ii) $\textbf{Test}_{\textbf{Cross-Task}}$: datasets and associated tasks entirely unseen during training; and (iii) $\textbf{Test}_{\textbf{Cross-Domain}}$: datasets, tasks, and document types entirely unseen during training.

Dataset Collection

In this section, we describe the collection process of the InstructDoc dataset. InstructDoc is designed to cover a wide range of VDU tasks with instructions that require reasoning among document layout, images, and text.

Source dataset collection. Figure 2 shows the source datasets in InstructDoc. We collected 30 publicly available datasets and 12 tasks in VDU areas from DUE (Borchmann et al. 2021) as well as through manual searches. Following the task clusters defined in previous works (Wei et al. 2021; Dai et al. 2023), we divided the QA datasets that require different reasoning abilities into different tasks. As a result, we divided the collected datasets into the following tasks:

- **Key Information Extraction (KIE)** assigns each word a semantic entity label from predefined categories (Šimsa et al. 2023; Jaume, Ekenel, and Thiran 2019; Sun et al. 2021; Park et al. 2019; Huang et al. 2019).

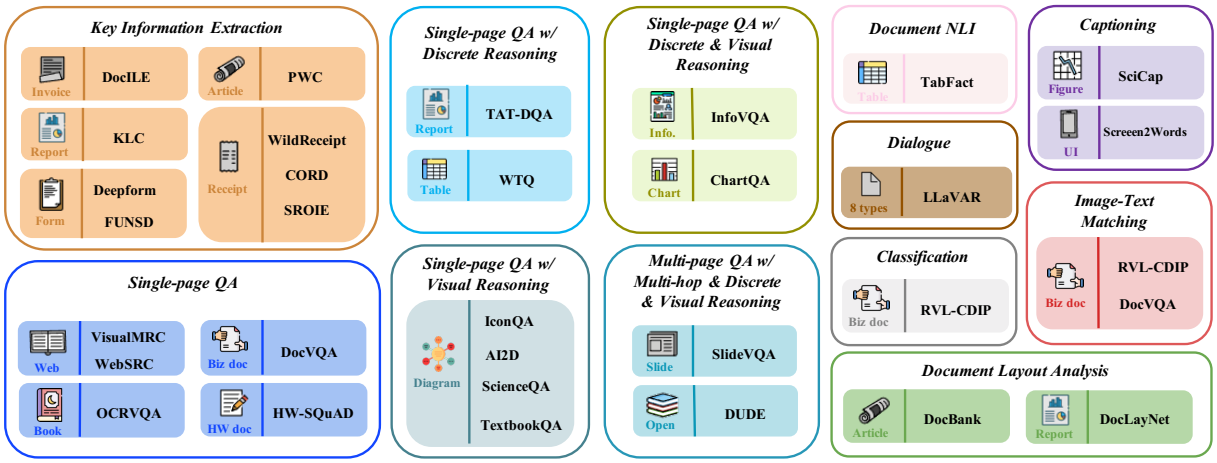


Figure 2: Datasets used in InstructDoc. InstructDoc covers a wide range of VDU tasks and document types and formats.

- **Single-page QA** is a task of QA on single-page documents and focuses on document layout and textual content understanding (Tanaka, Nishida, and Yoshida 2021; Chen et al. 2021; Mishra et al. 2019; Mathew, Karatzas, and Jawahar 2021; Tüselmann et al. 2022).
- **Single-page QA w/ Discrete Reasoning** requires various arithmetic abilities, including addition, sorting, or counting (Zhu et al. 2022).
- **Single-page QA w/ Visual Reasoning** requires a set of abilities, including object (e.g., icon) recognition, commonsense understanding, and relation extraction on single-page documents (Lu et al. 2021; Kembhavi et al. 2016; Lu et al. 2022; Kembhavi et al. 2016).
- **Single-page QA w/ Discrete & Visual Reasoning** requires both discrete and visual reasoning (Mathew et al. 2022; Masry et al. 2022) on single-page documents.
- **Multi-page QA w/ Multi-hop & Discrete & Visual Reasoning** requires understanding the content relationship via multi-hop reasoning as well as discrete/visual reasoning on multi-page documents (Tanaka et al. 2023; Landeghem et al. 2023).
- **Document NLI** is a task of natural language inference that predicts the entailment relationship between two sentences in a document (Borchmann et al. 2021)
- **Dialogue** involves a human-agent interaction on the basis of document images (Zhang et al. 2023).
- **Captioning** involves producing descriptions of documents (Hsu, Giles, and Huang 2021; Wang et al. 2021).
- **Classification** involves classifying a document from a set of candidate labels (Harley, Ufkes, and Derpanis 2015).
- **Document Layout Analysis (DLA)** determines a document's components with bounding boxes (Li et al. 2020; Pfitzmann et al. 2022)
- **Image-Text Matching (ITM)** requires the model to determine whether a given OCR text and image match.

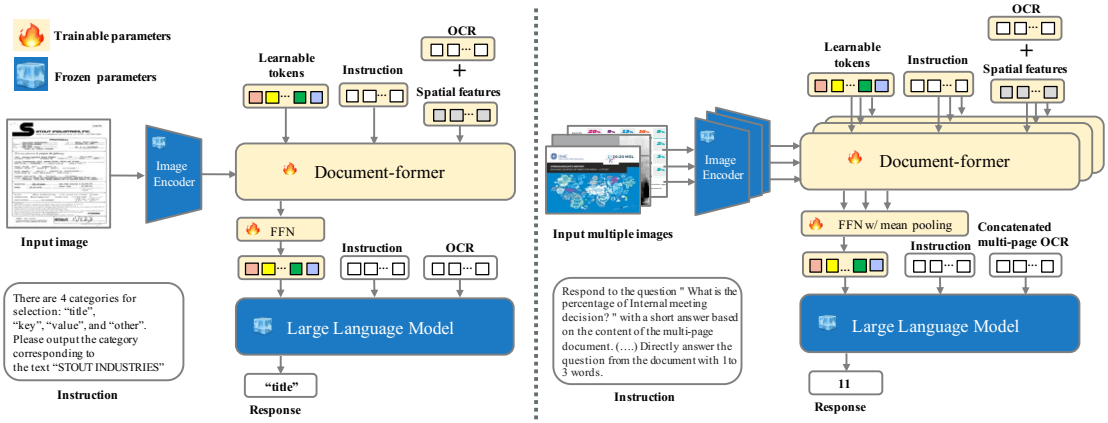
Query rephrasing. We found that two KIE datasets (FUNSD and CORD) are challenging because they contain abbreviated queries that are difficult for humans to comprehend. To bridge the gap between humans and machines, we replace these queries with complete and more easily understandable phrases (e.g., `menu.vatyn` → `menu_whether_price_tax_included`).

Instruction annotation. For each dataset, we manually crafted five to ten distinct instruction templates in a unified format. For QA tasks, the answers have diverse styles in the original datasets; for example, DocVQA's answer is extractive, which requires the model to extract a contiguous span of words from the document, but VisualMRC's answer is generative, which is not limited to the word spans. Hence, an instruction that sufficiently describes an arbitrary VDU task should include *intent* and *answer style* or only *intent*. Specifically, as shown in Figure 1, *intent* describes how the task can be performed and *answer style* describes how the model generates the output. If each dataset provides *query and options*, we fill it in annotated instruction templates.

Data split. We split InstructDoc into 23 held-in and seven held-out datasets. For the held-out evaluation, we aim to understand how instruction tuning on the held-in datasets improves the zero-shot generalization performance on unseen datasets, including (i) $\textbf{Test}_{\textbf{Cross-Dataset}}$: FUNSD and CORD datasets, (ii) $\textbf{Test}_{\textbf{Cross-Task}}$: ChartQA, InfoVQA, and TabFact datasets, and (iii) $\textbf{Test}_{\textbf{Cross-Domain}}$: DUDE and SlideVQA datasets. All other datasets were held-in ones to train our model. Note that the held-out datasets were carefully selected in order to avoid data contamination.

Comparison with Related Datasets

Table 1 shows the statistics of InstructDoc and other VDU instruction tuning datasets, including LLaVAR (Zhang et al. 2023) and DocOwl (Ye et al. 2023a). InstructDoc has three unique key properties; First, it is the first dataset to address open document types, including multi-page documents and has the highest standard deviation in the number of OCR tokens (1442.8) compared with LLaVAR (93.1) and DocOwl



(a) Our InstructDr for a single-page document

(b) Our InstructDr for a multi-page document

Figure 3: InstructDr model. We update only the parameters of Document-former and the projection FFN layer during training.

	LLaVAR	DocOwl	InstructDoc
Both Single/Multi-page docs			✓
Instruction annotation		✓	✓
Answer style annotation			✓
#Document types	8	7	Open
#Seed datasets	1	8	30
#Task clusters	1	3	12
#Avg.$_{\pm\text{Std.}}$ IT words	-	$5_{\pm 0}$	$20.3_{\pm 11.2}$
#Avg.$_{\pm\text{Std.}}$ IT	-	$1_{\pm 0}$	$7.4_{\pm 2.4}$
#Avg.$_{\pm\text{Std.}}$ OCR words	$52.5_{\pm 93.1}$	$270.1_{\pm 807.2}$	$443.2_{\pm 1442.8}$
#Avg.$_{\pm\text{Std.}}$ Answer words	$34.5_{\pm 27.5}$	$1.9_{\pm 2.7}$	$5.88_{\pm 17.7}$

Table 1: Statistics of InstructDoc and other VDU instruction tuning datasets. We excluded data other than the VDU tasks from DocOwl. IT denotes instruction templates.

(807.2). This implies that our dataset is a more challenging setting. Second, InstructDoc covers the widest range of tasks, offering four times more tasks compared with DocOwl, while LLaVAR provides only a single task. Finally, InstructDoc provides a more extensive set of instructions (20.3 words and 7.4 templates) and annotates various answer styles within the instructions to deal with various VDU tasks that require diverse abilities. In contrast, the instructions in DocOwl are limited (five words and a single template) and LLaVAR has only machine-generated instructions, and they may not generalize well to reformulations and new tasks.

Our Model

Figure 3 depicts our model, InstructDr (**Instruct**ion-based **D**ument **r**eading and understanding model). We use pre-trained BLIP-2 (Li et al. 2023), a state-of-the-art MLLM connected with instruction-tuned FlanT5 (Chung et al. 2022), as the base model. We extend BLIP-2 in three key ways; (i) equipping it with Document-former, an enhanced Q-former module that can capture and convert the visual and textual content/layout of documents into representations of the LLM, (ii) conducting multi-task instruction tuning with unified formats, and (iii) encoding multiple images in parallel to facilitate understanding of multi-page documents.

Spatial-aware Document Feature Extraction

Document image/OCR and instruction encoding. To encode a document image, we use a pre-trained CLIP (Radford et al. 2021) vision encoder to extract its visual features $\mathbf{z}^{\text{vis}}$. Additionally, we process the document image using an OCR engine and apply a sub-word tokenizer to obtain M word tokens $\{s_i\}_{i=1}^{M}$ and their corresponding bounding boxes $\{(x_i^1, y_i^1, x_i^2, y_i^2)\}_{i=1}^{M}$, where (x^1, y^1) and (x^2, y^2) represent the coordinates of the top-left and bottom-right corners, respectively. To learn the visual layout of the image, we construct a spatially aware OCR representation $\mathbf{z}_i^{\text{ocr}} = \mathbf{z}_i^{\text{word}} + \mathbf{z}_i^{\text{bbox}}$ with learnable embedding layers $\mathbf{W}^{\{s,x,y,h,w\}}$, where OCR text features are calculated as $\mathbf{z}_i^{\text{word}} = \mathbf{W}^s(s_i)$ and spatial features are calculated as $\mathbf{z}_i^{\text{bbox}} = \mathbf{W}^x(x_i^1, x_i^2) + \mathbf{W}^y(y_i^1, y_i^2) + \mathbf{W}^h(y_i^2 - y_i^1) + \mathbf{W}^w(x_i^2 - x_i^1)$. Similarly, we encode an instruction by $\mathbf{W}^s$ and obtain its features $\mathbf{z}^{\text{ins}}$.

Document-former. We introduce Document-former, which is a trainable module to bridge the gap between an image encoder and an LLM, enabling extraction of document content/layout that LLMs can understand. The architecture of Document-former is a stack of Transformer blocks with cross-attention layers. To map document features into the LLM's space, we use a set of m learnable tokens $\mathbf{z}^{\text{token}} \in \mathbb{R}^d$, where d is the dimension of the hidden size. These tokens $\mathbf{z}^{\text{token}}$ interact with $\mathbf{z}^{\text{vis}}$ through cross-attention layers and with the input sequence, composed of $\mathbf{z}^{\text{ins}}$ and $\mathbf{z}^{\text{ocr}}$, through self-attention layers. As a result, we obtain $\mathbf{z}^{\text{doc}}$ and transform it via a projection feed-forward network (FFN) layer to $\mathbf{h}^{\text{doc}} \in \mathbb{R}^{m \times d^{\text{LLM}}}$, which have the same dimension d^{LLM} as the LLM's input embedding.

Multimodal Document Large Language Model

Connecting document features to LLM. The LLM receives the document embeddings $\mathbf{h}^{\text{doc}}$, the instruction, and

Model	Modal	#TuP	#ToP	Cross-Dataset		Cross-Task			Cross-Domain		Held-out
				FUNSD eF1/F1	CORD eF1/F1	ChartQA RAcc./F1	InfoVQA ANLS/F1	TabFact Acc./F1	DUDE ANLS/F1	SlideVQA EM/F1	Avg.
LLMDoc	V	388M	7B	-/-	-/-	-/-	38.2†/-	60.2†/-	-/-	-/-	-/-
LLaVA	TV	13B	13B	12.0/1.3	0.2/ 5.1	0.0/1.7	3.4/3.5	0.0/0.0	6.5/5.9	0.0/2.3	3.1/2.8
LLaVAR	TV	13B	13B	12.0/2.0	0.1/10.8	0.0/3.0	6.2/4.6	0.0/2.1	8.1/5.1	0.0/6.2	3.8/4.8
MiniGPT-4	TV	3.1M	7B	12.0/2.2	0.2/ 2.1	0.0/0.4	4.3/0.5	0.3/0.2	5.9/1.1	0.0/0.4	3.2/1.0
mPLUG-Owl	TV	388M	7B	12.0/6.7	0.2/15.0	0.0/0.3	5.6/5.3	0.0/2.6	5.8/5.5	0.0/0.4	3.4/5.1
InstructBLIP	TV	103M	3.4B	16.8/15.0	4.9/9.5	3.3/7.2	8.7/7.3	33.6/33.7	11.0/8.8	5.2/9.0	11.9/12.9
BLIP-2	TV	103M	3.4B	19.6/19.6	32.0/51.9	23.6/21.5	48.2/36.7	58.6/58.6	39.8/35.4	28.3/38.8	35.7/37.5
BLIP-2 trained on IDoc	TV	103M	3.4B	26.0/26.1	33.8/54.7	24.7/21.2	47.8/35.4	58.8/58.8	43.9/40.4	30.1/38.8	37.9/39.3
InstructDr (Ours)	TLV	103.1M	3.4B	**38.2/38.1**	**46.0/62.7**	**29.4/22.3**	**50.9/37.6**	**59.4/59.4**	**45.2/41.6**	**31.9/40.2**	**43.0/43.1**

Table 2: Zero-shot performance of InstructDr and MLLMs on VDU tasks. "T/L/V" denotes the "text/layout/visual" modality of documents. #TuP/#ToP denotes the number of tuning/total parameters. The highest zero-shot performances are marked in bold. †denotes the supervised performance reported in the original paper, as it is not publicly available. IDoc denotes InstructDoc.

OCR tokens as input and outputs the answer $\mathbf{A}$, token by token. The parameters of the LLM are initialized from an instruction-tuned FlanT5.

Parameter-efficient multi-task instruction tuning. To achieve task-agnostic learning, we formulate the process of learning all held-in tasks in a unified sequence-to-sequence abstraction through instructions. To train the LLM efficiently, we update only the parameters of the Document-former (including $\mathbf{W}^{\{s,x,y,h,w\}}$) and the projection FFN layer, while keeping other parameters frozen during training. We optimize the model by minimizing the negative log-likelihood between the ground-truth and predictions.

Multi-page document understanding. We also support performing reasoning across multiple document pages. As shown in Figure 3b, each image is processed individually by the image encoder and Document-former, and their resulting document embeddings are mean-pooled together before being fed into the LLM. The OCR input to the LLM consists of concatenated tokens extracted from each page.

Experiments

Experimental Setup

We mainly conducted evaluations under three zero-shot settings, including $\textbf{Test}_{\textbf{Cross-Dataset}}$, $\textbf{Test}_{\textbf{Cross-Task}}$, and $\textbf{Test}_{\textbf{Cross-Domain}}$. Furthermore, we evaluated our model under the task-specific fine-tuning setting.

Baselines. We compared InstructDr with seven state-of-the-art (SOTA) MLLMs, including **LLaVA** (Liu et al. 2023a), **MiniGPT-4** (Zhu et al. 2023) and **mPLUG-Owl** (Ye et al. 2023b), which align CLIP visual encoder with Vicuna (Chiang et al. 2023) trained on a dialogue generated by GPT-4 (OpenAI 2023); **BLIP-2** (Li et al. 2023), which connects a FlanT5 with a vision encoder; **InstructBLIP** (Dai et al. 2023), which fine-tunes BLIP-2 with instructions on scene images; and **LLMDoc** (Ye et al. 2023a) and **LLaVAR** (Zhang et al. 2023), which fine-tune mPULG-Owl/LLaVA on the DocOwl/LLaVAR datasets. Additionally, we used **Supervised SOTA models** (Appalaraju et al. 2023; Chen et al. 2023; Huang et al. 2022; Landeghem et al. 2023) on each dataset and two text-based LLMs, **ChatGPT** (`gpt-3.5-turbo-0613`) and **GPT-4**. To control the answer's length, we added control phrases (e.g., *use 1 to 3 words to answer*) to the selected instructions.

Evaluation metrics. We followed the evaluation protocol of each dataset, we used **ANLS** (Biten et al. 2019) for InfoVQA, DUDE, Text-VQA and ST-VQA, **EM** for SlideVQA, Relaxed Accuracy (**RAcc.**) for ChartQA, entity F1 (**eF1**) for FUNSD and CORD, Accuracy (**Acc.**) for TabFact, and **ROUGE-L** for VisualMRC as evaluation metrics. Additionally, we used **F1** as the optional metrics.

Implementation details. Following (Wei et al. 2021), we balanced the training instances of different tasks by sampling a maximum of 5k instances for each held-in dataset while keeping all evaluation instances. We used the AdamW (Loshchilov and Hutter 2017) with a weight decay of 0.05. We applied a linear warmup during the initial 1,000 steps and used a cosine learning rate decay with a minimum learning rate of 0. We set the number of learnable tokens m to 32. All images of the model input were resized to 224. We trained on eight A100 (40G) GPUs for three epochs and completed the training within two hours. If each dataset does not provide OCR, we extracted it via the Google Vision API.

Experimental Results and Analysis

Does our model outperform existing MLLMs? Table 2 shows that our model achieved the highest performance on all datasets compared with other MLLMs. InstructDr consistently outperformed its original backbone, BLIP-2, by a significant margin, indicating that instruction tuning on InstructDoc effectively enhances performance on unseen VDU datasets, tasks, and domains. In contrast, InstructBLIP, which is instruction-tuned BLIP-2 trained on scene images, performed worse than BLIP-2. This is because that InstructBLIP does not assume that the images might contain text during instruction tuning. BLIP-2 fine-tuned on InstructDoc falls short of achieving the same level of performance compared with InstructDr, indicating that InstructDr is bet-

Model	Modal	Cross-Dataset		Cross-Task			Cross-Domain		Held-out
		FUNSD eF1/F1	CORD eF1/F1	ChartQA RAcc./F1	InfoVQA ANLS/F1	TabFact Acc./F1	DUDE ANLS/F1	SlideVQA EM/F1	Avg.
Supervised SOTA models	TLV	92.1/-	97.7/-	72.3/-	54.8*/-	83.2*/-	46.1*/-	33.5/41.7	-/-
ChatGPT	T	21.8/21.2	30.4/49.3	16.0/16.8	37.8/29.5	52.5/52.4	34.5/32.3	11.7/23.8	29.2/32.2
GPT-4	T	47.5/47.5	69.4/81.7	20.9/27.6	49.9/46.5	68.8/68.8	46.3/45.1	21.0/36.4	46.3/50.5
InstructDr (Ours)	TLV	38.2/38.1	46.0/62.7	29.4/22.3	50.9/37.6	59.4/59.4	45.2/ 41.6	31.9/40.2	43.0/43.1

Table 3: Zero-shot performance on VDU tasks of InstructDr and supervised SOTA models and powerful text-based LLMs. * denotes the performance on different splits we used because they evaluated on the leaderboard and F1 cannot be used.

Model	CORD eF1	TabFact Acc.	DUDE ANLS	Held-out Avg.
InstructDr	**46.0**	**59.4**	**45.2**	**43.0**
w/o Document-former	38.5	58.8	44.6	40.2
w/o Spatially OCR features	33.8	58.8	43.9	37.9
w/o Mean pooling (concat.)	-	-	43.8	-
w/o Instructions in test	24.0	4.0	38.9	28.0
w/o Instructions in train	17.3	58.2	34.0	28.9
w/o Instructions in both	0.4	3.7	24.4	21.3
w/o Query rephrasing	30.9	-	-	-
w/o Answer style annotation	-	-	44.2	-

Table 4: Ablation study of the architecture and instructions. We report the scores when the ablation can be conducted.

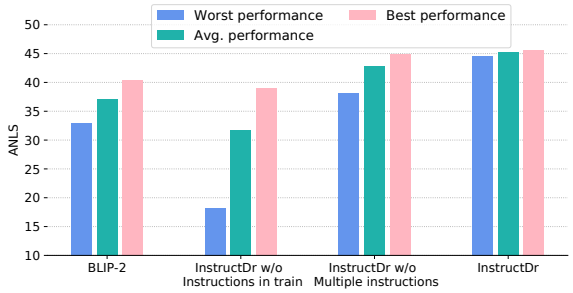


Figure 4: Comparison of zero-shot performance on DUDE for five different instructions. w/o Multiple instructions denotes our model trained with a single instruction per dataset.

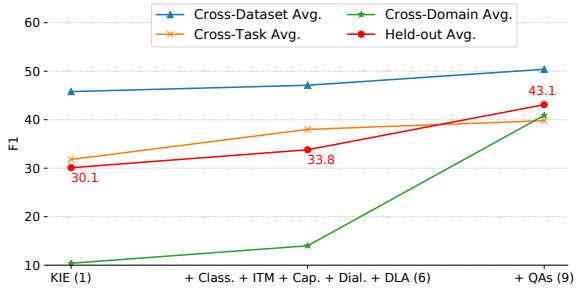


Figure 5: Model performance as the number of task clusters used in training. (·) denotes the number of tasks.

ter suited for comprehending diverse real-world documents. This conclusion is further supported by the results presented in Table 4, where ablations of Document-former, spatial information, and strategy of gathering multi-page features have a significant negative impact on performance.

How well does our model perform in comparison with supervised SOTA models and powerful LLMs? As shown in Table 3, our model outperformed ChatGPT on all datasets. Additionally, InstructDr achieved competitive results with supervised SOTA models and GPT-4 on the DUDE and SlideVQA datasets that require multiple reasoning skills (e.g., discrete, visual, and multi-hop reasoning). This indicates that our model can effectively learn diverse skills through instruction tuning with InstructDoc.

What is the role of instructions? As shown in Table 4, removing instructions (i.e., only *query and options* as the model input) significantly decreased zero-shot performance during training or/and test time, indicating the effectiveness of incorporating instructions. This result was observed on the high-quality instruction-tuning datasets (Wei et al. 2021; Xu, Shen, and Huang 2023). Moreover, our instruction annotations, including query rephrasing and answer styles, helped to improve the zero-shot performance.

Does our model have robustness towards diverse instructions? Figure 4 shows the performance variance when the models were given five different instructions; InstructDr exhibited the smallest performance variance and outperformed the other models. This indicates InstructDoc empowers the model with the ability to deal with a variety of instructions. Our results also suggest that using multiple instructions per dataset is important for achieving decent performance.

What is the impact of diverse task clusters? As shown in Figure 5, as the number of task clusters increases, we can observe an improvement in models' zero-shot performance.

Are our model weights effective for task-specific fine-tuning? We further fine-tuned InstructDr (only Document-former module) on a specific dataset to investigate the knowledge and transferability of our instruction-tuned model weights. Table 5 shows the fine-tuning performance on held-in (VisualMRC) and held-out (DUDE, SlideVQA) tasks. InstructDr achieved state-of-the-art finetuning

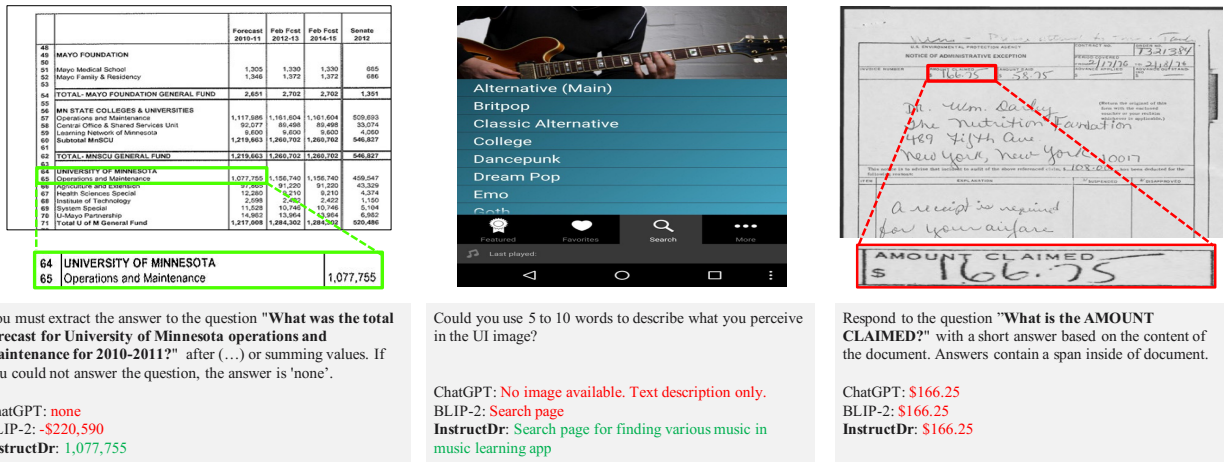


Figure 6: Qualitative examples. Outputs are correct (green) and incorrect (red) answers. (...) denotes ellipsis.

Model	VisualMRC ROUGE-L	DUDE ANLS	SlideVQA EM	SlideVQA F1
Supervised SOTA models	52.2	46.1	33.5	41.7
BLIP-2	60.5	45.6	36.9	46.5
InstructDr	**61.1**	**46.8**	**37.7**	**47.3**

Table 5: Fine-tuning performance in held-in (VisualMRC) and held-out (DUDE, SlideVQA) tasks on the test set.

Model	Image type in instruction tuning	TextVQA Acc.	TextVQA ANLS	ST-VQA ANLS
BLIP-2	-	48.7	64.8	39.1
InstructBLIP	Daily scene	52.8	67.3	**45.7**
InstructDr	Documents	**53.8**	**68.1**	43.3

Table 6: Zero-shot performance of scene-text VQA task.

performance on VisualMRC, DUDE, and SlideVQA using a unified model. Compared with BLIP-2, InstructDr exhibited superior fine-tuning performance on both held-in/out datasets, validating InstructDr as a better weight initialization model for task-specific fine-tuning.

Can our model also understand images other than documents? Table 6 shows the zero-shot performance of scene-text VQA (Singh et al. 2019; Biten et al. 2019) on scene images, which are the unseen image types in InstructDoc but were used for training our base model, BLIP-2. Note that ST-VQA's images include the part of COCO (Lin et al. 2014) that InstructBLIP was trained on. This result indicates that InstructDr can effectively learn visual reasoning skills without forgetting the abilities of the original backbone.

Qualitative examples. Figure 6 visualizes output examples, where the left/center/right examples require table/visual/hand-written text understanding skills. ChatGPT gave incorrect answers because it can only consider text information. Moreover, while BLIP-2 could not follow instructions (e.g., *use 5 to 10 words*) and extract items from structured text, InstructDr accomplished diverse VDU tasks with instructions. As shown in the right example, all models affected OCR quality, causing incorrect answers.

Limitations

Despite its impressive performance on various VDU tasks with instructions, InstructDr suffers from noisy OCR predictions, whose performance depends highly on OCR text qualities (right of Figure 6). We argue that our approach is more cost-efficient and accurate because another approach, the pixel-based ones (Kim et al. 2022; Chen et al. 2023), requires a large amount of computation to encode high-resolution images and cannot use document meta-information (e.g., bounding boxes). Moreover, since InstructDoc only contains a single document-text pair per instance, it cannot learn the correlation among multiple document-text pairs and lacks an in-context learning capability. The same observation has also been reported in the Flamingo (Alayrac et al. 2022) and BLIP-2. Finally, while we have constructed diverse VDU tasks, the number of tasks and corresponding instructions are still limited. We plan to consider utilizing automatic generation and augmentation techniques to increase the variety of instructions available.

Conclusion

We introduced a new large-scale instruction-tuning dataset, InstructDoc, to lay the foundation for building general-purpose VDU models that can follow natural language instructions. We also introduced a simple yet effective instruction tuning model, InstructDr, which unifies the vision, text, and layout modalities of documents by bridging the gap between a vision encoder and an LLM with Document-former. We performed a comprehensive study on instruction tuning with InstructDoc and demonstrated its generalization capability to a wide range of VDU datasets, tasks, and domains with instructions. We believe that our dataset will facilitate research on developing general-purpose document artificial intelligence systems.

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. In *NeurIPS*.

Appalaraju, S.; Jasani, B.; Kota, B. U.; Xie, Y.; and Manmatha, R. 2021. Docformer: End-to-End Transformer for Document Understanding. In *CVPR*, 993–1003.

Appalaraju, S.; Tang, P.; Dong, Q.; Sankaran, N.; Zhou, Y.; and Manmatha, R. 2023. DocFormerv2: Local Features for Document Understanding. *arXiv:2306.01733*.

Bach, S.; Sanh, V.; Yong, Z. X.; Webson, A.; Raffel, C.; Nayak, N. V.; Sharma, A.; Kim, T.; Bari, M. S.; Fevry, T.; Alyafeai, Z.; Dey, M.; Santilli, A.; Sun, Z.; Ben-david, S.; Xu, C.; Chhablani, G.; Wang, H.; Fries, J.; Al-shaibani, M.; Sharma, S.; Thakker, U.; Almubarak, K.; Tang, X.; Radev, D.; Jiang, M. T.-j.; and Rush, A. 2022. PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts. In *ACL-demo*, 93–104.

Biten, A. F.; Tito, R.; Mafla, A.; i Bigorda, L. G.; Rusiñol, M.; Jawahar, C. V.; Valveny, E.; and Karatzas, D. 2019. Scene Text Visual Question Answering. In *ICCV*, 4290–4300.

Borchmann, Ł.; Pietruszka, M.; Stanislawek, T.; Jurkiewicz, D.; Turski, M.; Szyndler, K.; and Graliński, F. 2021. DUE: End-to-End Document Understanding Benchmark. In *NeurIPS*.

Chen, X.; Djolonga, J.; Padlewski, P.; Mustafa, B.; Changpinyo, S.; Wu, J.; Ruiz, C. R.; Goodman, S.; Wang, X.; Tay, Y.; et al. 2023. PaLI-X: On Scaling up a Multilingual Vision and Language Model. *arXiv:2305.18565*.

Chen, X.; Zhao, Z.; Chen, L.; Ji, J.; Zhang, D.; Luo, A.; Xiong, Y.; and Yu, K. 2021. WebSRC: A Dataset for Web-Based Structural Reading Comprehension. In *EMNLP*, 4173–4185.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, E.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2022. Scaling Instruction-Finetuned Language Models. *arXiv:2210.11416*.

Dai, W.; Li, J.; Li, D.; Tiong, A. M. H.; Zhao, J.; Wang, W.; Li, B.; Fung, P.; and Hoi, S. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. *arXiv:2305.06500*.

Harley, A. W.; Ufkes, A.; and Derpanis, K. G. 2015. Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval. In *ICDAR*, 991–995.

Hsu, T.-Y.; Giles, C. L.; and Huang, T.-H. 2021. SciCap: Generating Captions for Scientific Figures. In *EMNLP Findings*, 3258–3264.

Huang, Y.; Lv, T.; Cui, L.; Lu, Y.; and Wei, F. 2022. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In *ACMM*, 4083–4091.

Huang, Z.; Chen, K.; He, J.; Bai, X.; Karatzas, D.; Lu, S.; and Jawahar, C. 2019. ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction. In *ICDAR*, 1516–1520.

Iyer, S.; Lin, X. V.; Pasunuru, R.; Mihaylov, T.; Simig, D.; Yu, P.; Shuster, K.; Wang, T.; Liu, Q.; Koura, P. S.; et al. 2022. OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization. *arXiv:2212.12017*.

Jaume, G.; Ekenel, H. K.; and Thiran, J.-P. 2019. FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. In *ICDARW*.

Kembhavi, A.; Salvato, M.; Kolve, E.; Seo, M.; Hajishirzi, H.; and Farhadi, A. 2016. A Diagram Is Worth A Dozen Images. In *ECCV*, 235–251.

Kim, G.; Hong, T.; Yim, M.; Nam, J.; Park, J.; Yim, J.; Hwang, W.; Yun, S.; Han, D.; and Park, S. 2022. OCR-free Document Understanding Transformer. In *ECCV*, 498–517.

Landeghem, J.; Tito, R.; Borchmann, Ł.; Pietruszka, M.; Józiak, P.; Powalski, R.; Jurkiewicz, D.; Coustaty, M.; Ackaert, B.; Valveny, E.; et al. 2023. Document Understanding Dataset and Evaluation (DUDE). *arXiv:2305.08455*.

Lee, K.; Joshi, M.; Turc, I. R.; Hu, H.; Liu, F.; Eisenschlos, J. M.; Khandelwal, U.; Shaw, P.; Chang, M.-W.; and Toutanova, K. 2023. Pix2Struct: Screenshot Parsing as Pretraining for Visual Language Understanding. In *ICML*, 18893–18912.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *ICML*.

Li, M.; Xu, Y.; Cui, L.; Huang, S.; Wei, F.; Li, Z.; and Zhou, M. 2020. DocBank: A Benchmark Dataset for Document Layout Analysis. In *COLING*, 949–960.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, 740–755.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023a. Visual Instruction Tuning. *arXiv:2304.08485*.

Liu, Y.; Li, Z.; Li, H.; Yu, W.; Huang, M.; Peng, D.; Liu, M.; Chen, M.; Li, C.; Jin, L.; et al. 2023b. On the Hidden Mystery of OCR in Large Multimodal Models. *arXiv:2305.07895*.

Longpre, S.; Hou, L.; Vu, T.; Webson, A.; Chung, H. W.; Tay, Y.; Zhou, D.; Le, Q. V.; Zoph, B.; Wei, J.; et al. 2023. The FLAN collection: Designing Data and Methods for Effective Instruction Tuning. *arXiv:2301.13688*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled Weight Decay Regularization. *arXiv:1711.05101*.

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In *NeurIPS*.

Lu, P.; Qiu, L.; Chen, J.; Xia, T.; Zhao, Y.; Zhang, W.; Yu, Z.; Liang, X.; and Zhu, S.-C. 2021. IconQA: A New Benchmark for Abstract Diagram Understanding and Visual Language Reasoning. In *NeurIPS*.

Masry, A.; Do, X. L.; Tan, J. Q.; Joty, S.; and Hoque, E. 2022. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *ACL Findings*, 2263–2279.

Mathew, M.; Bagal, V.; Tito, R.; Karatzas, D.; Valveny, E.; and Jawahar, C. 2022. InfographicVQA. In *WACV*, 1697–1706.

Mathew, M.; Karatzas, D.; and Jawahar, C. V. 2021. DocVQA: A Dataset for VQA on Document Images. In *WACV*, 2200–2209.

Mishra, A.; Shekhar, S.; Singh, A. K.; and Chakraborty, A. 2019. OCR-VQA: Visual Question Answering by Reading Text in Images. In *ICDAR*, 947–952.

Mishra, S.; Khashabi, D.; Baral, C.; and Hajishirzi, H. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *ACL*, 3470–3487.

OpenAI. 2023. GPT-4 Technical Report. *arXiv:2303.08774*.

Park, S.; Shin, S.; Lee, B.; Lee, J.; Surh, J.; Seo, M.; and Lee, H. 2019. CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In *Workshop on Document Intelligence at NeurIPS*.

Pfitzmann, B.; Auer, C.; Dolfi, M.; Nassar, A. S.; and Staar, P. 2022. DocLayNet: A Large Human-Annotated Dataset for Document-Layout Segmentation. In *KDD*, 3743–3751.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *ICML*, 8748–8763.

Šimsa, Š.; Šulc, M.; Uřičář, M.; Patel, Y.; Hamdi, A.; Kocián, M.; Skalickỳ, M.; Matas, J.; Doucet, A.; Coustaty, M.; and Karatzas, D. 2023. DocILE Benchmark for Document Information Localization and Extraction. *arXiv:2302.05658*.

Singh, A.; Natarajan, V.; Shah, M.; Jiang, Y.; Chen, X.; Batra, D.; Parikh, D.; and Rohrbach, M. 2019. Towards VQA Models That Can Read. In *CVPR*, 8317–8326.

Sun, H.; Kuang, Z.; Yue, X.; Lin, C.; and Zhang, W. 2021. Spatial Dual-Modality Graph Reasoning for Key Information Extraction. *arXiv:2103.14470*.

Tanaka, R.; Nishida, K.; Nishida, K.; Hasegawa, T.; Saito, I.; and Saito, K. 2023. SlideVQA: A Dataset for Document Visual Question Answering on Multiple Images. In *AAAI*, 13636–13645.

Tanaka, R.; Nishida, K.; and Yoshida, S. 2021. VisualMRC: Machine Reading Comprehension on Document Images. In *AAAI*, 13878–13888.

Tüselmann, O.; Müller, F.; Wolf, F.; and Fink, G. A. 2022. Recognition-free Question Answering on Handwritten Document Collections. In *ICFHR*, 259–273.

Wang, B.; Li, G.; Zhou, X.; Chen, Z.; Grossman, T.; and Li, Y. 2021. Screen2words: Automatic mobile UI summarization with multimodal learning. In *UIST*, 498–510.

Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2021. Finetuned language models are zero-shot learners. In *ICLR*.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *KDD*, 1192–1200.

Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florêncio, D. A. F.; Zhang, C.; Che, W.; Zhang, M.; and Zhou, L. 2021. LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. In *ACL/IJCNLP*, 2579–2591.

Xu, Z.; Shen, Y.; and Huang, L. 2023. MultiInstruct: Improving Multi-Modal Zero-Shot Learning via Instruction Tuning. In *ACL*, 11445–11465.

Ye, J.; Hu, A.; Xu, H.; Ye, Q.; Yan, M.; Dan, Y.; Zhao, C.; Xu, G.; Li, C.; Tian, J.; Qi, Q.; Zhang, J.; and Huang, F. 2023a. mPLUG-DocOwl: Modularized Multimodal Large Language Model for Document Understanding. *arXiv:2307.02499*.

Ye, Q.; Xu, H.; Xu, G.; Ye, J.; Yan, M.; Zhou, Y.; Wang, J.; Hu, A.; Shi, P.; Shi, Y.; Jiang, C.; Li, C.; Xu, Y.; Chen, H.; Tian, J.; Qi, Q.; Zhang, J.; and Huang, F. 2023b. mPLUG-Owl: Modularization Empowers Large Language Models with Multimodality. *arXiv:2304.14178*.

Zhang, Y.; Zhang, R.; Gu, J.; Zhou, Y.; Lipka, N.; Yang, D.; and Sun, T. 2023. LLaVAR: Enhanced Visual Instruction Tuning for Text-Rich Image Understanding. *arXiv:2306.17107*.

Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2023. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. *arXiv:2304.10592*.

Zhu, F.; Lei, W.; Feng, F.; Wang, C.; Zhang, H.; and Chua, T.-S. 2022. Towards Complex Document Understanding by Discrete Reasoning. In *ACMM*, 4857–4866.