A Bounding Box is Worth One Token: Interleaving Layout and Text in a Large Language Model for Document Understanding

Anonymous Author(s)
Affiliation
Address
email

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 27.0% increase on KIE tasks and 24.1% on VQA tasks compared to previous state-of-the-art document understanding MLLMs, as well as a 15.5% improvement over other SOTA OCR-based LLMs on KIE tasks.

1 Introduction

Recent research has increasingly focused on applying Large Language Models (LLMs) [1–17] 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 [6, 12, 18]. 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 [12].

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" [19]*. Following this premise, several studies [12, 20–23] 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 [20, 22, 23]. For example, Perot et al. [20] 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

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

the horizontal and vertical coordinates, respectively. Similarly, He et al. [23] 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 [20]. In contrast, DocLLM [19] 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) [23], 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 generalizability is limited. We believe that spatial cross attention and masked span tasks in DocLLM cannot fully utilize the autoregressive traits of LLMs.

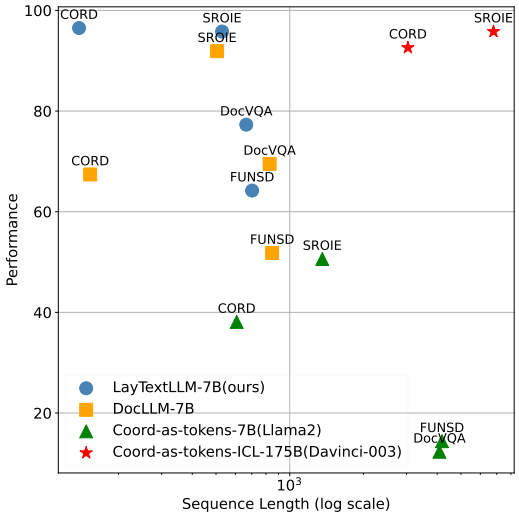


Figure 1: The performance against input sequence length of different datasets across various OCR-based methods where data is from Tab. 2 and 5.

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 [15, 24, 25], 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 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 [26–30] 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 [20], He et al. [23] 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 [19] designs a disentangled spatial attention mechanism to capture cross-alignment between text and layout modalities. Recently, LayoutLLM [21] utilizes the pre-trained layout-aware model [31], 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 [32] first presents an OCR-free method through mapping a text-rich document image into the desired answers. Pix2Struct [33] 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 [10–17], 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 [34] and UniDoc [10] are notable examples that expand upon the document-oriented VQA capabilities of LLaVA [35] 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 [9] 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 [18], Qwen-VL [6], and TextMonkey [12], 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 specific 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 [36, 37]. To enable the model to interleave spatial layouts with text, we introduce a novel Spatial Layout Projector. This projector converts OCR-derived 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.

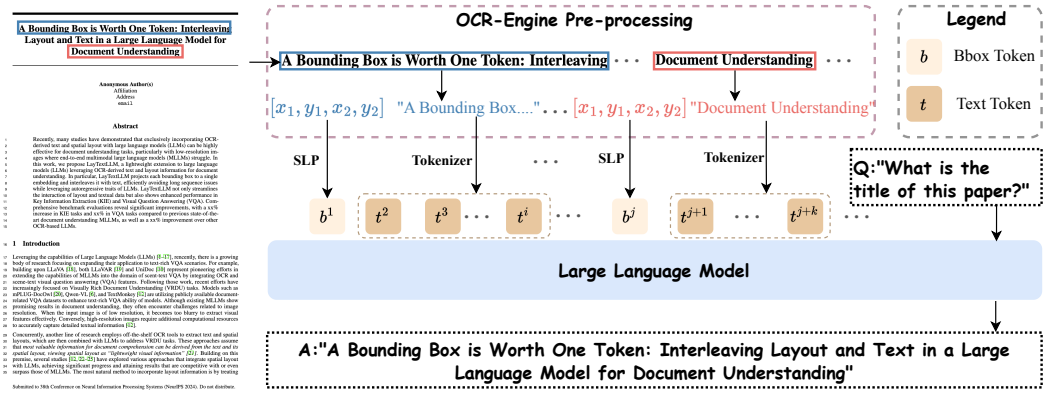


Figure 2: An overview of LayTextLLM incorporates interleaving bounding box tokens (b^i) with text tokens (t^i), where the superscripts represent the sequence positions of the tokens.

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. To be specifically, 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.

Compared to the coordinate-as-tokens scheme, the SLP represents each bounding box with a single token. This approach significantly reduces the number of input tokens and adheres to the practice of interleaving any modality with text, effectively integrating layout and textual information into a unified sequence. This allows the model to process both modalities simultaneously and coherently, fully leveraging the autoregressive traits of LLMs.

3.1.2 Layout Partial Low-Rank Adaptation

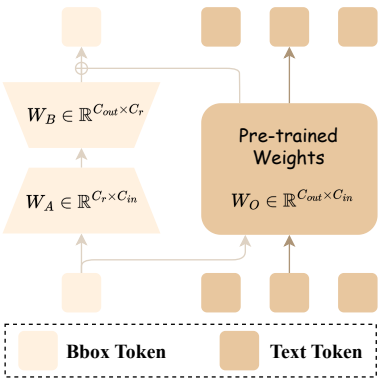
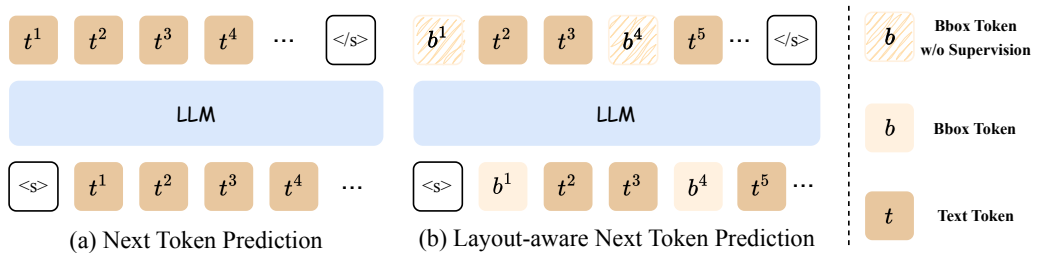


Figure 3: The illustration of P-LoRA, adapted from [15].

After using the SLP to generate bounding box tokens and a tokenizer to produce text tokens, these two modalities are then communicated using a Layout Partial Low-Rank Adaptation (P-LoRA) module in LLMs. P-LoRA, introduced in InternLM-XComposer2 [15], is originally used to adapt LLMs to 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, as shown in Fig. 3 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 $x =$



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

Figure 4: Comparison of Layout-aware Next Token Prediction and normal Next Token Prediction.

$[x_b, x_t]$ 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:

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

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 [1–7], we propose the Layout-aware Next Token Prediction (LNTP). Fig. 4 presents the contrast of the proposed Layout-aware Next Token Prediction and the conventional next token prediction task. The traditional next token prediction (Fig. 4(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. 4(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.

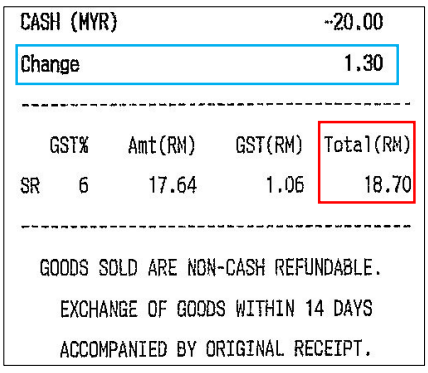


Figure 5: Receipt layout example.

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 [21]. Thus, LNTP facilitates the creation of large-scale, high-fidelity pre-training datasets at minimal cost.

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 [19, 21]. However, modern LLMs often exhibit a strong inductive bias toward the positions of input tokens, influenced by designs such as Rotary Position Embeddings (RoPE) [38]. 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 cons and pros.

Consider the example illustrated in Fig. 5, 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

Pre-training data In our 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** [39] and (2) **DocBank** [40]. 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 choose to utilize the entire DocBank dataset and only subsample 5 million pages from the IIT-CDIP collection 1.0.

SFT data For document-oriented VQA, we select **Document Dense Description (DDD)** and **Layout-aware SFT** data used in Luo et al. [21], which are two synthetic datasets generated by GPT-4. Besides, **DocVQA** [41], **InfoVQA** [42], **ChartQA** [43], **VisualMRC** [44] is included following [12]. For KIE task, we select **SROIE** [45], **CORD** [46], **FUNSD** [47], **POIE** [48] datasets following [12, 19, 21].

4.2 Implementation Detail

The LLM component of LayTextLLM is initialized from the Llama2-7B-base [36], 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. Other detailed setup can be found in Appendix B.

We have configured the model with three versions of LayTextLLM for a side-by-side comparison under different settings. Aligned with Luo et al. [21], the first version, **LayTextLLM**$_{zero}$, is trained exclusively with DDD and Layout-aware SFT data. Building upon this, and in alignment with the setting of Liu et al. [12], we introduce the DocVQA, InfoVQA, and ChartQA training sets to the dataset pool for our second version, termed **LayTextLLM**$_{vqa}$. Finally, we incorporate a comprehensive suite of KIE datasets—FUNSD, CORD, POIE, SROIE, and VisualMRC—as described by Wang et al. [19], creating our most extensive version, **LayTextLLM**$_{all}$. Note that all versions are based on the same pre-trained LayTextLLM weight.

	Document-Oriented VQA			KIE			
	DocVQA	InfoVQA	Avg	FUNSD	SROIE	POIE	Avg
Metric	*Accuracy %*						
OCR-free							
UniDoc [10]	7.7	14.7	11.2	1.0	2.9	5.1	3.0
DocPedia [9]	47.1*	15.2*	31.2	29.9	21.4	39.9	30.4
Monkey [49]	50.1*	25.8*	38.0	24.1	41.9	19.9	28.6
InternVL [50]	28.7*	23.6*	26.2	6.5	26.4	25.9	19.6
InternLM-XComposer2 [15]	39.7	28.6	34.2	15.3	34.2	49.3	32.9
TextMonkey [12]	64.3*	28.2*	46.3	32.3	47.0	27.9	35.7
TextMonkey$_+$ [12]	66.7*	28.6*	47.7	42.9	46.2	32.0	40.4
text + polys							
LayTextLLM$_{zero}$ (Ours)	71.8	33.8	52.8	**49.4**	**86.7**	66.1	**67.4**
LayTextLLM$_{vqa}$ (Ours)	**77.4***	**66.1***	**71.8**	48.9	74.6	**70.6**	64.7

Table 1: Comparison with SOTA OCR-free MLLMs. * indicates the training set used.

	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 + Polys							
Llama2-7B-base$_{coor}$ [23]	8.4	3.8	6.1	6.0	46.4	34.7	29.0
Llama2-7B-chat$_{coor}$ [23]	12.3	28.0	20.1	14.4	38.1	50.6	34.3
Davinci-003-175B$_{coor}$ [23]	-	-	-	-	92.6	95.8	-
DocLLM [19]	69.5*	264.1*	166.8	51.8*	67.4*	91.9*	70.3
LayTextLLM$_{zero}$ (Ours)	65.4	260.7	163.0	48.6	74.5	86.4	69.8
LayTextLLM$_{vqa}$ (Ours)	75.7*	260.2*	168.0	52.7	70.9	78.6	67.4
LayTextLLM$_{all}$ (Ours)	**77.3***	**295.9***	**186.6**	**64.2***	**96.5***	**95.8***	**85.8**

Table 2: Comparison with other OCR-based methods. * indicates the training set used.

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** [10], **DocPedia** [9], **Monkey** [49], **InternVL** [50], **InternLM-XComposer2** [15], **TextMonkey**, and **TextMonkey**$_+$ [12].

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. [23] for these models, resulting in two new variants: **Llama2-7B-base**$_{coor}$ and **Llama2-7B-chat**$_{coor}$. It's 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 [23], termed **Davinci-003-175B**$_{coor}$. One other recent SOTA OCR-based approach, **DocLLM** [19] is also considered in our analysis. Finally, **LayoutLLM** and **LayoutLLM**$_{CoT}$ [21], 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 [9, 10, 12]. 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 [19, 23], ANLS [19, 21, 51] and CIDEr [19, 52].

	Document-Oriented VQA			KIE			
	DocVQA	VisualMRC	Avg	FUNSD$^-$	CORD$^-$	SROIE$^-$	Avg
Metric	*ANLS %*						
Visual + Text + Polys							
LayoutLLM [21]	72.3	-	-	74.0	-	-	-
LayoutLLM$_{CoT}$ [21]	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 + Polys							
Llama2-7B-base$_{coor}$ [23]	8.4	6.7	7.5	4.3	33.0	47.2	28.1
Llama2-7B-chat$_{coor}$ [23]	12.3	12.2	12.2	11.9	6.4	39.4	19.2
LayTextLLM$_{zero}$ (Ours)	65.4	36.2	50.8	71.0	66.9	89.2	75.7
LayTextLLM$_{all}$ (Ours)	**77.3***	41.7*	59.5	**81.3***	**82.6***	**96.2***	**86.7**

Table 3: Comparison with LayoutLLM. $^-$ indicates that the cleaned test set used in Luo et al. [21].

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 [12], 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 71.8% and 33.8%, 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 VQA datasets. Specifically, its accuracies on DocVQA and InfoVQA increase to 77.4% and 66.1%, 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 86.7% and 66.1%, 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 correspinding experiments to align with OCR-based methods [19, 21]. 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 [19]. Specifically, when comparing with DocLLM, LayTextLLM$_{zero}$ demonstrates notably superior performance, with even its zero-shot capabilities being competitive with supervised 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 [53], 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. (3) The coordinate-as-tokens approach necessitates a considerably larger-sized LLM to comprehend the numerical tokens effectively.

In comparison to LayoutLLM [21], 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				
LNTP	SSFT	DocVQA	InfoVQA	VisualMRC	Avg	FUNSD	CORD	SROIE	POIE	Avg
	✓	72.3	35.7	24.4	44.2	50.6	91.6	92.8	58.6	73.4
✓		74.5	38.0	23.9	45.5	56.4	95.8	93.2	59.6	76.3
✓	✓	**78.8**	**42.7**	**35.1**	**52.2**	**62.9**	**95.9**	**95.2**	**61.7**	**78.9**

Table 4: Ablations on pre-training and SFT component of LayTextLLM (Accuracy).

Ablations To better assess the utility of Layout-aware Next Token Prediction and Shuffled-OCR Supervised Fine-tuning in LayTextLLM, an ablation study was performed (see Tab. 4). Details on the training setup for all variants are provided in Appendix B. It is evident that both LNTP and SSFT significantly enhance the utility of LayTextLLM. Specifically, disabling LNTP results in an 8% decrease in performance on VQA tasks and a 5.5% decrease on KIE tasks. Disabling SSFT leads to a decrease in average accuracy by 6.7% and 2.6% for VQA and KIE tasks, respectively.

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, more discussion in Appendix C.

Dataset	LayTextLLM	DocLLM [19]	Coor-as-tokens [23]
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 Limitation

Although LayTextLLM has shown significant capabilities in text-rich VQA and KIE tasks, this alone does not suffice for all real-world applications. There are some instances, particularly in chart analysis, where reasoning must be based solely on visual cues (*e.g.* size, color)—a challenge that remains unmet. Questions such as *"What is the difference between the highest and the lowest green bar?"* illustrate this gap. The ChartQA results, detailed in Appendix E, also underscore these limitations. Addressing these challenges highlights the urgent need for future enhancements that integrate visual cue within the capabilities of LayTextLLM.

6 Conclusion

We propose LayTextLLM for various VRDU tasks, in which spatial layouts and textual data are seamlessly interleaved to make more accurate prediction by introducing a innovative Spatial Layout Projector. Two tailored training tasks — Layout-aware Next Token Prediction and Shuffled-OCR Supervised Fine-tuning — are designed to improve the comprehension of document layouts. Extensive experiments confirm the effectiveness of LayTextLLM.

References

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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*, 2024.

[8] 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.

[9] 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*, 2023.

[10] 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*, 2023.

[11] 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.

[12] 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*, 2024.

[13] 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*, 2024.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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*, 2023.

[23] 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*, pages 19485–19494, 2023.

[24] 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.

[25] 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.

[26] 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.

[27] 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`.

[28] 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`.

[29] 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*, page 10767–10775, Jul 2022. doi: 10.1609/aaai.v36i10.21322. URL http://dx.doi.org/10.1609/aaai.v36i10.21322.

[30] 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.

[31] 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*, pages 4083–4091, 2022.

[32] 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*, pages 498–517, 2022.

[33] 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*, pages 18893–18912. PMLR, 2023.

[34] 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.

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

[36] 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.

[37] 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.

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

[39] 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.

[40] 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.

[41] 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*, pages 2200–2209, 2021.

[42] 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*, pages 1697–1706, 2022.

[43] 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, editors, *Findings of the Association*

for Computational Linguistics: ACL 2022, pages 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.

[44] 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, pages 13878–13888, 2021.

[45] 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)*, pages 1516–1520. IEEE, 2019.

[46] 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.

[47] 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, pages 1–6. IEEE, 2019.

[48] 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*, pages 36–53. Springer, 2023.

[49] 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.

[50] 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*, 2023.

[51] 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.

[52] 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*, pages 4566–4575, 2015.

[53] 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.

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

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.

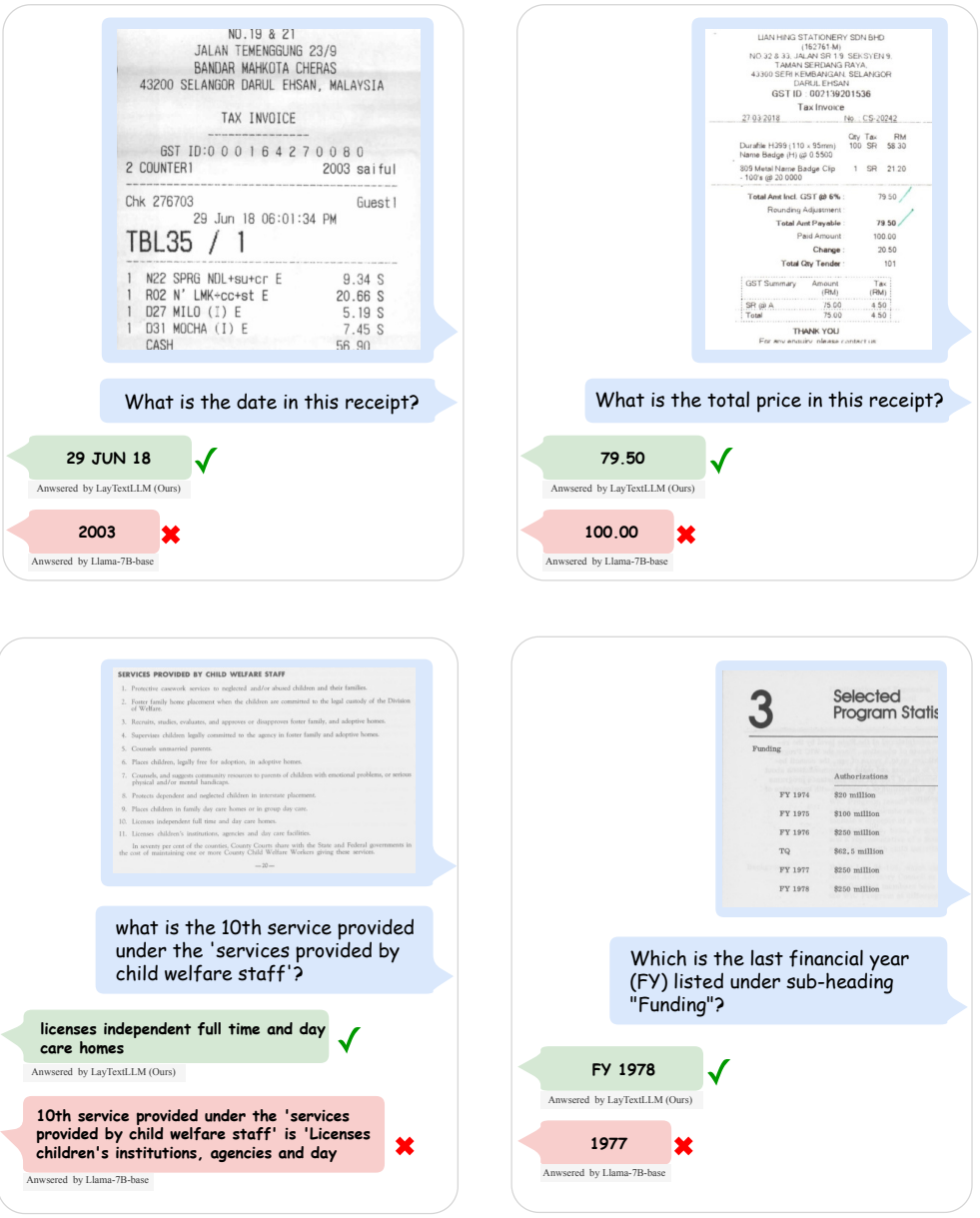


Figure 6: Qualitative comparison with the baseline method.

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 [21, 23].

	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 [12, 19, 21]. Additionally, for the FUNSD dataset, we focus our testing on the entity linking annotations as described in [21].

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 [54] prevalently used in modern LLMs such as Llama2.

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 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, 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	
Ratio	Llama2-7B-base	LayTextLLM
100%	20.3	44.7
50%	49.1	62.1
20%	50.2	**65.4**
0%	52.3	65.1

Table 7: Shuffling ratio exploration in FUNSD dataset.

E Results of ChartQA

As shown in Fig. 7, the question-answer pairs in ChartQA [43] 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 [10]. After incorporating the training set of ChartQA, the performance of LayTextLLM can be boosted to 35.7%. Considering the importance of visual cues in ChartQA-like tasks, we will try to involve the visual information into LayTextLLM in future work.

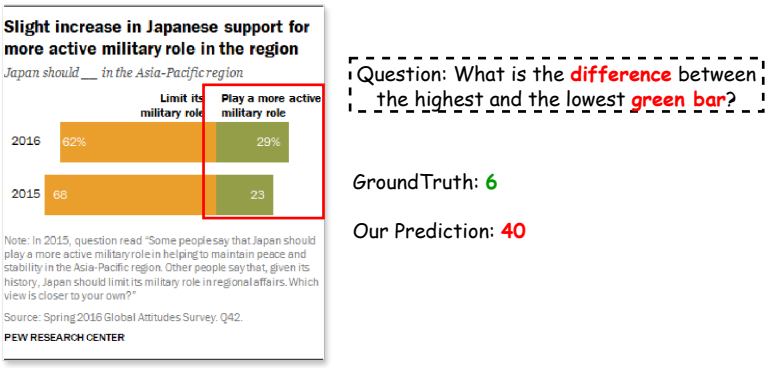


Figure 7: A failure case of LayTextLLM on CharQA.

	ChartQA
OCR-free	
UniDoc [10]	10.9
DocPedia [9]	46.9*
Monkey [49]	54.0*
InternVL [50]	45.6*
InternLM-XComposer2 [15]	51.6*
TextMonkey [12]	58.2*
TextMonkey$_+$ [12]	**59.9***
text + polys	
LayTextLLM$_{zero}$ (Ours)	21.4
LayTextLLM$_{vqa}$ (Ours)	29.8*
LayTextLLM$_{all}$ (Ours)	35.7*

Table 8: Comparison with SOTA OCR-free MLLMs on ChartQA. * indicates the training set used.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have detailed the contributions accurately in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: As discussed in Section. 5, we have listed some limitations of our work and shown corresponding failure cases in the supplementary material.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This work does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: In Section 4.1, 4.2 and Appendix B, we have described the details of implementing and training the proposed model to ensure the reproducibility of our work.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We would release our code after this paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We have detailed the experimental setting and implementation details in Section 4.1, 4.2 and Appendix B of the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Our deep learning model is designed for a complex task (requiring huge computing resources) where traditional error bars are less informative due to the high variability in model training and initialization. We ensured the robustness of our model by fix the random seed during inference. In addition, comparative analysis with baseline models demonstrated improvements in key performance areas, underscoring the practical effectiveness of our approach. We acknowledge the limitation of not using traditional statistical tests and suggest that future work could explore statistical significance in more controlled settings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have reported the needed computer resources in Section B of the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research in this paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: All the datasets used in this paper are publicly available and they contain no unsafe images.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: For the used datasets and pre-trained models, we have cited their corresponding works.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.