A BOUNDING BOX IS WORTH ONE TOKEN - INTERLEAVING LAYOUT AND TEXT IN A LARGE LANGUAGE MODEL FOR DOCUMENT UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, many studies have demonstrated that exclusively incorporating OCR-derived text and spatial layouts with large language models (LLMs) can be highly effective for document understanding tasks. However, existing methods that integrate spatial layouts with text have limitations, such as producing overly long text sequences or failing to fully leverage the autoregressive traits of LLMs. In this work, we introduce *Interleaving **Lay**out and **Text** in a **L**arge **L**anguage **M**odel (LayTextLLM)* for document understanding. In particular, LayTextLLM projects each bounding box to a single embedding and interleaves it with text, efficiently avoiding long sequence issues while leveraging autoregressive traits of LLMs. LayTextLLM not only streamlines the interaction of layout and textual data but also shows enhanced performance in Key Information Extraction (KIE) and Visual Question Answering (VQA). Comprehensive benchmark evaluations reveal significant improvements, with a 27.2% increase on KIE tasks and 12.0% on VQA tasks compared to previous state-of-the-art document understanding MLLMs, as well as a 15.1% improvement over other SOTA OCR-based LLMs on KIE tasks.

1 INTRODUCTION

Recent research has increasingly focused on applying Large Language Models (LLMs) (Achiam et al., 2023; Yang et al., 2023; Team et al., 2023; Anthropic, 2024; Reid et al., 2024; Bai et al., 2023; Lu et al., 2024a; Young et al., 2024; Feng et al., 2023a;b; Hu et al., 2024; Liu et al., 2024c; Tang et al., 2024a; Chen et al., 2024; Dong et al., 2024; Li et al., 2024; Liu et al., 2024a; Zhu et al., 2024; Yang et al., 2024; Tang et al., 2024b; Liu et al., 2023; Lu et al., 2024b; 2023) to document-oriented Visual Question Answering (VQA) and Key Information Extraction (KIE) scenarios. Efforts to build a text-sensitive MultiModal Large Language Models (MLLMs) based on existing LLMs, particularly aimed at enhancing Visually Rich Document Understanding (VRDU), have made significant progress (Ye et al., 2023; Liu et al., 2024c; Bai et al., 2023). Although existing MLLMs show promising results in document understanding, they often encounter challenges related to image resolution. When the input image is of low resolution, it is too blurry to extract visual features effectively. Conversely, high-resolution images require additional computational resources to capture detailed textual information (Liu et al., 2024c).

Concurrently, another line of research employs off-the-shelf OCR tools to extract text and spatial layouts, which are then combined with LLMs to address VRDU tasks. These approaches assume that *most valuable information for document comprehension can be derived from the text and its spatial layouts, viewing spatial layouts as "lightweight visual information" (Wang et al., 2023)*. Following this premise, several studies (Liu et al., 2024c; Perot et al., 2023; Luo et al., 2024; Chen et al., 2023a; He et al., 2023) have explored various approaches that integrate spatial layouts with text for LLMs, achieving results that are competitive with, or even surpass, those of MLLMs.

The most natural method to incorporate layout information is by treating spatial layouts as tokens, which allows for the seamless interleaving of text and layout into a unified text sequence (Perot et al., 2023; Chen et al., 2023a; He et al., 2023). For example, Perot et al. (2023) employ format such as *"HARRISBURG 78|09"* to represent OCR text and corresponding layout, where *"HARRISBURG"* is OCR text and *"78|09"* indicates the mean of the horizontal and vertical coordinates, respectively.

Similarly, He et al. (2023) use *"[x_min, y_min, x_max, y_max]"* to represent layout information. These approaches can effectively take advantage of autoregressive characteristics of LLMs and is known as the *"coordinate-as-tokens"* scheme (Perot et al., 2023). In contrast, DocLLM (Wang et al., 2023) explores interacting spatial layouts with text through a disentangled spatial attention mechanism that captures cross-alignment between text and layout modalities.

However, we argue that both of the previous approaches have limitations. As shown in Fig. 1, coordinate-as-tokens significantly increases the number of tokens. Additionally, to accurately comprehend coordinates and enhance zero-shot capabilities, this scheme often requires few-shot in-context demonstrations and large-scale language models, such as ChatGPT Davinci-003 (175B) (He et al., 2023), which exacerbates issues related to sequence length and GPU resource demands. Meanwhile, although DocLLM does not increase sequence length and integrates spatial layouts through attention, its performance is limited.

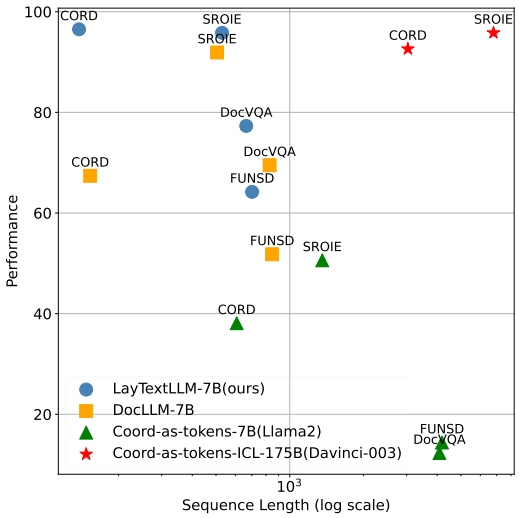


Figure 1: The performance against input sequence length of different datasets across various OCR-based methods where data is from Tab. 2 and 5.

To address these problems, this paper explores a simple yet effective approach to enhance the interaction between spatial layouts and text — *Interleaving **Lay**out and **Text** in a **L**arge **L**anguage **M**odel (LayTextLLM)* for document understanding. Adhering to the common practice of interleaving any modality with text (Huang et al., 2023; Peng et al., 2023; Dong et al., 2024), we specifically apply this principle to spatial layouts. In particular, we maps each bounding box to a single embedding, which is then interleaved with its corresponding text. Then we propose a tailored pre-training task—Layout-aware Next Token Prediction—a completely self-supervised task that enhances the alignment between layout and textual modalities without using synthetic data. Finally, through the proposed Shuffled-OCR Supervised Fine-tuning, LayTextLLM significantly improves performance on downstream document-related VQA and KIE tasks. As shown in Fig. 1, LayTextLLM significantly outperforms the 175B models, while only slightly increasing or even reducing the sequence length compared to DocLLM. Our contributions can be listed as follows:

- We propose LayTextLLM for document understanding. To the best of the authors' knowledge, this is the first work to employ a unified embedding approach (Sec. 3.1.1) that interleaves spatial layouts directly with textual data within a LLM. By representing each bounding box with one token, LayTextLLM efficiently addresses sequence length issues brought by coordiante-as-tokens while fully leveraging autoregressive traits for enhanced document understanding.
- We propose two tailored training tasks: (1) Layout-aware Next Token Prediction (Sec. 3.2.1), a completely self-supervised training task to enhance the alignment between layout and textual modality; (2) Shuffled-OCR Supervised Fine-tuning task (Sec. 3.2.2) to better elicit the model generalizability in downstream tasks.
- Comprehensive experimental results demonstrate quantitatively that LayTextLLM significantly outperforms previous state-of-the-art (SOTA) OCR-free MLLMs by a large margin in zero-shot scenarios, particularly in KIE tasks with an improvement of 27.0%. Additionally, we illustrate that LayTextLLM competes effectively or even surpasses previous SOTA OCR-based methods in both zero-shot and SFT scenarios. Specifically, it surpasses DocLLM by 19.8% on VQA and 15.5% on KIE tasks (Sec. 4).
- Extensive ablations and visualizations demonstrate the utility of the proposed component, with analysis showing that LayTextLLM not only improves performance but also reduces input sequence length compared to current OCR-based models.

2 RELATED WORK

2.1 OCR-BASED LLMS FOR DOCUMENT UNDERSTANDING

Early document understanding methods (Hwang et al., 2020; Xu et al., 2020; 2021; Hong et al., 2022; Tang et al., 2022) tend to solve the task in a two-stage manner, *i.e.*, first reading texts from input document images using off-the-shelf OCR engines and then understanding the extracted texts. Considering the advantages of LLMs (*e.g.*, high generalizability), some recent methods endeavor to combine LLMs with OCR-derived results to solve document understanding. For example, inspired by the "coordinate-as-tokens" scheme proposed in ICL-D3IE (Perot et al., 2023), He et al. (2023) propose to use *"[x_min, y_min, x_max, y_max]"* to introduce the layout information, which can fuse the layout information and texts into a unified text sequence and fully exploit the autoregressive merit of LLMs. To reinforce the layout information while avoiding increasing the number of tokens, DocLLM (Wang et al., 2023) designs a disentangled spatial attention mechanism to capture cross-alignment between text and layout modalities. Recently, LayoutLLM (Luo et al., 2024) utilizes the pre-trained layout-aware model (Huang et al., 2022), to insert the visual information, layout information and text information. However, the aforementioned methods neither suffer from the computational overhead leading by the increasing tokens or hardly take advantage of autoregressive characteristics of LLMs. Thus, it is an urgent problem to address how to better incorporate layout information without significantly increasing the number of tokens.

2.2 OCR-FREE MLLMS FOR DOCUMENT UNDERSTANDING

Another approach to solve document understanding tasks is the OCR-free method. Benefiting from the end-to-end training framework, it involves processing the text content of documents directly, without relying on OCR engines. Donut (Kim et al., 2022) first presents an OCR-free method through mapping a text-rich document image into the desired answers. Pix2Struct (Lee et al., 2023) is trained to parse masked screenshots of web pages into simplified HTML, where variable resolution inputs are supported. While these approaches eliminate the need for OCR tools, they still necessitate task-specific fine-tuning. With the increasing popularity of LLMs/MLLMs (Feng et al., 2023b; Hu et al., 2024; Liu et al., 2024c; Tang et al., 2024a; Chen et al., 2024; Dong et al., 2024; Li et al., 2024; Liu et al., 2024a), various methods are proposed to solve the document understanding task through explicitly training models on visual text understanding datasets and fine-tuning them with instructions to perform a zero-shot prediction. LLaVAR (Zhang et al., 2023) and UniDoc (Feng et al., 2023b) are notable examples that expand upon the document-oriented VQA capabilities of LLaVA (Liu et al., 2024b) by incorporating document-based tasks. These models pioneer the use of MLLMs for predicting texts and coordinates from document images, enabling the development of OCR-free document understanding methods. Additionally, DocPedia (Feng et al., 2023a) operates document images in the frequency domain, allowing for higher input resolution without increasing the input sequence length. Recent advancements in this field, including mPLUG-DocOwl (Ye et al., 2023), Qwen-VL (Bai et al., 2023), and TextMonkey (Liu et al., 2024c), leverage publicly available document-related VQA datasets to further enhance the document understanding capability. Although these OCR-free methods have exhibited their advantages, they still struggle with the high-resolution input to reserve more text-related details.

3 METHOD

In this section, we present our LayTextLLM. First, we introduce a innovative Spatial Layout Projector (Sec. 3.1.1) converts four-dimensional layout coordinates into a single-token embedding. To reduce parameter overhead, we apply Partial Low-Rank Adaptation (Sec. 3.1.2). We also introduce two training tasks: Layout-aware Next Token Prediction (Sec. 3.2.1) to align layouts with text during pre-training, and Shuffled-OCR Supervised Fine-tuning (Sec. 3.2.2) to enhance the generalizability of the model. An illustration of our approach is shown in Fig. 2.

3.1 MODEL ARCHITECTURE

LayTextLLM is built on the Llama2-7B-base model, which was originally designed to accept only text inputs (Touvron et al., 2023; Gao et al., 2023). To enable the model to interleave spatial layouts

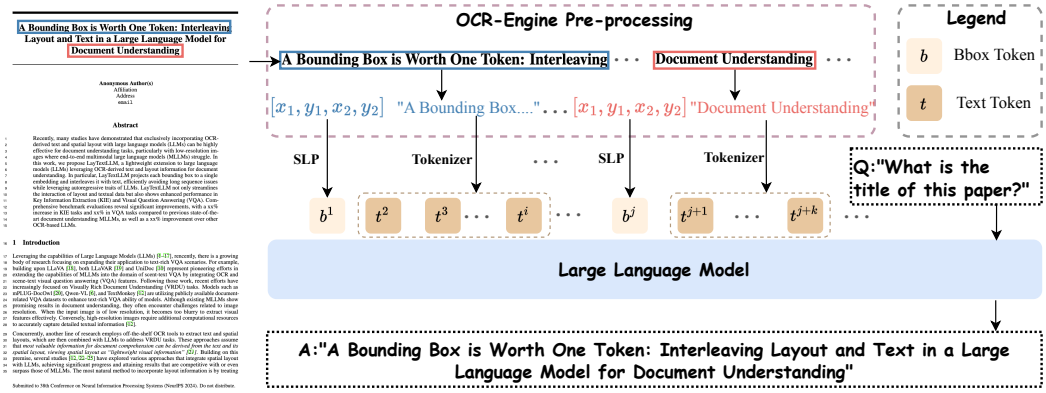


Figure 2: An overview of LayTextLLM incorporates interleaving bounding box tokens (b^i) with text tokens (t^i), where the superscripts represent the sequence positions of the tokens.

with text, we introduce a novel Spatial Layout Projector. This projector converts coordinates into bounding box tokens. We also adopt the Partial Low-Rank Adaptation, a minimally invasive method to incorporate additional modalities while preserving the LLM's inherent knowledge intact.

3.1.1 Spatial Layout Projector (SLP)

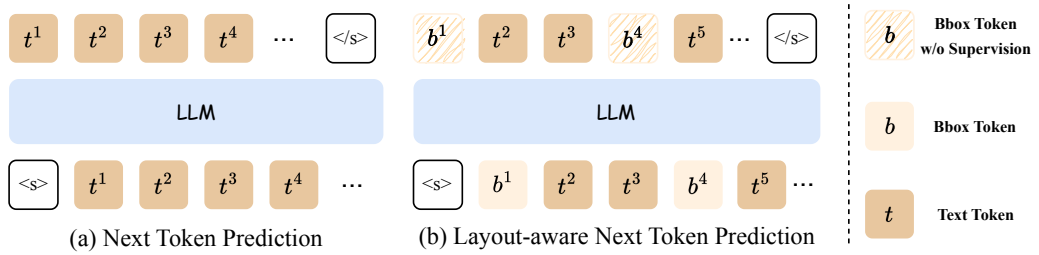
A key innovation in LayTextLLM is the Spatial Layout Projector (SLP), which transforms a spatial layout into a singular bounding box token. This enhancement enables the model to process both spatial layouts and textual inputs simultaneously. Each OCR-derived spatial layout is represented by a bounding box defined by four-dimensional coordinates $[x_1, y_1, x_2, y_2]$, these coordinates represent the normalized minimum and maximum horizontal and vertical extents of the box, respectively. The SLP maps these coordinates into a high-dimensional space that the language model can process as a single token. The process can be computed as $z = W \cdot c + b$, where $c \in \mathbb{R}^4$ is the vector of the bounding box coordinates. $W \in \mathbb{R}^{d \times 4}$ is a weight matrix with d represents the dimension of the embedding, $b \in \mathbb{R}^{d \times 1}$ is a bias vector, z is the resulting bounding box token represented as an d-dimensional embedding. As illustrated in Fig. 2, the resulting bounding box token z will be interleaved with corresponding textual embeddings to put into LLMs. Note that the SLP is shared by all bounding box tokens so very limited number of parameters are introduced.

3.1.2 Layout Partial Low-Rank Adaptation

After using the SLP to generate bounding box tokens and a tokenizer to produce text tokens, these two modalities are then interacted using a Layout Partial Low-Rank Adaptation (P-LoRA) module in LLMs. P-LoRA, introduced in InternLM-XComposer2 (Dong et al., 2024), is originally used to adapt LLMs to the visual modality. It applies plug-in low-rank modules specified to the visual tokens, which adds minimal parameters while preserving the LLMs inherent knowledge.

Formally, for a linear layer in the LLM, the original weights $W_O \in \mathbb{R}^{C_{out} \times C_{in}}$ and bias $B_O \in \mathbb{R}^{C_{out}}$ are specified for input and output dimensions C_{in} and C_{out}. P-LoRA modifies this setup by incorporating two additional matrices, $W_A \in \mathbb{R}^{C_r \times C_{in}}$ and $W_B \in \mathbb{R}^{C_{out} \times C_r}$. These matrices are lower-rank, with C_r being considerably smaller than both C_{in} and C_{out}, and are specifically designed to interact with new modality tokens, which in our case are bounding box tokens. For example, given an input $x = [x_b, x_t]$ comprising of bounding box tokens (x_b) and textual tokens (x_t) is fed into the system, the forward process is as follows, where $\hat{x}_t, \hat{x}_b$ and $\hat{x}$ are outputs:

$$\begin{aligned} \hat{x}_t &= W_0 x_t + B_0 \\ \hat{x}_b &= W_0 x_b + W_B W_A x_b + B_0 \\ \hat{x} &= [\hat{x}_b, \hat{x}_t] \end{aligned} \tag{1}$$



(a) Next Token Prediction (b) Layout-aware Next Token Prediction

Figure 3: Comparison of Layout-aware Next Token Prediction and normal Next Token Prediction.

3.2 Training Procedure

LayTextLLM is trained with innovative layout-aware training procedure, which consists of two stages: Layout-aware Next Token Prediction pre-training and Shuffled-OCR Supervised Fine-tuning.

3.2.1 Layout-aware Next Token Prediction

Inspired by the next token prediction commonly used in current LLM pre-training (Achiam et al., 2023; Yang et al., 2023; Team et al., 2023; Anthropic, 2024; Reid et al., 2024; Bai et al., 2023; Lu et al., 2024a), we propose the Layout-aware Next Token Prediction (LNTP). Fig. 3 presents the contrast of the proposed Layout-aware Next Token Prediction and the conventional next token prediction task. The traditional next token prediction (Fig. 3(a)) relies solely on the textual content, predicting each subsequent token based on the prior sequence of tokens without considering their spatial layouts. Layout-aware next token prediction (Fig. 3(b)), however, interleaves the spatial information encoded by SLP (*i.e.*, b^i) with the text tokens (*i.e.*, t^i). This integration considers both the content and its layout within the document, leading to a richer, more precise understanding of both the structure and the content.

Similarly, primary objective of LNTP is to maximize the likelihood of its predictions for the next token. Thus the loss function is defined as

$$\mathcal{L} = -\frac{1}{T}\sum_{i=1}^{T} \log P\left(t^i \mid t^1, t^2, \ldots, t^{i-1}\right) \tag{2}$$

where $P\left(t^i \mid t^1, t^2, \ldots, t^{i-1}\right)$ represents the probability of i^{th} token t^i given the sequence of preceding tokens $t^1, t^2, \ldots, t^{i-1}$, as predicted by the model. Note that we compute the loss only for text tokens, excluding bounding box tokens. During pre-training, our goal is to enhance the alignment between spatial layouts and textual modality, while preserving the LLM's inherent knowledge as much as possible. Thus, we freeze the LLMs and only update the parameters of SLP and P-LoRA.

It is important to note that the proposed Layout-aware Next Token Prediction is a completely self-supervised pre-training procedure, unlike previous works that require human annotations of document structure data or synthetic data generated by larger LLMs such as GPT-4 (Luo et al., 2024). Thus, LNTP facilitates the creation of large-scale, high-fidelity pre-training datasets at minimal cost.

Although the loss is not directly computed for the bounding box tokens, the SLP is still effectively trained. In Fig. 3(b), the prediction of t^2 relies on the hidden state of b^1, allowing the supervision signal to backpropagate through the SLP. Note that the goal is to understand the bounding boxes, not to generate them, making it unnecessary to apply a loss function to the bounding box tokens.

3.2.2 Shuffled-OCR Supervised Fine-tuning

OCR engines typically process text from top to bottom and left to right. This order is also adopted as the input sequence for current OCR-based LLMs (Wang et al., 2023; Luo et al., 2024). However, modern LLMs often exhibit a strong inductive bias toward the positions of input tokens, influenced by designs such as Rotary Position Embeddings (RoPE) (Su et al., 2024). Specifically, tokens that are close together in the input sequence are likely to receive higher attention scores, which is

advantageous for processing standard text sequences. Such inductive bias brings advantages and disadvantages.

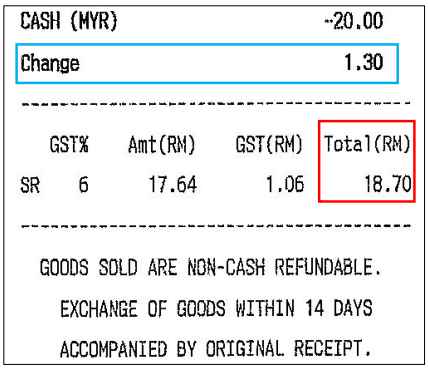


Figure 4: Receipt layout example.

Consider the example illustrated in Fig. 4, where the OCR input text reads: *" ... Change, 1.30, GST%, Amt(RM), GST(RM), Total(RM), SR, 6, 17.64, 1.06, 18.70 ... "*. If the question posed is *"What is the value of the field Change?"* (highlighted in a blue box), the model easily identifies *"1.30"* as it is closely positioned to the word *"Change"* in the sequence. However, for a more challenging query like *"What is the value of the field Total(RM)?"* (highlighted in a red box), the model struggles to determine the correct answer due to the presence of multiple subsequent numbers closed to *"Total(RM)"*. LayTextLLM integrates spatial layouts with textual data, reducing reliance on input sequence order. Thus, we posit that shuffling the OCR input order could enhance the resilience of LayTextLLM in discerning relevant information irrespective of token proximity in the sequence.

Specifically, we propose Shuffled-OCR Supervised Fine-tuning (SSFT) that randomly shuffles the order of OCR-derived text in a certain proportion of examples. The range of exploration for the shuffling ratio can be found in Tab. 7 and 20% shuffled ratio is applied. The training objective is equivalent to predicting the next tokens, but in this scenario, only the tokens of the response are used to compute loss. During SSFT, we unfreeze all parameters including those of LLMs. Experimental results in Section 4.6 demonstrate that utilizing SSFT can further enhance model performance, making it more robust to disruptions in input token order.

4 Experiments

4.1 Datasets

LNTP Data In training process, we exclusively use open-source data to facilitate replication. We collect data from two datasets for pre-training: (1) **IIT-CDIP Test Collection 1.0** (Lewis et al., 2006) and (2) **DocBank** (Li et al., 2020). The IIT-CDIP Test Collection 1.0 comprises an extensive repository of more than 16 million document pages. DocBank consists of 500K documents, each presenting distinct layouts with a single page per document. For training efficiency, we sue the entire DocBank dataset and only subsample 5 million pages from the IIT-CDIP collection 1.0.

SSFT data For document-oriented VQA, we select **Document Dense Description (DDD)** and **Layout-aware SFT** data used in Luo et al. (2024), which are two synthetic datasets generated by GPT-4. Besides, **DocVQA** (Mathew et al., 2021), **InfoVQA** (Mathew et al., 2022), **ChartQA** (Masry et al., 2022), **VisualMRC** (Tanaka et al., 2021) is included following (Liu et al., 2024c). For KIE task, we select **SROIE** (Huang et al., 2019), **CORD** (Park et al., 2019), **FUNSD** (Jaume et al., 2019), **POIE** (Kuang et al., 2023) datasets following (Wang et al., 2023; Luo et al., 2024; Liu et al., 2024c).

4.2 Implementation Detail

The LLM component of LayTextLLM is initialized from the Llama2-7B-base (Touvron et al., 2023), which is a widely-used backbone. Other parameters including SLP and P-LoRA are randomly initialized. During pre-training, the LLM is frozen, and the parameters of SLP and P-LoRA modules are updated. During SFT, all parameters are fine-tuned. Detailed setup can be found in Appendix B.

We implemented LayTexLLM using Llama2-7B, consistent with previous OCR-based methods like DocLLM (Wang et al., 2023), which also use Llama2-7B. We also replicated the results of the coor-as-tokens scheme using Llama2-7B for consistency. Noting the LayoutLLM (Luo et al., 2024) utilizes Llama2-7B and Vicuna 1.5 7B, which is fine-tuned from Llama2-7B. Thus, for the majority of our comparisons, the models are based on the same or similar LLM backbones, allowing for a fair comparison between approaches.

Other MLLM baselines use backbones like Qwen-VL (Bai et al., 2023), InternLM (Dong et al., 2024), and Vicuna (Chiang et al., 2023), all with at least 7B parameters, excluding the visual encoder. This also makes the comparison fair, at least in terms of model size.

In this work, we configure three versions of LayTextLLM for a side-by-side comparison under different training settings. Aligned with the terminology used in (Luo et al., 2024), "zero-shot" refers to models trained solely via Supervised Self-Fine-Tuning (SSFT) using Document Dense Description (DDD) and Layout-aware SFT data. Our first version, **LayTextLLM**$_{zero}$, follows this zero-shot setup, trained exclusively with DDD and Layout-aware SFT data (Luo et al., 2024). Building upon this, our second version, **LayTextLLM**$_{vqa}$, introduces the DocVQA and InfoVQA training sets, as in Liu et al. (2024c), while retaining the same SSFT process. Lastly, we create **LayTextLLM**$_{all}$, which, in addition to the VQA datasets, incorporates a comprehensive set of KIE datasets—FUNSD, CORD, POIE, SROIE, ChartQA, and VisualMRC (Wang et al., 2023). All these versions share the same pre-trained LayTextLLM weights, with the distinction that "supervised" (SFT) versions (**LayTextLLM**$_{vqa}$ and **LayTextLLM**$_{all}$) include additional downstream training sets.

We use word-level OCR from the respective datasets for a fair comparison, with the exception of the ChartQA dataset, which does not provide OCR.

4.3 Baselines

OCR-free baselines In the category of OCR-free MLLMs, we have chosen the following SOTA models as our strong baselines due to their superior performance in both document-oriented VQA and KIE tasks. These include **UniDoc** (Feng et al., 2023b), **DocPedia** (Feng et al., 2023a), **Monkey** (Li et al., 2023), **InternVL** (Chen et al., 2023b), **InternLM-XComposer2** (Dong et al., 2024), **TextMonkey**, and **TextMonkey**$_{+}$ (Liu et al., 2024c).

OCR-based baselines For OCR-based baseline models, we implemented a basic approach using only OCR-derived text as input. This was done using two versions: **Llama2-7B-base** and **Llama2-7B-chat**. We also adapted the coordinate-as-tokens scheme from He et al. (2023) for these models, resulting in two new variants: **Llama2-7B-base**$_{coor}$ and **Llama2-7B-chat**$_{coor}$. It is important to note that we did not employ the ICL strategy with these models, as it would significantly exceed their maximum sequence length constraints. Additionally, we included results from a stronger baseline using the ChatGPT Davinci-003 (175B) model (He et al., 2023), termed **Davinci-003-175B**$_{coor}$. One other recent SOTA OCR-based approach, **DocLLM** (Wang et al., 2023) is also considered in our analysis. Finally, **LayoutLLM** and **LayoutLLM**$_{CoT}$ (Luo et al., 2024), which integrates visual cues, text and layout is also included.

4.4 Evaluation Metrics

To ensure a fair comparison with OCR-free methods, we adopted the accuracy metric, where a response from the model is considered correct if it fully captures the ground truth. This approach aligns with the evaluation criteria described by (Liu et al., 2024c; Feng et al., 2023a;b). To further enhance the comparability with other OCR-based methods, we conducted additional evaluations using original metrics specific to certain datasets, such as F1 score (Wang et al., 2023; He et al., 2023), ANLS (Gao et al., 2019; Wang et al., 2023; Luo et al., 2024) and CIDEr (Vedantam et al., 2015; Wang et al., 2023).

4.5 Quantitative Results

Comparison with SOTA OCR-free Methods The experimental results shown in Tab. 1 demonstrate the outstanding performance of the LayTextLLM series across various tasks. Note that the results for ChartQA are reported in Appendix E due to concerns about fairness in comparison, as the dataset does not include OCR-derived results and we used in-house OCR tools instead. Firstly, LayTextLLM$_{zero}$ significantly outperforms previous SOTA OCR-free methods, such as TextMonkey (Liu et al., 2024c), in zero-shot capabilities, even when these methods use the training set of the dataset. For example, in the DocVQA and InfoVQA datasets, LayTextLLM$_{zero}$ achieves accuracies of 72.1% and 35.7%, respectively, which are markedly higher than existing OCR-free methods such as TextMonkey and InternLM-XComposer2. When fine-tuned with corresponding datasets, LayTextLLM shows even greater performance improvements, particularly in document-oriented

	Document-Oriented VQA			KIE			
	DocVQA	InfoVQA	Avg	FUNSD	SROIE	POIE	Avg
Metric	*Accuracy %*						
OCR-free							
UniDoc (Feng et al., 2023b)	7.7	14.7	11.2	1.0	2.9	5.1	3.0
DocPedia (Feng et al., 2023a)	47.1*	15.2*	31.2	29.9	21.4	39.9	30.4
Monkey (Li et al., 2023)	50.1*	25.8*	38.0	24.1	41.9	19.9	28.6
InternVL (Chen et al., 2023b)	28.7*	23.6*	26.2	6.5	26.4	25.9	19.6
InternLM-XComposer2 (Dong et al., 2024)	39.7	28.6	34.2	15.3	34.2	49.3	32.9
TextMonkey (Liu et al., 2024c)	64.3*	28.2*	46.3	32.3	47.0	27.9	35.7
TextMonkey$_+$ (Liu et al., 2024c)	66.7*	28.6*	47.7	42.9	46.2	32.0	40.4
Text + Coordinates							
LayTextLLM$_{zero}$ (Ours)	72.1	35.7	53.9	**47.5**	**86.4**	68.9	**67.6**
LayTextLLM$_{vqa}$ (Ours)	**77.2***	**42.1***	**59.7**	48.8	75.7	**70.6**	65.0

Table 1: Comparison with SOTA OCR-free MLLMs. * denotes the use of the dataset's training set.

	Document-Oriented VQA			KIE			
	DocVQA	VisualMRC	Avg	FUNSD	CORD	SROIE	Avg
Metric	*ANLS % / CIDEr*			*F-score %*			
Text							
Llama2-7B-base	34.0	182.7	108.3	25.6	51.9	43.4	40.3
Llama2-7B-chat	20.5	6.3	13.4	23.4	51.8	58.6	44.6
Text + Coordinates							
Llama2-7B-base$_{coor}$ (He et al., 2023)	8.4	3.8	6.1	6.0	46.4	34.7	29.0
Llama2-7B-chat$_{coor}$ (He et al., 2023)	12.3	28.0	20.1	14.4	38.1	50.6	34.3
Davinci-003-175B$_{coor}$ (He et al., 2023)	-	-	-	-	92.6	95.8	-
DocLLM (Wang et al., 2023)	69.5*	264.1*	166.8	51.8*	67.6*	91.9*	70.3
LayTextLLM$_{zero}$ (Ours)	65.5	200.2	132.9	47.2	77.2	83.7	69.4
LayTextLLM$_{vqa}$ (Ours)	75.6*	179.5	127.6	52.6	70.7	79.3	67.5
LayTextLLM$_{all}$ (Ours)	**77.2***	**277.8***	**177.6**	**64.0***	**96.5***	**95.8***	**85.4**

Table 2: Comparison with other OCR-based methods. * denotes the use of the dataset's training set.

VQA datasets. Specifically, its accuracies on DocVQA and InfoVQA increase to 77.2% and 42.1%, respectively, demonstrating the model's strong ability to leverage task-specific data. Additionally, LayTextLLM$_{zero}$ excels in KIE datasets, particularly on the SROIE and POIE datasets, achieving accuracies of 86.4% and 68.9%, respectively. These results significantly surpass those of previous SOTA OCR-free model (*i.e.*, TextMonkey$_+$) by margins of 40.5% and 34.1%, respectively. This significant performance gain is likely due to these datasets containing low-resolution images that are too blurred for current MLLMs to extract visual features, whereas LayTextLLM shows robustness in such challenging scenarios.

Comparison with SOTA OCR-based Methods For comprehensive comparison, we have also conducted corresponding experiments to align with OCR-based methods (Wang et al., 2023; Luo et al., 2024). The experimental results presented in Tab. 2 showcase significant performance improvements achieved by LayTextLLM models compared to pure OCR-based SOTA methods such as DocLLM (Wang et al., 2023). Specifically, when comparing with DocLLM, LayTextLLM$_{zero}$ demonstrates notably superior performance, with even its zero-shot capabilities being competitive with SFT approaches.

We believe that the subpar performance of DocLLM is likely due to its use of cross-attention and the masked span pre-training tasks (Raffel et al., 2020), which fail to leverage the autoregressive features of LLMs effectively. Similarly, when contrasting with coordinate-as-tokens employed in Llama2-7B, LayTextLLM$_{zero}$ again outperforms significantly. This disparity in performance can be attributed to the following three reasons: (1) The coordinate-as-tokens approach tends to introduce an excessive number of tokens, often exceeding the pre-defined maximum length of Llama2-7B (*i.e.*, 4096). Consequently, this leads to a lack of crucial OCR information, resulting in hallucination and subpar performance. (2) When re-implementing the coordinate-as-tokens method with Llama2-7B, we did not introduce the ICL strategy, as it would contribute additional length to the input sequence.

	Document-Oriented VQA			KIE			
	DocVQA	VisualMRC	Avg	FUNSD$^-$	CORD$^-$	SROIE$^-$	Avg
Metric	*ANLS %*						
Visual + Text + Coordinates							
LayoutLLM (Luo et al., 2024)	72.3	-	-	74.0	-	-	-
LayoutLLM$_{CoT}$ (Luo et al., 2024)	74.2	**55.7**	**64.9**	79.9	63.1	72.1	71.7
Text							
Llama2-7B-base	34.0	25.4	29.7	42.1	46.7	60.6	49.8
Llama2-7B-chat	20.5	9.9	15.2	15.1	20.0	35.6	23.5
Text + Coordinates							
Llama2-7B-base$_{coor}$ (He et al., 2023)	8.4	6.7	7.5	4.3	33.0	47.2	28.1
Llama2-7B-chat$_{coor}$ (He et al., 2023)	12.3	12.2	12.2	11.9	6.4	39.4	19.2
LayTextLLM$_{zero}$ (Ours)	65.5	37.4	51.5	72.0	45.5	82.0	66.5
LayTextLLM$_{all}$ (Ours)	**77.2***	41.7*	59.5	**81.0***	**82.5***	**96.1***	**86.5**

Table 3: Comparison with LayoutLLM. $^-$ indicates that the cleaned test set used in Luo et al. (2024).

(3) The coordinate-as-tokens approach necessitates a considerably larger-sized LLM to comprehend the numerical tokens effectively.

In comparison to LayoutLLM (Luo et al., 2024), our approach exhibits discrepant performance in different tasks, as shown in Tab. 3. In zero-shot scenarios, we outperform LayoutLLM in most KIE datasets, validating our capability to leverage OCR-based results effectively. However, we fall short on document-oriented VQA tasks since answering some questions that are strongly related to vision information may challenge our approach. Two main reasons may well explain this performance discrepancy: (1) The visual encoder in LayoutLLM provides additional visual information. (2) LayoutLLM incorporates the Chain-of-Thought (CoT) mechanism to model contextual information while it is not used in our approach. However, when fine-tuned with tailored data, LayTextLLM significantly outperforms LayoutLLM, showcasing its strong ability to utilize task-specific data. More qualitative example demonstrates can be found in Appendix A.

4.6 Analysis

			Document-Oriented VQA				KIE				
SLP	P-LoRA	LNTP+SSFT	DocVQA	InfoVQA	VisualMRC	Avg	FUNSD	CORD	SROIE	POIE	Avg
			71.5	31.9	31.1	44.8	50.5	90.2	91.6	54.1	71.6
✓			74.7	35.7	32.5	47.6	55.1	94.9	94.6	68.3	78.2
✓	✓		76.5	38.0	30.6	48.4	54.3	**95.9**	**95.3**	**70.6**	79.0
✓	✓	✓	**78.8**	**42.7**	**34.4**	**52.0**	**63.0**	**95.9**	95.2	62.1	**79.1**

Table 4: Ablations on each component of LayTextLLM (Accuracy).

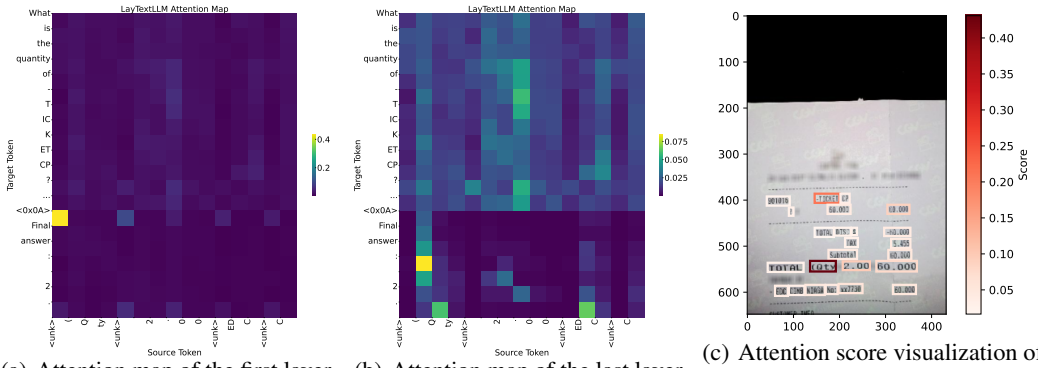
Ablations To better assess the utility of each component in LayTextLLM, an ablation study was conducted, the results of which are presented in Tab. 4. Detailed information on the training setup for all variants is provided in Appendix B. The results clearly show that incorporating interleaved spatial layouts and texts significantly enhances the performance, evidenced by a 2.8% improvement in VQA and a 6.6% increase in KIE compared to the plain version, indicating that SLP is a critical component. Furthermore, enabling P-LoRA results in a modest performance increase of 0.8% in both VQA and KIE tasks. Finally, the enabling of LNTP+SSFT leads to a significant improvement in VQA and KIE tasks, with an increase of 3.6% and 0.1%.

Advantage of Interleaving Layout and Text

We visualize the attention patterns between input and output tokens in Fig. 5. The attention pattern is insightful with the specific question, *"What is the quantity of - TICKET CP?<0x0A>"*

As shown in Fig. 5(a), when the model begins predicting the answer *"Final"*, *"<0x0A>"(newline symbol)* is heavily focusing on layout information, as seen by the significant attention on the bounding box embedding *"<unk>"* token before *"(Qty"*. This highlights the model's effort to orient itself spatially and understand the structural context of the tokens. At this stage, the model is developing a cognitive understanding of how the elements are laid out on the page. We extract and visualize the

attention scores that *"<0x0A>"* assigns to each bounding box in Fig. 5(c). The visualization shows that the model focuses most on *"Qty"*, followed by *"-TICKET"* and *"2.00"*, which reflects the layout information essential for making the prediction. In the final layer (Fig. 5(b)), the model's attention shifts dramatically towards the *"Qty"* token, which holds the semantic meaning necessary to answer the question. This shift from layout-based cognition to content-based reasoning illustrates how the bounding box tokens act as spatial anchors that help the model pinpoint and organize the relevant information (such as *"Qty"*) to make the correct prediction.



(a) Attention map of the first layer. (b) Attention map of the last layer. (c) Attention score visualization of bounding box tokens.

Figure 5: Visualization of attention maps of LayTextLLM. Best viewed in color and with zoom. *"<unk>"* is the placeholder for the bounding box token.

The attention of LayTextLLM exhibits a distinct pattern compared to models like DocLLM, which uses block infilling to predict missing blocks from both preceding and succeeding context. In contrast, LayTextLLM adheres to an auto-regressive approach, focusing its attention solely on preceding information. Furthermore, interleaving bounding box and text embeddings creates strong attention connections between textual and spatial representations, as shown in Fig.5. In contrast, DocLLM integrates spatial information into the calculation of attention score which is implicitly. As shown in Tab. 2, LayTextLLM significantly outperforms DocLLM, again underscoring the advantage of interleaving bounding box and text embeddings. Also, we found that the spatial information can be decoded back into coordinates even without inputing visual information, as discussed in Appendix F.

Sequence Length Tab. 5 presents statistics on the average input sequence length across different datasets. Intriguingly, despite interleaving bounding box tokens, LayTextLLM consistently exhibits the shortest sequence length in three out of four datasets, even surpassing DocLLM, which is counterintuitive. We attribute this to the tokenizer mechanism. For example, using `tokenizer.encode()`, a single word from the OCR engine, like *"International"* is encoded into a single ID [4623]. Conversely, when the entire OCR output is processed as one sequence, such as *"... CPC,International,Inc..."*, the word *"International"* is split into two IDs [17579, 1288], corresponding to *"Intern"* and *"ational"* respectively. This type of case occurs frequently, we provide further discussion in Appendix C.

Dataset	LayTextLLM	DocLLM (Wang et al., 2023)	Coor-as-tokens (He et al., 2023)
DocVQA	**664.3**	827.5	4085.7
CORD	**137.9**	153.2	607.3
FUNSD	**701.9**	847.5	4183.4
SROIE	529.2	**505.1**	1357.7

Table 5: Average sequence length of each data for different methods using Llama2 tokenizer.

5 Conclusion

We propose LayTextLLM for VRDU tasks, interleaving spatial layouts and text to improve predictions through an innovative Spatial Layout Projector and the LNTP and SSFT training processes. Extensive experiments confirm the effectiveness of LayTextLLM.

REFERENCES

OpenAI:Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, FlorenciaLeoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, HyungWon Chung, Dave Cummings, and Jeremiah Currier. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, Dec 2023.

AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023b.

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

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, et al. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024.

Hao Feng, Qi Liu, Hao Liu, Wengang Zhou, Houqiang Li, and Can Huang. Docpedia: Unleashing the power of large multimodal model in the frequency domain for versatile document understanding. *arXiv preprint arXiv:2311.11810*, 2023a.

Hao Feng, Zijian Wang, Jingqun Tang, Jinghui Lu, Wengang Zhou, Houqiang Li, and Can Huang. Unidoc: A universal large multimodal model for simultaneous text detection, recognition, spotting and understanding. *arXiv preprint arXiv:2308.11592*, 2023b.

Liangcai Gao, Yilun Huang, Herve Dejean, Jean-Luc Meunier, Qinqin Yan, Yu Fang, Florian Kleber, and Eva Lang. Icdar 2019 competition on table detection and recognition (ctdar). In *International Conference on Document Analysis and Recognition*, 2019.

Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv:2304.15010*, 2023.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19485–19494, 2023.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10767–10775, Jul 2022. doi: 10.1609/aaai.v36i10.21322. URL http://dx.doi.org/10.1609/aaai.v36i10.21322.

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

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. Language is not all you need: Aligning perception with language models. *arXiv:2302.14045*, 2023.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 4083–4091, 2022.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1516–1520. IEEE, 2019.

Wonseok Hwang, Jinyeong Yim, Seung-Hyun Park, Sohee Yang, and Minjoon Seo. Spatial dependency parsing for semi-structured document information extraction. *Cornell University - arXiv,Cornell University - arXiv*, May 2020.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pp. 1–6. IEEE, 2019.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pp. 498–517, 2022.

Jianfeng Kuang, Wei Hua, Dingkang Liang, Mingkun Yang, Deqiang Jiang, Bo Ren, and Xiang Bai. Visual information extraction in the wild: practical dataset and end-to-end solution. In *International Conference on Document Analysis and Recognition*, pp. 36–53. Springer, 2023.

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 *International Conference on Machine Learning*, pp. 18893–18912. PMLR, 2023.

D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, Aug 2006. doi: 10.1145/1148170.1148307. URL `http://dx.doi.org/10.1145/1148170.1148307`.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, Jan 2020. doi: 10.18653/v1/2020.coling-main.82. URL `http://dx.doi.org/10.18653/v1/2020.coling-main.82`.

Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. *arXiv preprint arXiv:2311.06607*, 2023.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024b.

Xuejing Liu, Wei Tang, Xinzhe Ni, Jinghui Lu, Rui Zhao, Zechao Li, and Fei Tan. What large language models bring to text-rich vqa? *arXiv preprint arXiv:2311.07306*, 2023.

Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. Textmonkey: An ocr-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*, 2024c.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024a.

Jinghui Lu, Rui Zhao, Brian Mac Namee, and Fei Tan. Punifiedner: A prompting-based unified ner system for diverse datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 13327–13335, 2023.

Jinghui Lu, Ziwei Yang, Yanjie Wang, Xuejing Liu, and Can Huang. Padellm-ner: Parallel decoding in large language models for named entity recognition. *arXiv preprint arXiv:2402.04838*, 2024b.

Chuwei Luo, Yufan Shen, Zhaoqing Zhu, Qi Zheng, Zhi Yu, and Cong Yao. Layoutllm: Layout instruction tuning with large language models for document understanding. *CVPR 2024*, 2024.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022.

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.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv:2306.14824*, 2023.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. Visualmrc: Machine reading comprehension on document images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 13878–13888, 2021.

Jingqun Tang, Chunhui Lin, Zhen Zhao, Shu Wei, Binghong Wu, Qi Liu, Hao Feng, Yang Li, Siqi Wang, Lei Liao, et al. Textsquare: Scaling up text-centric visual instruction tuning. *arXiv preprint arXiv:2404.12803*, 2024a.

Jingqun Tang, Qi Liu, Yongjie Ye, Jinghui Lu, Shu Wei, Chunhui Lin, Wanqing Li, Mohamad Fitri Faiz Bin Mahmood, Hao Feng, Zhen Zhao, et al. Mtvqa: Benchmarking multilingual text-centric visual question answering. *arXiv preprint arXiv:2405.11985*, 2024b.

Zineng Tang, Zhenfeng Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Zhu C, Michael Zeng, Zhang Cha, and Mohit Bansal. Unifying vision, text, and layout for universal document processing. *Cornell University - arXiv,Cornell University - arXiv*, Dec 2022.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

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

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 *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Jan 2021. doi: 10.18653/v1/2021.acl-long.201. URL http://dx.doi.org/10.18653/v1/2021.acl-long.201.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Aug 2020. doi: 10.1145/3394486.3403172. URL http://dx.doi.org/10.1145/3394486.3403172.

Rui Yang, Boming Yang, Sixun Ouyang, Tianwei She, Aosong Feng, Yuang Jiang, Freddy Lecue, Jinghui Lu, and Irene Li. Leveraging large language models for concept graph recovery and question answering in nlp education. *arXiv preprint arXiv:2402.14293*, 2024.

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*, Sep 2023.

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:2307.02499*, 2023.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

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.

Dongsheng Zhu, Daniel Tang, Weidong Han, Jinghui Lu, Yukun Zhao, Guoliang Xing, Junfeng Wang, and Dawei Yin. VisLingInstruct: Elevating zero-shot learning in multi-modal language models with autonomous instruction optimization. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2122–2135, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.117.

APPENDIX

A QUALITATIVE EXAMPLES

Qualitative examples of document-oriented VQA (upper row) and KIE (bottom row) are shown in Fig. 6. The results indicate that LayTextLLM is highly effective in utilizing spatial layout information to make more accurate predictions for these challenging examples. For example, in the upper right figure, many numeric texts in the receipt act as noise for the baseline method. In contrast, LayTextLLM integrates layout information to accurately predict the total price, as demonstrated by the other examples, underscoring the utility of LayTextLLM.

B IMPLEMENTATION DETAIL

All training and inference procedures are conducted on eight NVIDIA A100 GPUs.

Training LayTextLLM is initialized with Llama2-7B-Base model, the pre-training, SFT, and other model hyper-parameters can be seen in Tab. 6. Please note that all variants of LayTextLLM, including those utilized in ablation studies, are trained in accordance with the SFT settings. All baseline results are sourced from their respective original papers, with the exception of the Llama2-7B series and the Llama2-7B$_{\text{coor}}$ series. These were re-implemented and can be referenced in (He et al., 2023; Luo et al., 2024).

	Backbone	**Plora rank**	**Batch size**	**Max length**	**Precision**	**Train params**	**Fix params**
Pretrain	Llama2-7B-base	256	128	2048	bf16	648 M	6.7 B
SFT	Llama2-7B-base	256	256	4096	bf16	7.4 B	0B
	Learning rate	**Weight decay**	**Scheduler**	**Adam betas**	**Adam epsilon**	**Warm up**	**Epoch**
Pretrain	1.0e-04	0.01	cosine	[0.9, 0.999]	1.0e-08	0.005	2
SFT	2.0e-05	0.01	cosine	[0.9, 0.999]	1.0e-08	0.005	2

Table 6: LayTextLLM trainng Hyper-parameters.

Inference For the document-oriented VQA test set, we use the original question-answer pairs as the prompt and ground truth, respectively. For Key Information Extraction (KIE) tasks, we reformat the key-value pairs into a question-answer format, as described in (Wang et al., 2023; Luo et al., 2024; Liu et al., 2024c). Additionally, for the FUNSD dataset, we focus our testing on the entity linking annotations as described in (Luo et al., 2024).

To eliminate the impact of randomness on evaluation, no sampling methods are employed during testing for any of the models. Instead, beam search with a beam size of 1 is used for generation across all models. Additionally, the maximum number of new tokens is set to 512, while the maximum number of input tokens is set to 4096.

C DISCUSSION OF INPUT SEQUENCE LENGTH

As mentioned in Section 4.6, it is intriguing that LayTextLLM has fewer input sequences than DocLLM, which is counterintuitive given that LayTextLLM interleaves bounding box tokens, typically resulting in longer sequence lengths. We attribute this to the Byte Pair Encoding (BPE) tokenizers (Sennrich et al., 2016) prevalently used in modern LLMs such as Llama2.

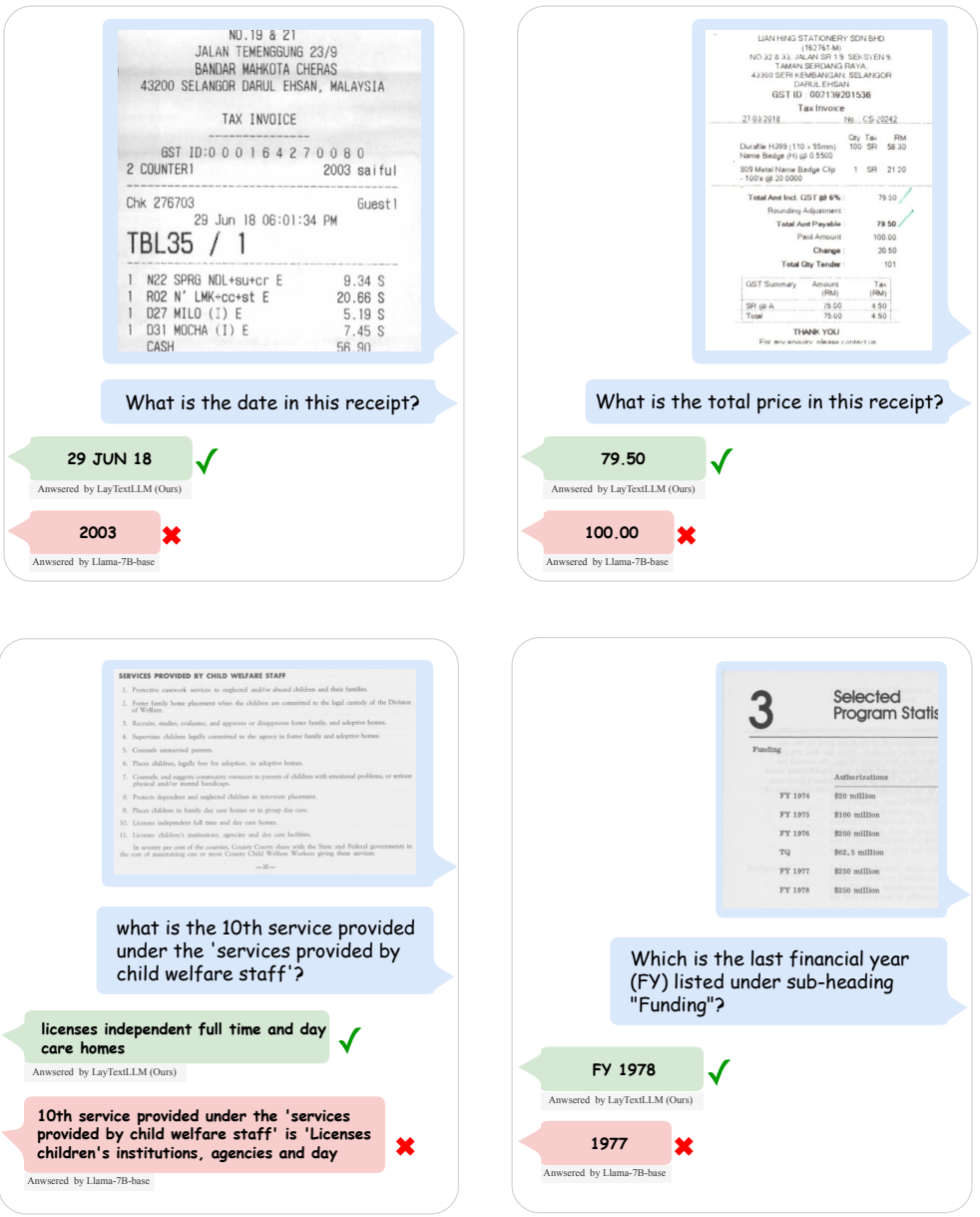


Figure 6: Qualitative comparison with the baseline method.

BPE operates by building a vocabulary of commonly occurring subwords (or token pieces) derived from the training data. Initially, it tokenizes the text at the character level and then progressively merges the most frequent adjacent pairs of characters or sequences. The objective is to strike a balance between minimizing vocabulary size and maximizing encoding efficiency.

Thus, when tokenizing a single word like *"International"* on its own, the tokenizer might identify it as a common sequence in the training data and encode it as a single token. This is especially likely if *"International"* frequently appears as a standalone word in the training contexts. However, when the word *"International"* is part of a larger sequence of words such as including in a long sequence of OCR-derived texts like *"...335 CPC,International,Inc..."*, the context changes. The tokenizer might split *"International"* into sub-tokens like *"Intern"* and *"ational"* because, in various contexts within

the training data, these subwords might appear more frequently in different combinations or are more useful for the model to understand variations in meaning or syntax.

When using LayTextLLM, we input word-level OCR results into the tokenizer, typically resulting in the former situation, where words are encoded as single tokens. Conversely, with DocLLM, the entire OCR output is processed as one large sequence, leading to the latter situation and a longer sequence length than in LayTextLLM. This difference underscores the utility of LayTextLLM in achieving both accuracy and inference efficiency due to its shorter sequence length.

D Shuffle Ratio Exploration

Tab. 7 presents the results of exploring training and testing shuffling ratios on the FUNSD and DocVQA dataset using two different models: Llama2-7B-base and LayTextLLM. The table shows the performance of these models at various shuffling ratios (100%, 50%, 20%, and 0%).

LayTextLLM consistently outperforms Llama2-7B-base across all levels of shuffling, which further underscores the significance of interleaving spatial layouts with text. Particularly at the 100% shuffle level, Llama2-7B-base demonstrates limited accuracy at only 20.3 (FUNSD) and 34.8 (DocVQA), while LayTextLLM maintains a relatively higher performance. It is also interesting to note that Llama2-7B-base generally improves as the shuffling percentage decreases, whereas LayTextLLM performs best when 20% of the examples with OCR-derived text are shuffled. This observation suggests that LayTextLLM effectively utilizes spatial layouts and is less dependent on the sequence of input tokens. Therefore, a certain proportion of shuffled examples can serve as adversarial examples to enhance the model's robustness, addressing situations such as errors in the text order from the OCR engine, which are caused by subtle differences in horizontal or vertical coordinates.

	FUNSD		DocVQA	
Ratio	Llama2-7B-base	LayTextLLM	Llama2-7B-base	LayTextLLM
100%	20.3	44.7	34.8	53.1
50%	49.1	62.1	63.1	72.8
20%	50.2	**65.4**	64.7	**73.4**
0%	**52.3**	65.1	**65.5**	73.0

Table 7: Shuffling ratio exploration in FUNSD and DocVQA dataset.

E Results of ChartQA

As shown in Fig. 7, the question-answer pairs in ChartQA (Masry et al., 2022) tend to involve the visual cues for reasoning. However, with only text and layout information as input, the proposed LayTextLLM inevitably have difficulties in reasoning visual-related information. Thus, on the ChartQA dataset, LayTextLLM can hardly achieve better performance than previous methods that include visual inputs. Although the visual information is not used in LayTextLLM, it can still exhibit better zero-shot ability than UniDoc (Feng et al., 2023b). After incorporating the training set of ChartQA, the performance of LayTextLLM can be boosted to 35.4%. Considering the importance of visual cues in ChartQA-like tasks, we will try to involve the visual information into LayTextLLM in future work.

F Decoding Bounding Box Coordinates

Although LayTextLLM is not explicitly designed to generate bounding boxes, we found that it can still do so. We believe this behavior arises from the chain-of-thought examples in the Layout-ware SFT dataset used by Luo et al. (2024). This ability is triggered by the prompt *Please think step-by-step."*, as shown in the example in Tab. 9. In response to the question, *"What is the content in the "NUMBER OF PAGES INCLUDING COVER SHEET:" field?"*, the model accurately outputs the bounding box for the region of interest, as demonstrated in Fig. 8. Notably, the model only uses bounding box tokens as input, without visual information. Even more intriguing is that the input OCR texts and

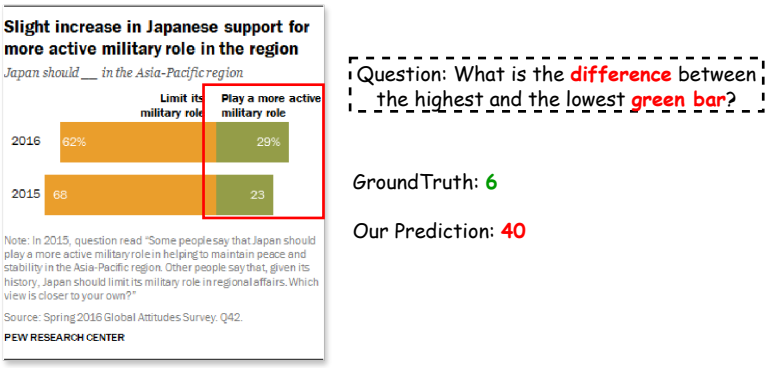


Figure 7: A failure case of LayTextLLM on ChartQA.

	ChartQA
OCR-free	
UniDoc (Feng et al., 2023b)	10.9
DocPedia (Feng et al., 2023a)	46.9*
Monkey (Li et al., 2023)	54.0*
InternVL (Chen et al., 2023b)	45.6*
InternLM-XComposer2 (Dong et al., 2024)	51.6*
TextMonkey (Liu et al., 2024c)	58.2*
TextMonkey$_+$ (Liu et al., 2024c)	**59.9***
Text + Coordinates	
LayTextLLM$_{zero}$ (Ours)	22.8
LayTextLLM$_{vqa}$ (Ours)	23.4*
LayTextLLM$_{all}$ (Ours)	35.4*

Table 8: Comparison with SOTA OCR-free MLLMs on ChartQA. * denotes the use of the dataset's training set.

bounding boxes are word-level and disordered, such as *...<unk> NUMBER<unk> OF<unk> INCLUDING<unk> COVER<unk> SHEET:<unk> PAGES<unk> 3<unk> SENDER<unk>...*". Despite this, the model accurately combines coordinates to produce a block-level bounding box.

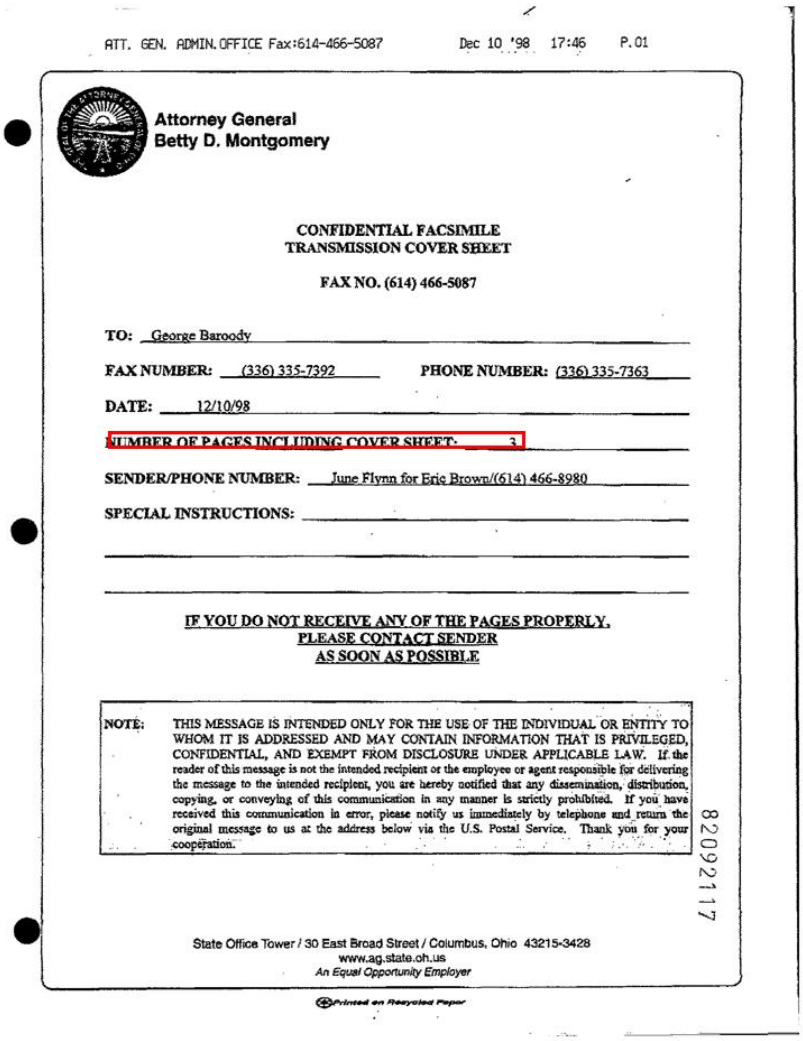
Input:

given document <document><unk> ATT.<unk> GEN.<unk> ADMIN.<unk> OFFICE<unk> Fax:<unk> 614<unk> -466<unk> -5087<unk> Dec<unk> 10<unk> ’98<unk> 17<unk> :46<unk> P.<unk> 01<unk><unk> Attorney<unk> General<unk> D.<unk> Betty<unk> Montgomery<unk> CONFIDENTIAL<unk> FACSIMILE<unk> TRANSMISSION<unk> COVER<unk> SHEET<unk> FAX<unk> NO.<unk> (614)<unk> 466-<unk> 5087<unk> TO:<unk> George<unk> Baroody<unk> FAX<unk> NUMBER:<unk> (336)<unk> 335-<unk> 7392<unk> PHONE<unk> NUMBER:<unk> (336)<unk> 335-<unk> 7363<unk> DATE:<unk> 12<unk> /10<unk> /98<unk> NUMBER<unk> OF<unk> INCLUDING<unk> COVER<unk> SHEET:<unk> PAGES<unk> 3<unk> SENDER<unk> /PHONE<unk> NUMBER:<unk> June<unk> Flynn<unk> for<unk> Eric<unk> Brown/<unk> (614)<unk> 466-<unk> 8980<unk> SPECIAL<unk> INSTRUCTIONS:<unk> NOT<unk> OF<unk> THE<unk> PAGES<unk> IF<unk> YOU<unk> DO<unk> RECEIVE<unk> ANY<unk> PROPERLY,<unk> PLEASE<unk> CONTACT<unk> SENDER<unk> AS<unk> SOON<unk> AS<unk> POSSIBLE<unk> NOTE:<unk> THIS<unk> MESSAGE<unk> IS<unk> INTENDED<unk> ONLY<unk> FOR<unk> THE<unk> USE<unk> OF<unk> THE<unk> INDIVIDUAL<unk> OR<unk> ENTITY<unk> TO<unk> WHOM<unk> IT<unk> IS<unk> ADDRESSED<unk> AND<unk> MAY<unk> CONTAIN<unk> INFORMATION<unk> THAT<unk> IS<unk> PRIVILEGED.<unk> CONFIDENTIAL,<unk> UNDER<unk> AND<unk> EXEMPT<unk> FROM<unk> DISCLOSURE<unk> APPLICABLE<unk> LAW.<unk> If<unk> the<unk> reader<unk> of<unk> this<unk> message<unk> is<unk> not<unk> the<unk> intended<unk> recipient<unk> of<unk> the<unk> employee<unk> or<unk> agent<unk> responsible<unk> for<unk> delivering<unk> the<unk> message<unk> to<unk> the<unk> intended<unk> recipient,<unk> you<unk> are<unk> hereby<unk> notified<unk> that<unk> any<unk> dissemination,<unk> distribution,<unk> copying,<unk> or<unk> conveying<unk> of<unk> this<unk> communication<unk> in<unk> any<unk> manner<unk> is<unk> strictly<unk> prohibited.<unk> If<unk> you<unk> have<unk> received<unk> this<unk> comunication<unk> in<unk> error,<unk> please<unk> notify<unk> us<unk> immediately<unk> by<unk> telephone<unk> and<unk> return<unk> the<unk> 82092117<unk> original<unk> message<unk> to<unk> us<unk> at<unk> the<unk> address<unk> below<unk> via<unk> the<unk> U.<unk> S.<unk> Postal<unk> Service.<unk> Thank<unk> you<unk> for<unk> your<unk> cooperation.<unk> State<unk> Tower<unk> /<unk> 30<unk> East<unk> Broad<unk> Street/<unk> Columbus,<unk> Ohio<unk> 43215<unk> -3428<unk> Office<unk> www<unk> ag<unk> state<unk> oh<unk> us<unk> An<unk> Equal<unk> Opportunity<unk> Employer<unk><unk> Primed<unk> an<unk><unk></document>, answer following question: **what is the content in the "NUMBER OF PAGES INCLUDING COVER SHEET:" field? P lease think step-by-step.**

Output:

answer: Step 1 (Question Analysis): The question is about "key-value extraction". And the question asks for the content in the "NUMBER OF PAGES INCLUDING COVER SHEET" field in the given document.
Step 2 (Relevant Area Concentration): The answer can be found in the area: **[142, 438, 635, 453]**.
Step 3 (Answer Formation): Based on the first two steps, regarding the question "what is the content in the "NUMBER OF PAGES INCLUDING COVER SHEET:" field?", the answer can be found in the sentence "NUMBER OF PAGES INCLUDING COVER SHEET: 3". The content in the "NUMBER OF PAGES INCLUDING COVER SHEET" field is "3".
Final answer: 3.

Table 9: FUNSD example with chain-of-thought and bounding box generation.



ATT. GEN. ADMIN. OFFICE Fax:614-466-5087 Dec 10 '98 17:46 P.01

Attorney General
Betty D. Montgomery

CONFIDENTIAL FACSIMILE
TRANSMISSION COVER SHEET

FAX NO. (614) 466-5087

TO: George Baroody

FAX NUMBER: (336) 335-7392 PHONE NUMBER: (336) 335-7363

DATE: 12/10/98

NUMBER OF PAGES INCLUDING COVER SHEET: 3

SENDER/PHONE NUMBER: June Flynn for Eric Brown/(614) 466-8980

SPECIAL INSTRUCTIONS:

IF YOU DO NOT RECEIVE ANY OF THE PAGES PROPERLY,
PLEASE CONTACT SENDER
AS SOON AS POSSIBLE

NOTE: THIS MESSAGE IS INTENDED ONLY FOR THE USE OF THE INDIVIDUAL OR ENTITY TO WHOM IT IS ADDRESSED AND MAY CONTAIN INFORMATION THAT IS PRIVILEGED, CONFIDENTIAL, AND EXEMPT FROM DISCLOSURE UNDER APPLICABLE LAW. If the reader of this message is not the intended recipient or the employee or agent responsible for delivering the message to the intended recipient, you are hereby notified that any dissemination, distribution, copying, or conveying of this communication in any manner is strictly prohibited. If you have received this communication in error, please notify us immediately by telephone and return the original message to us at the address below via the U.S. Postal Service. Thank you for your cooperation.

82092117

State Office Tower / 30 East Broad Street / Columbus, Ohio 43215-3428
www.ag.state.oh.us
An Equal Opportunity Employer

Printed on Recycled Paper

Figure 8: FUNSD example with bounding box generated by LayTextLLM.