LayoutLLM: Layout Instruction Tuning with Large Language Models for Document Understanding

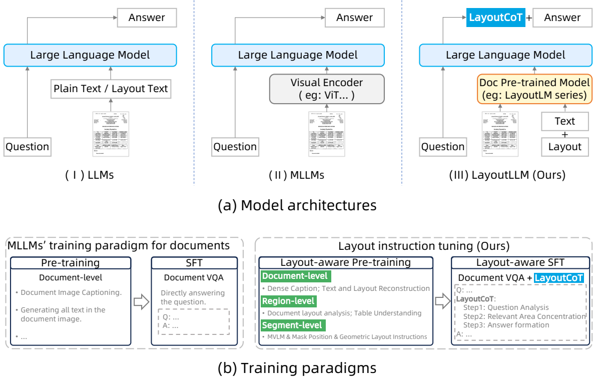
Chuwei Luo[1*], Yufan Shen[12*], Zhaoqing Zhu[1*], Qi Zheng[1], Zhi Yu[2], Cong Yao[1]
[1]Alibaba Group, [2]Zhejiang University
{luochuwei,zzhaoqing.z,zhengqisjtu,yaocong2010}@gmail.com
{syficy,yuzhirenzhe}@zju.edu.cn

Abstract

Recently, leveraging large language models (LLMs) or multimodal large language models (MLLMs) for document understanding has been proven very promising. However, previous works that employ LLMs/MLLMs for document understanding have not fully explored and utilized the document layout information, which is vital for precise document understanding. In this paper, we propose LayoutLLM, an LLM/MLLM based method for document understanding. The core of LayoutLLM is a layout instruction tuning strategy, which is specially designed to enhance the comprehension and utilization of document layouts. The proposed layout instruction tuning strategy consists of two components: Layout-aware Pre-training and Layout-aware Supervised Fine-tuning. To capture the characteristics of document layout in Layout-aware Pre-training, three groups of pre-training tasks, corresponding to document-level, region-level and segment-level information, are introduced. Furthermore, a novel module called layout chain-of-thought (LayoutCoT) is devised to enable LayoutLLM to focus on regions relevant to the question and generate accurate answers. LayoutCoT is effective for boosting the performance of document understanding. Meanwhile, it brings a certain degree of interpretability, which could facilitate manual inspection and correction. Experiments on standard benchmarks show that the proposed LayoutLLM significantly outperforms existing methods that adopt open-source 7B LLMs/MLLMs for document understanding.

1. Introduction

Document AI [7], including its document understanding tasks such as document VQA [33, 47] and document visual information extraction [18, 19, 37], is currently a hot topic in both academia and industry. In recent years, document pre-trained models [2, 8, 12, 13, 16, 17, 23, 25, 26, 32, 38, 52, 54, 55, 59] have achieved excellent performance in document AI downstream tasks. However, due to the necessity for fine-tuning on corresponding downstream task data, it is challenging to directly adapt such pre-trained models for *zero-shot* document understanding. In this paper, *zero-shot* refers to not using training sets of downstream tasks.



(a) Model architectures

(b) Training paradigms

Figure 1. LLMs/MLLMs for document understanding. The LayoutLLM is an LLM/MLLM based method that integrates a document pre-trained model as encoder. It is trained by the newly proposed layout instruction tuning strategy which consists of Layout-aware Pre-training and Layout-aware Supervised Fine-tuning.

Recently, large language models (LLMs) such as ChatGPT [35] and LLaMA [49, 50], or multimodal large language models (MLLMs) like GPT-4V [1, 36, 56], have shown remarkable zero-shot capabilities across various applications. For Document AI, as shown in Fig. 1 (a), (I) directly prompting LLMs with document text [15, 39] and (II) training document-based MLLMs [3, 57, 60] have also achieved promising results under the zero-shot setting [3, 39, 57, 60].

It is widely accepted that document layout information is vital for document understanding [2, 8, 12, 13, 16, 17, 23, 25, 26, 32, 38, 41, 52, 54, 55, 59]. However, it is difficult to convey document layout information by directly feeding text to LLMs. As Fig. 1(a)(I) shows, representing documents as either flattened plain text or layout text such as text with coordinates [15, 39, 44, 64] is often used for LLMs.

[*]Equal contribution.

Flattened plain text completely excludes any layout information of the document [54]. Additionally, as Tab. 1 shows, using layout text that represents both textual and layout information as inputs for LLMs does not guarantee LLMs can effectively comprehend this formatted text.

Moreover, existing works that employ MLLMs for document understanding also have not fully explored the document layout information. Document-based MLLMs integrate visual models [11, 42] with LLM [48–50, 61] for document understanding. As Fig. 1(b) shows, they are typically based on pre-training and supervised fine-tuning (SFT) on document datasets. In the pre-training stage, tasks such as image captioning [29, 57, 60] or generating all text in a document as flattened plain text [9, 27, 60] are commonly applied. Both these image captions [43, 46] and plain text only provide a brief representation and fail to capture the layout information of the document. So it is difficult for the model to learn document layout in the existing pre-training stage. In the SFT stage, document-related VQA or information extraction data [3, 57] is often used. The answers are directly provided during SFT, lacking explicit learning about document layout. In summary, current approaches using plain or layout text to prompt LLMs and training document-based MLLMs have not effectively captured layout information, limiting their zero-shot document understanding capability. Therefore, for better document understanding with the power of LLMs, it is necessary to investigate how to effectively incorporate layout information into LLMs.

To this end, we propose ***LayoutLLM***, an LLM/MLLM based method for document understanding, in which a layout instruction tuning strategy is designed to enhance the comprehension of document layouts. Different from the existing MLLMs that use a general visual pre-trained model [11, 42] as the encoder, we integrate document pre-trained models [2, 12, 16, 17, 23, 25, 32, 52, 54, 55] as the encoder in order to better leverage the model's foundational understanding capability for documents. The proposed layout instruction tuning consists of two stages: layout-aware pre-training and layout-aware supervised fine-tuning (SFT). Due to the complex nature of documents in real-world scenarios, encompassing rich textual content and diverse layout structures, achieving a thorough understanding involves not only capturing the document's fundamental content at global but also delving into local details. In the layout-aware pre-training stage, to ensure the model learns not only the global information of documents but also detailed information at different hierarchical levels, three groups of different level pre-training tasks are proposed: document-level, region-level, and segment-level. All the proposed pre-training tasks are unified in the format of instruction tuning.

Furthermore, in the layout-aware SFT stage, to enhance the model's comprehension and utilization of layout information for question answering, a novel strategy called *LayoutCoT* is proposed, motivated by the chain-of-thought (CoT) [21, 53] ability in LLMs. Unlike existing methods that are directly supervised by the answer to the document understanding question, *LayoutCoT* consists of three successive steps: *Question Analysis*, *Relevant Area Concentration*, and *Answer Formation*. Through these steps, the model gains a deeper understanding of the questions, becomes capable of focusing the the relevant areas instead of searching answers in the entire document and can leverage the specific characteristics of identified areas (such as tables, paragraphs, etc.) to accurately infer the answers. It not only brings a certain degree of interpretability, but also provides a feasible way for manual intervention or correction of model results. Extensive zero-shot experiments on five widely-used document understanding benchmarks demonstrate the effectiveness of the proposed LayoutLLM.

Our contributions are summarized as follows:

1) To better learn document layouts from global to local in layout-aware pre-training, three groups of different level pre-training tasks, which are all implemented through instruction tuning, are proposed.
2) A novel *LayoutCoT* strategy is proposed to achieve layout-aware supervised fine-tuning. It enables LayoutLLM to focus on the relevant document area and leverage the region's features to generate accurate answers, exhibiting a certain degree of interpretability.
3) Experimental results on zero-shot document understanding tasks show that the proposed LayoutLLM significantly outperforms existing methods that adopt LLMs/MLLMs for document understanding, demonstrating the great potential of document layout modeling.

2. Related Works

Pre-trained models for document understanding. Document pre-trained models have demonstrated the effectiveness of layout information in document understanding [2, 5, 8, 10, 12, 13, 16, 17, 20, 22, 23, 25, 26, 32, 38, 41, 52, 54, 55, 59]. As a pioneer, LayoutLM [54] is the first to encode spatial coordinates of text for layout representation learning in pre-training documents. The following works [2, 8, 12, 13, 16, 17, 23, 25, 26, 32, 38, 52, 55, 59] then joint text, layout and images in document pre-training by combining visual models as document image encoders with the text and layout transformers, and various works [5, 10, 20, 22] start to explore pre-training end-to-end models for document understanding. These studies have achieved significant advancements in document understanding by exploring various model architectures [2, 8, 10, 12, 13, 20, 25, 26, 38, 41, 52, 59] and attention mechanisms [16, 17, 55] for modeling layout information. These methods also have proposed layout-related pre-training tasks that have been demonstrated to be highly effective in document understanding tasks. For instance, tasks like masked vision-language modeling [13, 17, 55], where the model is required to generate the original text corresponding to the randomly masked text in the document; position masking [31, 51], involving the randomly position masking and subsequent recovery of position information in the document; geometric pre-training [26, 32], focusing on learning direction, distance, etc.; and layout-aware generation tasks [5, 22], aiming to make the model generate structured text with layout information. However, due to the necessity of fine-tuning with annotated data for downstream tasks, these efforts face challenges in extending to zero-shot document understanding.

LLMs/MLLMs for document understanding. Recently, LLMs like ChatGPT [35] and MLLMs like GPT-4 [36, 56] have demonstrated remarkable zero-shot performance across a wide range of NLP/CV tasks. Leveraging LLMs/MLLMs for zero-shot document understanding has also shown promising progress [3, 30, 39, 45, 57, 60]. Perot et al. [39] explore the use of LLMs for document visual information extraction, emphasizing the importance of the document layout. LLaVAR [60] which extends LLaVA [28, 29] to the document domain is pre-trained by generating plain text in the document image. During SFT, it is trained by document-related instructions which are generated by GPT-4. Expanding on mPLUG-Owl [58], mPLUG-DocOwl [57] is trained using publicly available datasets for document understanding. It includes tasks like document-level image captioning, direct information extraction, and direct document VQA. Moreover, Qwen-VL [3] proposes a general MLLM that performs well on document understanding tasks, utilizing document-level pre-training and direct VQA for SFT. Though existing LLMs/MLLMs have shown promising results in document understanding, their limited focus on document layout in pre-training and SFT has hindered their ability to achieve higher accuracy in zero-shot document understanding and better interpretability.

3. LayoutLLM

LayoutLLM is an LLM/MLLM based method that incorporates document pre-trained models for document understanding. To enhance the document layout comprehension in LayoutLLM, a novel layout instruction tuning strategy is proposed, which consists of two stages: layout-aware pre-training and layout-aware supervised fine-tuning (SFT).

3.1. Model Architecture

The overall architecture of LayoutLLM is shown in Fig. 2. In LayoutLLM, given an input document image and its corresponding text and layout information, the document pre-trained model encoder is required to obtain the multimodal document features. Then, these features are encoded by multimodal projectors, and together with the instruction embeddings, fed into the LLM to generate the final results.

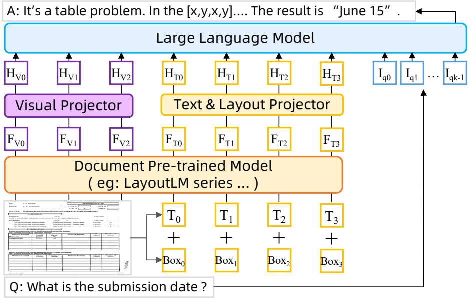


Figure 2. Overall architecture of LayoutLLM.

Document pre-trained model encoder. To leverage the foundational document comprehension capability of document pre-trained models, in this work, we utilize LayoutLMv3 [17], a widely-used document pre-training model, as our basic document encoder. The document image, text, and layout are initially inputted into the document pre-trained model ($DocPTM$). They are then encoded by the $DocPTM$ to obtain the corresponding features as follows:

$$F_V, F_T = DocPTM(V, T, Box) \tag{1}$$

where V represents the document image, $T = T_{0:n-1}$ and $Box = Box_{0:n-1}$ indicate the text sequences in the document and their corresponding bounding-box coordinates respectively. After being encoded by the $DocPTM$, the visual features of the document $F_V = F_{V_0:V_{m-1}} \in \mathbb{R}^{d_0}$ and the text layout features $F_T = F_{T_0:T_{n-1}} \in \mathbb{R}^{d_0}$ are acquired. m signifies the number of visual features and n represents the number of tokens contained in the document. d_0 denotes the dimension of $DocPTM$ feature space.

Multimodal projectors. To project multi-modality features from $DocPTM$ into the LLM's embedding space, inspired by the simple yet effective projector design in LLaVA [28, 29], two different Multilayer Perceptrons (MLPs) are used as visual projector and text & layout projector respectively. Formally, the projected features can be obtained by:

$$H_V = P_V(F_V) \tag{2}$$

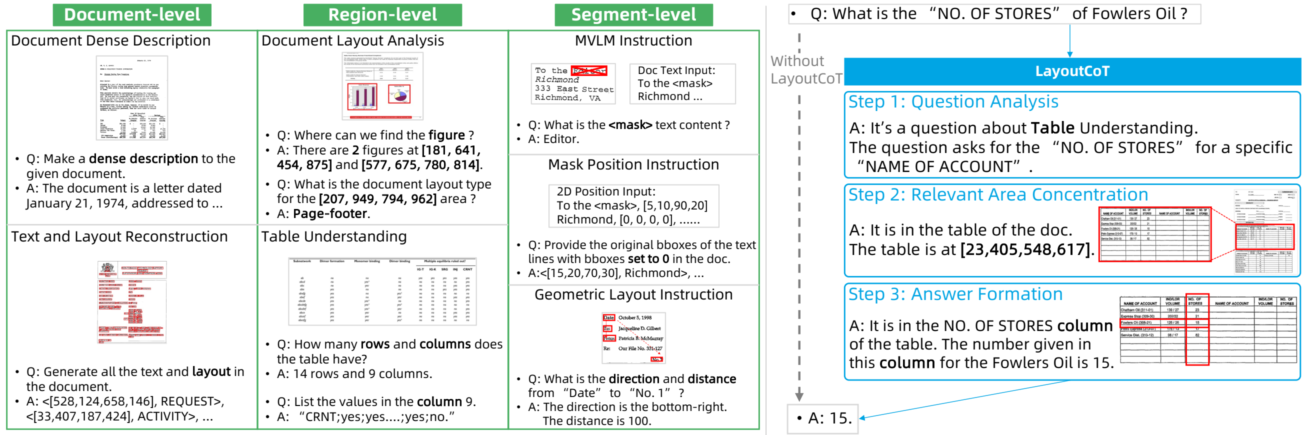
$$H_T = P_T(F_T) \tag{3}$$

where $H_V = H_{V_0:V_{m-1}} \in \mathbb{R}^{d_1}$ is the feature encoded by the visual projector, $H_T = H_{T_0:T_{n-1}} \in \mathbb{R}^{d_1}$ is the feature encoded by the text & layout projector, and d_1 is the dimensional of the LLM embedding features.

Large language model. Finally, the H_V, H_T and the embedding of the question's instruction text, $I_q = I_{q_{0:l_q-1}}$, are inputted together into the LLM, generating the target answer $I_a = I_{a_{0:l_a-1}}$. l_q and l_a represent the length of the question's instruction text and the answer text, respectively.

3.2. Layout Instruction Tuning

The LayoutLLM model is trained using the layout instruction tuning, which consists of two stages: layout-aware pre-training and layout-aware SFT.



- Q: Make a **dense description** to the given document.
- A: The document is a letter dated January 21, 1974, addressed to ...

- Q: Generate all the text and **layout** in the document.
- A: <[528,124,658,146], REQUEST>, <[33,407,187,424], ACTIVITY>, ...

- Q: Where can we find the **figure** ?
- A: There are **2** figures at **[181, 641, 454, 875]** and **[577, 675, 780, 814]**.
- Q: What is the document layout type for the **[207, 949, 794, 962]** area ?
- A: **Page-footer**.

- Q: How many **rows** and **columns** does the table have?
- A: 14 rows and 9 columns.

- Q: List the values in the **column** 9.
- A: “CRNT;yes;yes....;yes;no.”

To the Editor
Richmond
333 East Street
Richmond, VA

Doc Text Input:
To the <mask>
Richmond ...

- Q: What is the **<mask>** text content ?
- A: Editor.

2D Position Input:
To the <mask>, [5,10,90,20]
Richmond, [0, 0, 0, 0], ...

- Q: Provide the original bboxes of the text lines with bboxes **set to 0** in the doc.
- A:<[15,20,70,30], Richmond>, ...

- Q: What is the **direction** and **distance** from “Date” to “No. 1” ?
- A: The direction is the bottom-right. The distance is 100.

- Q: What is the “NO. OF STORES” of Fowlers Oil ?

Step 1: Question Analysis

A: It’s a question about **Table** Understanding.
The question asks for the “NO. OF STORES” for a specific “NAME OF ACCOUNT” .

Step 2: Relevant Area Concentration

A: It is in the table of the doc.
The table is at **[23,405,548,617]**.

Step 3: Answer Formation

A: It is in the NO. OF STORES **column** of the table. The number given in this **column** for the Fowlers Oil is 15.

- A: 15.

(a) Stage 1: Layout-aware Pre-training

(b) Stage 2: Layout-aware Supervised Fine-tuning

Figure 3. Overview of the Layout Instruction Tuning. (a) Document-level, region-level, and segment-level pre-training tasks, unified in instruction tuning format, are introduced. (b) A novel module called *LayoutCoT* is designed to enable LayoutLLM to focus on regions relevant to the question and generate accurate answers through three intermediate steps.

3.2.1 Layout-aware Pre-training

The goal of pre-training the LayoutLLM is to enhance the model’s comprehensive understanding of documents at different levels through layout learning, rather than only focusing on global document-level understanding like existing MLLM methods [57, 60]. To this end, during the pre-training stage, three different level pre-training strategies are simultaneously applied to the LayoutLLM, namely document-level, region-level, and segment-level.

Document-level To enable the model to possess fundamental global document understanding, pre-training tasks, namely Document Dense Description (DDD) and Text and Layout Reconstruction (TLR), are proposed. As Fig. 3 (a) shows, like the image caption task, the DDD task requires the model to learn to describe the input document. Moreover, in the DDD task, the descriptions for document images are more detailed. For instance, in the document image caption data used for the LLaVAR [60] pre-training, captions contain an average of 36.27 words, while in the proposed dataset of the DDD task, the description contains an average of 373.25 words. Through the DDD task, the model can obtain basic document-level information, such as document type and detailed content. The TLR task aims to reconstruct the complete text and layout information of a document and output it in the format “$<\{\text{box}\}, \{\text{text}\}>$”. The TLR task aligns the text and layout embeddings output from DocPTM with the LayoutLLM’s LLM space. Consequently, it enables the LLM in LayoutLLM to comprehend the text and layout information contained in the documents.

Region-level The information contained in specific regions of a document, such as the titles, figures, tables, is essential for document understanding [4, 6, 8, 24, 62]. These regions serve as important characteristics that differentiate a document from plain text in natural language. For the LayoutLLM to achieve the basic region-level understanding, two pre-training tasks, namely Document Layout Analysis (DLA) and Table Understanding (TU), are utilized. The DLA task is achieved in two ways as shown in Fig. 3. One involves locating the layout region based on the layout type, the other involves identifying the type of a given area. Furthermore, the table region differs from other regions in that it requires additional focus on 2D layout understanding. The TU task enables the model to understand the basic row and column information in the table region of a document. As shown in Fig. 3, the TU task includes instruction tuning for the number of rows and columns, logical coordinates, and the content within rows and columns.

Segment-level Early works on document pre-trained models [13, 17, 26, 31, 32, 51, 55] have demonstrated the effectiveness of segment-level document pre-training tasks to document layout understanding ability, such as masked vision language modeling (MVLM) [13, 17, 55], position masking [31, 51], and geometric pre-training [26, 32]. Inspired by these works, to make LayoutLLM have segment-level layout understanding, these tasks are transformed into instruction formats for pre-training as Fig. 3(a) shows. For the MVLM instructions, random masking of the text input to LayoutLLM is performed, and the model is instruction tuned by asking the masked words and answering them. For mask position instruction, the layout information (coordinates) to a specific text line, when input to LayoutLLM, is randomly set to 0. The instruction is constructed by asking about the text line with zeroed coordinates and requesting the model to respond with the original coordinates with text content. For geometric layout instruction, text lines are randomly selected, and an instruction is constructed by asking questions about the direction and distance between them.

3.2.2 Layout-aware Supervised Fine-tuning

In the SFT stage of existing document-based MLLMs, models are directly supervised by the answer to the document understanding instructions. Consequently, these methods lack explicit learning of document layout which is crucial for document understanding. Considering this limitation, and inspired by previous works related to chain-of-though (CoT) [21, 53], which have shown that inferencing with intermediate steps can greatly enhance performance. A novel module called *LayoutCoT* is proposed, which incorporates the layout information into every intermediate step of CoT explicitly. Meanwhile, by introducing the layout-aware intermediate steps, the answer process gains a certain degree of interpretability for LayoutLLM and also provides interactive correction possibilities based on *LayoutCoT*.

LayoutCoT Details. As shown in Fig. 3(b), the *LayoutCoT* involves the following three intermediate steps:

Step 1: Question Analysis. To effectively address a document understanding problem, analyzing the key characteristics of the question is very important. Identifying the question type, such as table understanding or entity extraction from paragraphs, and assessing whether the question is a straightforward extraction query or a more complex reasoning problem, can help guide the direction for the subsequent inference process. Therefore, to give basic guidance to the subsequent steps, the question analysis step is designed, encompassing an analysis of the question type from a layout perspective and a detailed understanding of the question itself. Benefiting from the layout understanding ability by layout-aware pre-training, this step can extract the types and key information mentioned in the question, which are related to the specific characteristics of the document.

Step 2: Relevant Area Concentration. For most document understanding tasks, the entire document contains a large amount of irrelevant information that may confuse the model [5]. This step aims to focus on the relevant area and generate its location information, which is used to assist the model to accurately infer the answer. Benefiting from the layout information conveyed by step 1 and the positioning capabilities learned from the region&segment-level pre-training, the model can accurately generate the location of the relevant area. For example in Fig. 3(b), according to the question type "table" in step 1, the relevant "table" can be located. By guiding the model to focus on the relevant area, this step largely narrows the search scope, increasing the possibility of giving the right answer. Meanwhile, the location information provides a way for visual inspection and interactive correction (see Sec. 4.7 for details).

Step 3: Answer Formation. Finally, the last step, the answer formation, provides explanations based on the layout characteristics of the relevant areas located in step 2 and key points analyzed in step 1 to get the final answer. For example in Fig. 3(b), for a "table" type question, this step involves analyzing the row and column in the relevant table in step 2, and inferencing the answer step-by-step. For a "key-value" question, analyzing the keywords in concentrated areas can help get the final answer. Analyzing answers in different ways based on the features of different layout regions not only improves the document understanding performance but also brings a certain level of interpretability.

```
Algorithm 1 CONSTRUCT($\mathcal{D}$): LayoutCoT Construction.
Definition: $\mathcal{H}$: Document HTMLs; $\mathcal{I}$: Document Images; $\mathcal{T}$: MRC
    Texts; $\mathcal{R}$: Document Language Representation; $\mathcal{QA}$: QA pairs; $\mathcal{T}_c$:
    Text CoT; $\mathcal{L}_c$: LayoutCoT;
Input: Document Dataset $\mathcal{D} = \{\mathcal{D}_H, \mathcal{D}_I, \mathcal{D}_M\}$
    ($\mathcal{D}_H$=$\{\mathcal{H}\}$, $\mathcal{D}_I$=$\{\mathcal{I}\}$, $\mathcal{D}_M = \{\mathcal{T}, \mathcal{QA}\}$);
Output: Constructed Dataset $\mathcal{D}_c$
Procedure CONSTRUCT($\mathcal{D} = \{\mathcal{D}_H, \mathcal{D}_I, \mathcal{D}_M\}$)
    1) $\mathcal{R}$ = getDocRep($\mathcal{I}$) if $\mathcal{D} \subseteq \mathcal{D}_I$ else $\mathcal{H}$ if $\mathcal{D} \subseteq \mathcal{D}_H$ else pass;
    2) $\mathcal{QA}, \mathcal{T}_c$ = getQACoT($\mathcal{D}$) if $\mathcal{D} \subseteq \mathcal{D}_M$ else getQACoTGPT($\mathcal{R}$);
    3) $\mathcal{L}_c$ = getLayoutCoT($\mathcal{T}_c$)
    4) if $\mathcal{D} \subseteq \{\mathcal{D}_H, \mathcal{D}_M\}$: $\mathcal{I}$ = Html2Img($\mathcal{D}_H$ ? $\mathcal{H}$ : $\mathcal{T}$);
return $\mathcal{D}_c \leftarrow \{\mathcal{I}, \mathcal{QA}, \mathcal{L}_c\}$
```

LayoutCoT Construction. Given the need for both text and image annotations in constructing *LayoutCoT*, manual labeling can be difficult. Algorithm 1 proposes a manual-labeling-free method, generating *LayoutCoT* data using public datasets with GPT (GPT-3.5 Turbo). It involves representing document text and layout in a format understandable by GPT. GPT is then utilized to generate document-content-based QA and corresponding text CoT. Finally, use rules for transforming the text CoT to LayoutCoT.

Three types of publicly available document datasets are focused on: HTML documents ($\mathcal{D}_H$), image documents ($\mathcal{D}_I$), and text documents ($\mathcal{D}_M$) for machine reading comprehension (MRC). The construction process is as follows:

1) **Document Representation**: To fully leverage the capabilities of GPT, it is crucial to ensure that the document content fed to GPT contains accurate layout information. Since HTML is the formatted language that can represent documents accurately, $\mathcal{D}_H$ is represented using the original HTML. By transforming HTML to PDF and using the PDF parser, the text and bounding-boxes are obtained. For $\mathcal{D}_I$, the layout-aware text [15] is used. The text and bounding-boxes in $\mathcal{D}_I$ are from the original dataset annotations.

2) **QA&Text CoT Generation**: The language representation $\mathcal{R}$ for the document is employed for prompting GPT to generate QA pairs $\mathcal{QA}$ with text CoTs $\mathcal{T}_c$. In addition, the $\mathcal{D}_M$ includes the QA pairs and reasoning process, thereby directly reusing the $\mathcal{QA}$ and manually organizing $\mathcal{T}_c$. The generated $\mathcal{T}_c$ includes the step 1 (question analysis) and step 3 (answer formation) for LayoutCoT, and locates all relevant sentences in the document for $\mathcal{QA}$.

3) **LayoutCoT Generation**: The step 1 & 3 in $\mathcal{T}_c$ are used as the step 1 & 3 in $\mathcal{L}_c$. To construct the step 2 (relevant area concentration) of $\mathcal{L}_c$, the union bounding-box of all located relevant sentences in $\mathcal{T}_c$ are taken as the relevant area.

4) **Document Images Generation**: For $\mathcal{D}_H$ and $\mathcal{D}_M$, the HTMLs and MRC text are converted to images. Overall, the document images $\mathcal{I}$, generated QA pairs $\mathcal{QA}$ and LayoutCoTs $\mathcal{L}_c$ constitute the final LayoutCoT dataset $\mathcal{D}_c$.

4. Experiments

4.1. Dataset Collection

Layout-aware pre-training data of LayoutLLM is from publicly available document understanding datasets. It does not incorporate any data from the training, validation, and test sets of downstream benchmarks. Region-level pre-training tasks, most document-level and segment-level tasks are self-supervised. Thus, only document images and images converted from PDFs in the original datasets, along with the corresponding OCR or text-layout results from PDF parsing, are needed. For these tasks, data is randomly sampled from PubLayNet [62], DocLayNet [40], Docbank [24], RVL-CDIP [14], and DocILE [65]. Particularly, data for document dense description is from inputting the document text content into GPT-3.5 Turbo, prompting it to generate an average of 373.25 words document dense descriptions. For the region-level tasks, specifically the document layout analysis task, publicly available document layout analysis datasets are utilized, including PubLayNet [62], DocLayNet [40], and Docbank [24]. Data for another region-level task, table understanding, is sourced from PubTabNet [63] with its table annotations. All data is transformed into the instruction format illustrated in Fig. 3(a). In total, 5.7 million instructions are constructed, with a ratio of 1:4:4 for document-level, region-level, and segment-level tasks, respectively. For detailed instruction templates and dataset descriptions, please refer to the supplementary material.

Layout-aware SFT data of LayoutLLM is generated by GPT (GPT-3.5 Turbo) and converted from existing textual Machine Reading Comprehension (MRC) datasets, as discussed in Sec. 3.2.2. To generate high-quality document-based textual QA and textual CoT, it is essential to make GPT comprehend the document layout. So, the document is represented using both layout text [15] and HTML. Similar to the pre-training data, the D_I in Algorithm 1 is also randomly sampled from PubLayNet [62], DocLayNet [40], Docbank [24], RVL-CDIP [14], and DocILE [65] for building layout text. The D_H in Algorithm 1 is from GPT's free generation. The D_M in Algorithm 1 is randomly sampled from the FeTaQA [34] which is a wikipedia question answering dataset. A total of 300K instructions are constructed, with a ratio of 5:4.5:0.5 for D_I, D_H, and D_M, respectively. For detailed prompt templates of document-based text QA and text CoT generation using GPT, prompts for HTML generation using GPT, and dataset description, please refer to the supplementary material.

4.2. Training Setup

The encoder weight of LayoutLLM is initialized from the LayoutLMv3-large [17] which is a widely-used document pre-trained model. And the LLM backbone is initialized from Vicuna-7B-v1.5 [61]. Other parameters are randomly initialized. During pre-training, the LLM is frozen, and the parameters of the two projectors and document pre-trained model encoder are updated. During SFT, both LLM and two projectors are fine-tuned while keeping the document pre-trained model encoder frozen. For detailed training setup, please refer to the supplementary material.

4.3. Evaluation Setup

The zero-shot ability is highly expected in real-world document understanding scenarios [7, 30, 45]. Therefore, zero-shot evaluations on widely-used document understanding benchmarks including document visual question answering (Document VQA) and visual information extraction (VIE) are conducted. Only the test sets are utilized in all benchmarks and only the official provided image, text, and layout information are used. The Document VQA datasets comprise the **DocVQA**[33] test set, consisting of 5,188 questions, and the **VisualMRC**[47] test set containing 6,708 questions. Following the evaluation metric settings of the original datasets, the ANLS [33] is utilized for evaluating DocVQA, and Rouge-L is used for evaluating VisualMRC. For the VIE task, **FUNSD** [19], **CORD** [37], and **SROIE** [18] are used. The test set of FUNSD comprises 50 form images, each annotated with entity-level headers, questions, answers, and others, along with entity linking annotations. CORD's test set consists of 100 receipt images, annotated with 30 entity types, such as the tax amount, total price, etc. SROIE's test set includes 347 receipt images, annotated with 4 entity types: company, date, address, and total. To prompt LLMs/MLLMs for zero-shot VIE, annotations in VIE datasets are transformed into question answering format (QA for VIE). For key-value annotations with linking in the FUNSD, the format is {Q: What is the "key" in the document? A: "value"}. For entity annotations in CORD and SROIE, directly asking for the entity in the document like {Q: What is the address in the document? A: "the address annotation"} is utilized. Following DocVQA, the QA for VIE task is evaluated by ANLS.

4.4. Main Results

As shown in Tab. 1, the zero-shot document understanding performance of LayoutLLM and existing open-source LLMs and MLLMs is evaluated. Generally, the existing LLMs are better than MLLMs for zero-shot document VQA and VIE. For example in the results on DocVQA, most LLMs can achieve a performance of around 60% or higher, while most MLLMs can only attain around 10%, except for mPLUG-DocOWL and Qwen-VL that trained with the

	Models	Document VQA		QA for VIE		
		DocVQA	VisualMRC	FUNSD	CORD	SROIE
Fine-tuned PTM	LayoutLMv3 [17]	83.37^{*}	-	$92.08^{*\ddagger}$	$97.46^{*\ddagger}$	-
LLM *Plain Text*	Llama2-7B [50]	61.34	29.73	40.78	4.39	15.86
	Llama2-7B-chat [50]	64.99	52.84	48.20	47.70	68.97
	Vicuna-7B [61]	61.39	53.63	49.79	44.67	67.49
	Vicuna-1.5-7B [61]	66.99	52.13	48.06	51.40	68.20
LLM *Layout Text* (Text + Box) [15]	Llama2-7B [50]	37.32	33.82	51.40	28.04	34.96
	Llama2-7B-chat [50]	56.55	49.26	58.34	50.93	51.15
	Vicuna-7B [61]	37.21	52.55	42.73	46.59	45.43
	Vicuna-1.5-7B [61]	56.81	47.22	59.63	56.13	66.20
MLLM	LLaVAR-7B [60]	11.6†	36.37	1.71	13.55	2.38
	LLaVA-1.5-7B [28]	13.34	35.23	1.93	18.06	3.83
	mPLUG-DocOWL-7B [57]	62.2^{*}†	-	-	-	-
	Qwen-VL-7B [3]	65.1^{*}†	42.52	47.09	30.00	58.59
	LayoutLLM-7B△ (Ours)	**74.25**	**55.73**	**78.65**	**62.21**	**70.97**
	LayoutLLM-7B⋆ (Ours)	**74.27**	**55.76**	**79.98**	**63.10**	**72.12**

Table 1. Zero-shot document understanding results on open-source LLMs and MLLMs. $*$ signifies training set use; unmarked results are zero-shot. Results marked with ‡ are F1 scores for VIE. Results marked with † are from the original paper and others are re-implemented by us. △ marks LayoutLLM's LLM backbone as initialized with Llama2-7B-chat, and ⋆ with Vicuna-1.5-7B.

training set. One possible reason is that it's difficult for these MLLMs to obtain accurate textual information from document images. Additionally, for LLMs, using Plain Text and Layout Text respectively as the document representation are further discussed, where the Layout Text introduces layout information by adding text coordinates in the format: {text:"text", box:[x1,y1,x2,y2]} [15]. Compared to the Plain Text, the Layout Text variant doesn't show stable performance improvements, noticeable in certain tasks, for example in Vicuna-1.5, an improvement in VIE (FUNSD 48.06% to 59.63%) but a decline in DocVQA (66.99% to 56.81%). LLMs may lack the ability to learn this formatted layout text, and directly adding layout information (e.g., coordinates) to the text will also highly increase the token length, making the answer inference more challenging.

Compared to the prior SOTA model, LayoutLMv3, which is fine-tuned using the training set of downstream tasks, LayoutLLM demonstrates competitive performance on the DocVQA benchmark. Compared with these LLMs and MLLMs, LayoutLLM achieves consistent and significant improvements over them on all evaluation benchmarks. Notably, LayoutLLM which employs zero-shot performance, outperforms mPLUG-DocOWL and Qwen-VL by around 10% on the DocVQA dataset, both of which are trained with this dataset. This demonstrates that LayoutLLM can learn more robust and discriminative representations for document understanding. Furthermore, experiments of the different initialization of the LLM backbone have all achieved optimal results across all benchmarks, substantiating that LayoutLLM can adapt to various LLMs. In summary, our method explores a more effective way to utilize layout information for document understanding, which significantly improves the performance of zero-shot document understanding.

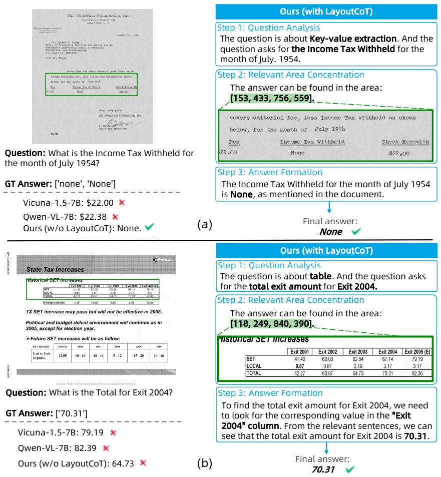
#	Layout-aware Pre-training	Layout-aware SFT	DocVQA	FUNSD
0			70.82	70.96
1	✓		72.31	74.02
2	✓	✓	**74.27**	**79.98**

Table 2. Ablation study on the DocVQA and FUNSD test sets.

4.5. Ablation study

To better verify the effectiveness of the layout-aware pre-training and layout-aware SFT in the layout instruction tuning, an ablation study is conducted (see Tab. 2).

Initial baseline. The #0 baseline disables both layout-aware pre-training and layout-aware SFT. It only adopts SFT (same SFT data but without LayoutCoT steps) on the LayoutLLM. Even without any alignment pre-training and LayoutCoT steps guidance, the baseline outperforms previous SOTAs, achieving 70.82% on DocVQA and 70.96% on FUNSD. This indicates the document understanding ability of DocPTM benefits document understanding with LLMs. **Effect of Layout-aware pre-training.** In #1, the layout-aware pre-training and the same SFT with #0 is conducted. Compared to #0, benefiting from the basic document understanding capability learned through the layout-aware pre-training, #1 shows an increase of 1.49% on DocVQA and 3.06% on FUNSD. It can be observed that the basic document understanding ability learned from the layout-aware pre-training significantly enhances the performance of the basic key-value extraction tasks in the FUNSD dataset. Compared to the FUNSD VIE task, it also shows that the DocVQA is a more complex task. **Effect of Layout-aware SFT.** Compared with #1, #2 further incorporates the layout-aware SFT strategy, resulting in a performance gain of 1.96% on DocVQA and even 5.96% on FUNSD. This indicates the LayoutCoT in layout-aware SFT can help the LayoutLLM to handle complex document tasks and it is effec-



Ours (with LayoutCoT)

Step 1: Question Analysis
The question is about **Key-value extraction**. And the question asks for **the Income Tax Withheld** for the month of July. 1954.

Step 2: Relevant Area Concentration
The answer can be found in the area: **[153, 433, 756, 559]**.

covers editorial fee, less Income Tax withheld as shown below, for the month of July 1954

Fee	Income Tax Withheld	Check Herewith
$22.00	None	$22.00

Step 3: Answer Formation
The Income Tax Withheld for the month of July 1954 is **None**, as mentioned in the document.

Final answer: ***None*** ✓

Question: What is the Income Tax Withheld for the month of July 1954?

GT Answer: ['none', 'None']

Vicuna-1.5-7B: $22.00 ✗
Qwen-VL-7B: $22.38 ✗
Ours (w/o LayoutCoT): None. ✓

(a)

Ours (with LayoutCoT)

Step 1: Question Analysis
The question is about **table**. And the question asks for the **total exit amount** for **Exit 2004.**

Step 2: Relevant Area Concentration
The answer can be found in the area: **[118, 249, 840, 390]**.

Historical SET increases

	Exit 2001	Exit 2002	Exit 2003	Exit 2004	Exit 2005 (E)
SET	41.40	60.00	62.54	67.14	79.19
LOCAL	0.87	0.87	2.19	3.17	3.17
TOTAL	42.27	60.87	64.73	70.31	82.36

Step 3: Answer Formation
To find the total exit amount for Exit 2004, we need to look for the corresponding value in the **"Exit 2004" column**. From the relevant sentences, we can see that the total exit amount for Exit 2004 is **70.31**.

Final answer: ***70.31*** ✓

Question: What is the Total for Exit 2004?

GT Answer: ['70.31']

Vicuna-1.5-7B: 79.19 ✗
Qwen-VL-7B: 82.39 ✗
Ours (w/o LayoutCoT): 64.73 ✗

(b)

Figure 4. Qualitative results on DocVQA. **Green** boxes are the areas concentrated in the step 2 of LayoutCoT.

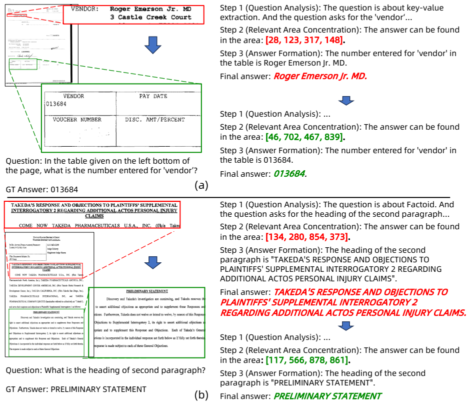
tive for boosting the performance on both document VQA and VIE. Moreover, it also promotes a certain degree of interpretability. Overall, the ablation study demonstrates the importance of layout-aware pre-training and layout-aware SFT for zero-shot document understanding.

4.6. Qualitative Results

Two examples are shown in Fig. 4. Through combined with layout-aware pre-training and layout-aware SFT, LayoutLLM can accurately focus on the relevant areas, utilize the layout information to assist in problem-solving and provide interpretability. For example, in Fig. 4(a) question about key-value extraction in up-down layout, different from the left-right variant, relies more on document layout to infer the right answer. Since the keywords "Income Tax" in the question often co-occur with numerical data, Vicuna-1.5 and Qwen-VL find numerical answers relying more on the semantics than the layout, resulting in incorrect responses. In contrast, benefiting from the layout-aware pre-training, our model can effectively leverage layout information to give accurate answers. In addition, the model using LayoutCoT can further provide the location and the reasoning process, showing a certain degree of interpretability. But in certain situations, only combined with layout pre-training, our model might fail to give accurate answers. As shown in Fig. 4(b), without LayoutCoT, our model identifies "Exit 2003" as the relevant column and generates a wrong answer. However, with the help of LayoutCoT, LayoutLLM can correctly identify the question type as "Table", locate the relevant table area, and finally infer the right answer from the corresponding "Exit 2004" column.

4.7. Interactive Correction with LayoutCoT

Since LayoutCoT runs in a step-by-step fashion and produces intermediate results in the inference stage, it can facilitate *interactive inspection and correction*, when processing a document. As shown in Fig. 5(a), there are two areas



VENDOR: Roger Emerson Jr. MD
3 Castle Creek Court

VENDOR 013684	PAY DATE
VOUCHER NUMBER	DISC. AMT/PERCENT

Step 1 (Question Analysis): The question is about key-value extraction. And the question asks for the 'vendor'...

Step 2 (Relevant Area Concentration): The answer can be found in the area: **[28, 123, 317, 148]**.

Step 3 (Answer Formation): The number entered for 'vendor' in the table is Roger Emerson Jr. MD.

Final answer: ***Roger Emerson Jr. MD.***

Step 1 (Question Analysis): ...

Step 2 (Relevant Area Concentration): The answer can be found in the area: **[46, 702, 467, 839]**.

Step 3 (Answer Formation): The number entered for 'vendor' in the table is 013684.

Final answer: ***013684.***

Question: In the table given on the left bottom of the page, what is the number entered for 'vendor'?

GT Answer: 013684

(a)

TAKEDA'S RESPONSE AND OBJECTIONS TO PLAINTIFFS' SUPPLEMENTAL INTERROGATORY 2 REGARDING ADDITIONAL ACTOS PERSONAL INJURY CLAIMS

COME NOW TAKEDA PHARMACEUTICALS U.S.A., INC. (f/k/a Taked

PRELIMINARY STATEMENT

Discovery and Takeda's investigation are continuing, and Takeda reserves th to assert additional objections as appropriate and to supplement these Responses and tions. Furthermore, Takeda does not waive or intend to waive, by reason of this Response Objections to Supplemental Interrogatory 2, its right to assert additional objections a opriate and to supplement this Response and Objections. Each of Takeda's Genera ctions is incorporated in the individual response set forth below as if fully set forth therein response is made subject to each of these General Objections.

Step 1 (Question Analysis): The question is about Factoid. And the question asks for the heading of the second paragraph...

Step 2 (Relevant Area Concentration): The answer can be found in the area: **[134, 280, 854, 373]**.

Step 3 (Answer Formation): The heading of the second paragraph is "TAKEDA'S RESPONSE AND OBJECTIONS TO PLAINTIFFS' SUPPLEMENTAL INTERROGATORY 2 REGARDING ADDITIONAL ACTOS PERSONAL INJURY CLAIMS".

Final answer: ***TAKEDA'S RESPONSE AND OBJECTIONS TO PLAINTIFFS' SUPPLEMENTAL INTERROGATORY 2 REGARDING ADDITIONAL ACTOS PERSONAL INJURY CLAIMS.***

Step 1 (Question Analysis): ...

Step 2 (Relevant Area Concentration): The answer can be found in the area: **[117, 566, 878, 861]**.

Step 3 (Answer Formation): The heading of the second paragraph is "PRELIMINARY STATEMENT".

Final answer: ***PRELIMINARY STATEMENT***

Question: What is the heading of second paragraph?

GT Answer: PRELIMINARY STATEMENT

(b)

Figure 5. Interactive correction with LayoutCoT. **Green** represents the correct relevant areas and answers, while **Red** represents the original incorrect ones. Best viewed in digital version.

in the image that are relevant to the keyword "vendor" of the question. LayoutCoT focused on a wrong area containing "vendor", as it missed "left bottom" in the question, the answer was incorrect. However, after the right area is given manually, it can finally give the correct answer. Similarly, in Fig. 5(b) the question asks about "the heading of the second paragraph". However, the term "paragraph" does not have a universal definition, and in this case, the sentences in the area below the main heading were considered as a *paragraph*, causing the model to predict "TAKEDA's...CLAIMS", which was incorrect according to the GT. Once the right "second paragraph" region is fed to the model, the answer can be successfully revised. This unique ability of LayoutCoT could be very valuable in high-stake scenarios (e.g., a bank transaction), where the standards are extremely high, and manual checking and correction (i.e., human-in-the-loop) are required.

5. Limitations

Through LayoutCoT, LayoutLLM demonstrates the capability of interactive correction, but in real-world applications, this is not enough. The ability to refuse false-positive outputs and generate hints (e.g. "The answer is not mentioned in the document.") is crucial. However, it is currently absent in LayoutLLM. In addition, despite achieving notable improvements through layout-aware pre-training, LayoutLLM struggles in precisely understanding region-level relationships, as evidenced in Fig. 5(a). We will study how to endow LayoutLLM with such abilities.

6. Conclusion

We propose LayoutLLM for document understanding, in which a layout instruction tuning strategy comprising layout-aware pre-training and layout-aware SFT is designed to improve the comprehension of document layouts. Extensive experiments confirm the effectiveness of LayoutLLM.

References

[1] GPT-4V(ision) system card. 2023. 1

[2] Srikar Appalaraju, Bhavan Jasani, and Bhargava Urala Kota. DocFormer: End-to-end transformer for document understanding. In *ICCV*, pages 4171–4186, 2021. 1, 2

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 1, 2, 3, 7

[4] Galal M Binmakhashen and Sabri A Mahmoud. Document layout analysis: a comprehensive survey. *ACM Computing Surveys (CSUR)*, 52(6):1–36, 2019. 4

[5] Haoyu Cao, Changcun Bao, Chaohu Liu, Huang Chen, Kun Yin, Hao Liu, Yinsong Liu, Deqiang Jiang, and Xing Sun. Attention where it matters: Rethinking visual document understanding with selective region concentration. In *ICCV*, pages 19517–19527, 2023. 2, 3, 5

[6] Hiuyi Cheng, Peirong Zhang, Sihang Wu, Jiaxin Zhang, Qiyuan Zhu, Zecheng Xie, Jing Li, Kai Ding, and Lianwen Jin. M6doc: A large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis. In *CVPR*, pages 15138–15147, 2023. 4

[7] Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. Document ai: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*, 2021. 1, 6

[8] Cheng Da, Chuwei Luo, Qi Zheng, and Cong Yao. Vision grid transformer for document layout analysis. In *ICCV*, pages 19462–19472, 2023. 1, 2, 4

[9] Cheng Da, Peng Wang, and Cong Yao. Multi-granularity prediction with learnable fusion for scene text recognition. *arXiv preprint arXiv:2307.13244*, 2023. 2

[10] Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, Curtis Wigington, and Vlad Morariu. End-to-end document recognition and understanding with dessurt. In *ECCV*, pages 280–296. Springer, 2022. 2

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[12] Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Nikolaos Barmpalios, Rajiv Jain, Ani Nenkova, and Tong Sun. Unified pretraining framework for document understanding. In *NeurIPS*, 2021. 1, 2

[13] Zhangxuan Gu, Changhua Meng, Ke Wang, Jun Lan, Weiqiang Wang, Ming Gu, and Liqing Zhang. Xylayoutlm: Towards layout-aware multimodal networks for visually-rich document understanding. In *CVPR*, pages 4583–4592, 2022. 1, 2, 3, 4

[14] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *ICDAR*, pages 991–995. IEEE, 2015. 6

[15] Jiabang He, Lei Wang, Yi Hu, Ning Liu, Hui Liu, Xing Xu, and Heng Tao Shen. Icl-d3ie: In-context learning with diverse demonstrations updating for document information extraction. *ICCV*, 2023. 1, 5, 6, 7

[16] Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *AAAI*, 2022. 1, 2

[17] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *ACM Multimedia*, 2022. 1, 2, 3, 4, 6, 7

[18] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. ICDAR2019 competition on scanned receipt OCR and information extraction. In *ICDAR*. IEEE, 2019. 1, 6

[19] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents, 2019. 1, 6

[20] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *ECCV*, pages 498–517. Springer, 2022. 2

[21] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *NeurIPS*, 35:22199–22213, 2022. 2, 5

[22] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *ICML*, pages 18893–18912. PMLR, 2023. 2, 3

[23] Chenliang Li, Bin Bi, and Ming Yan. StructuralLM: Structural pre-training for form understanding. In *ACL*, 2021. 1, 2

[24] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*, 2020. 4, 6

[25] Peizhao Li, Jiuxiang Gu, and Jason Kuen. SelfDoc: Self-supervised document representation learning. In *CVPR*, pages 5652–5660, 2021. 1, 2

[26] Yulin Li, Yuxi Qian, Yuechen Yu, Xiameng Qin, Chengquan Zhang, Yan Liu, Kun Yao, Junyu Han, Jingtuo Liu, and Errui Ding. Structext: Structured text understanding with multi-modal transformers. In *ACM Multimedia*, pages 1912–1920, 2021. 1, 2, 3, 4

[27] Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):919–931, 2022. 2

[28] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 3, 7

[29] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 2, 3

[30] Yuliang Liu, Zhang Li, Hongliang Li, Wenwen Yu, Yang Liu, Biao Yang, Mingxin Huang, Dezhi Peng, Mingyu Liu, Mingrui Chen, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. On the hidden mystery of ocr in large multimodal models, 2023. 3, 6

[31] Chuwei Luo, Guozhi Tang, Qi Zheng, Cong Yao, Lianwen Jin, Chenliang Li, Yang Xue, and Luo Si. Bi-vldoc: Bidirectional vision-language modeling for visually-rich document understanding. *arXiv preprint arXiv:2206.13155*, 2022. 3, 4

[32] Chuwei Luo, Changxu Cheng, Qi Zheng, and Cong Yao. Geolayoutlm: Geometric pre-training for visual information extraction. In *CVPR*, pages 7092–7101, 2023. 1, 2, 3, 4

[33] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, pages 2200–2209, 2021. 1, 6

[34] Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, et al. Fetaqa: Freeform table question answering. *TACL*, 10:35–49, 2022. 6

[35] OpenAI. Introducing chatgpt. `https://openai.com/blog/chatgpt`, 2022. 1, 3

[36] R OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023. 1, 3

[37] 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 *Workshop on Document Intelligence at NeurIPS 2019*, 2019. 1, 6

[38] Qiming Peng, Yinxu Pan, Wenjin Wang, Bin Luo, Zhenyu Zhang, Zhengjie Huang, Teng Hu, Weichong Yin, Yongfeng Chen, Yin Zhang, Shikun Feng, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. In *EMNLP findings*, 2022. 1, 2

[39] Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Jiaqi Mu, Hao Zhang, and Nan Hua. Lmdx: Language model-based document information extraction and localization. *arXiv preprint arXiv:2309.10952*, 2023. 1, 3

[40] Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. DocLayNet: A large human-annotated dataset for document-layout segmentation. In *SIGKDD*. ACM, 2022. 6

[41] Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. Going full-tilt boogie on document understanding with text-image-layout transformer. In *ICDAR*, pages 732–747. Springer, 2021. 1, 2

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2

[43] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 35:25278–25294, 2022. 2

[44] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016. 1

[45] Yongxin Shi, Dezhi Peng, Wenhui Liao, Zening Lin, Xinhong Chen, Chongyu Liu, Yuyi Zhang, and Lianwen Jin. Exploring ocr capabilities of GPT-4V(ision): A quantitative and in-depth evaluation. *arXiv preprint arXiv:2310.16809*, 2023. 3, 6

[46] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *ECCV*, pages 742–758. Springer, 2020. 2

[47] Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. Visualmrc: Machine reading comprehension on document images. In *AAAI*, pages 13878–13888, 2021. 1, 6

[48] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023. 2

[49] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[50] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. 1, 2, 7

[51] Yi Tu, Ya Guo, Huan Chen, and Jinyang Tang. LayoutMask: Enhance text-layout interaction in multi-modal pre-training for document understanding. In *ACL*, 2023. 3, 4

[52] Jiapeng Wang, Lianwen Jin, and Kai Ding. Lilt: A simple yet effective language-independent layout transformer for structured document understanding. In *ACL*, 2022. 1, 2

[53] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35:24824–24837, 2022. 2, 5

[54] Yiheng Xu, Minghao Li, Lei Cui, and Shaohan Huang. LayoutLM: Pre-training of text and layout for document image understanding. In *KDD*, pages 1192–1200, 2020. 1, 2

[55] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *ACL*, 2021. 1, 2, 3, 4

[56] 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*, 9, 2023. 1, 3

[57] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, et al. mplug-docowl: Modularized multimodal large language model for document understanding. *arXiv preprint arXiv:2307.02499*, 2023. 1, 2, 3, 4, 7

[58] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. mplug-owl: Modularization empowers large language models with multimodality, 2023. 3

[59] Yuechen Yu, Yulin Li, Chengquan Zhang, Xiaoqiang Zhang, Zengyuan Guo, Xiameng Qin, Kun Yao, Junyu Han, Errui Ding, and Jingdong Wang. Structextv2: Masked visual-textual prediction for document image pre-training. In *ICLR*, 2023. 1, 2

[60] Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. Llavar: Enhanced visual instruction tuning for text-rich image understanding. *arXiv preprint arXiv:2306.17107*, 2023. 1, 2, 3, 4, 7

[61] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 2, 6, 7

[62] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *ICDAR*, pages 1015–1022. IEEE, 2019. 4, 6

[63] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: data, model, and evaluation. In *ECCV*, pages 564–580. Springer, 2020. 6

[64] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017. 1

[65] 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, 2023. 6