DocLLM: A Layout-Aware Generative Language Model for Multimodal Document Understanding

Anonymous ACL submission

Abstract

Enterprise documents such as forms, receipts, reports, and other such records, often carry rich semantics at the intersection of textual and spatial modalities. The visual cues offered by their complex layouts play a crucial role in comprehending these documents effectively. In this paper, we present DocLLM, a lightweight extension to traditional large language models (LLMs) for reasoning over visual documents, taking into account both textual semantics and spatial layout. Our model differs from existing multimodal LLMs by avoiding expensive image encoders and focuses exclusively on bounding box information to incorporate the spatial layout structure. Specifically, the cross-alignment between text and spatial modalities is captured by decomposing the attention mechanism in classical transformers to a set of disentangled matrices. Furthermore, we devise a pre-training objective that learns to infill text segments. This approach allows us to address irregular layouts and heterogeneous content frequently encountered in visual documents. The pre-trained model is fine-tuned using a large-scale instruction dataset, covering four core document intelligence tasks. We demonstrate that our solution outperforms SotA LLMs on 14 out of 16 datasets across all tasks, and generalizes well to 4 out of 5 previously unseen datasets.

1 Introduction

Documents with rich layouts, including invoices, contracts, and forms, constitute a significant portion of enterprise corpora, and the automatic analysis of these documents offer considerable advantages (Kunduru, 2023). Although Document AI (DocAI) has made tremendous progress, there remains a significant performance gap in real-world applications due to the complex layouts, bespoke type-setting and template diversity exhibited by these visually rich documents. In particular, accuracy, reliability, contextual understanding and generalization to previously unseen domains continues to be a challenge (Cui et al., 2021).

Conventional large language models (LLMs) such as GPT-3.5 (Brown et al., 2020), Llama (Touvron et al., 2023) or Falcon (Penedo et al., 2023) primarily accept text-only inputs and assume that the documents exhibit simple layouts and uniform formatting. They are not suitable for document intelligence tasks, which are inherently multi-modal, requiring the understanding of both text content and visual layout cues. Numerous vision-language frameworks (Li et al., 2022; Huang et al., 2022) that can process documents as images and capture the interactions between textual and visual modalities do exist. However, these frameworks necessitate the use of complex vision backbone architectures (Dosovitskiy et al., 2021) to encode image information, and often make use of spatial information as an auxiliary contextual signal (Xu et al., 2021; Lee et al., 2022).

In this paper, we present DocLLM, a lightweight extension to standard LLMs that excels in several visually rich form understanding tasks. Unlike traditional LLMs, it models both spatial layouts and text semantics, and therefore is intrinsically multimodal. The spatial layout information is incorporated through bounding box coordinates of the text tokens obtained typically using optical character recognition (OCR), and does not rely on a complex vision encoder component. Consequently, our solution preserves the causal decoder architecture, introduces only a marginal increase in the model size, and has reduced processing times. We demonstrate that merely including the spatial layout structure is sufficient for various document intelligence tasks such as form understanding, table alignment and visual question answering.

Existing efforts to incorporate spatial layout information typically involve either concatenating spatial and textual embeddings (Tang et al., 2023) or summing the two (Xu et al., 2020). In contrast,

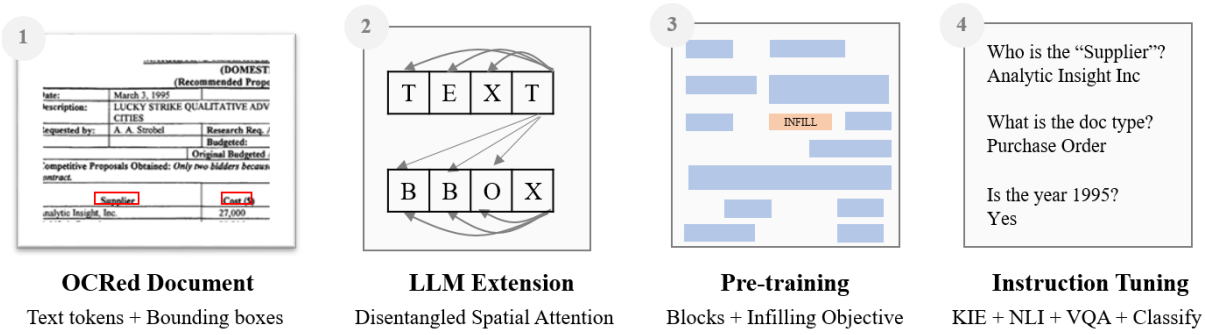


Figure 1: Key elements of DocLLM. (1) Input documents with text tokens and bounding boxes. (2) Extended attention mechanism captures cross-alignment between text semantics and spatial layouts. (3) Infilling text blocks is used as pre-training objective. (4) Task adaptation is performed on a newly collated dataset of instructions.

we treat the spatial information as a distinct modality and compute its inter-dependency with the text modality in a disentangled manner (Meng et al., 2021). Specifically, we extend the self-attention mechanism of transformers to include new attention scores that capture cross-modal relationships. There is often a correlation between the content, position and size of the fields in a form and hence representing their alignments at various abstraction levels across the transformer layers can enhance document understanding.

Visual documents often feature heterogeneous content, irregular layouts, and disjointed text segments. A classical next token prediction in self-supervised pre-training can be restrictive for these documents since the preceding tokens may not always be relevant due to the diverse arrangements of text. To tackle this issue, we propose two modifications to the pre-training objective: (a) adopting cohesive blocks of text that account for broader contexts, and (b) implementing an infilling approach by conditioning the prediction on both preceding and succeeding tokens. Due to these modifications, the model is better equipped to address misaligned text, contextual completions, intricate layouts, and mixed data types. Although text spans and infilling tasks have been studied before (Du et al., 2021), our solution is tailored for visual documents with an emphasis on semantically coherent blocks.

We tune DocLLM on instruction data curated from multiple datasets for several document intelligence tasks including Key Information Extraction (KIE), Natural Language Inference (NLI), Visual Question-Answering (VQA) and document classification (CLS). The modifications introduced by DocLLM enhances the performance of Llama2-7B model by 15-60% in four of five datasets unseen during training.

Our contributions include: (1) A lightweight extension to LLMs designed for understanding visual documents. (2) A disentangled spatial attention mechanism that captures cross-alignment between text and layout modalities. (3) An infilling pre-training objective tailored to address irregular layouts effectively. (4) A large instruction tuning dataset (with OCR data) specially curated towards visual document intelligence tasks. (5) Comprehensive experiments and insights into the model behavior. Fig. 1 summarizes the framework.

2 Related Work

General Purpose Models. By treating a document as text content, many text based LLMs (OpenAI, 2023; Touvron et al., 2023; Anil et al., 2023) can be directly utilized for document intelligence tasks. Despite the remarkable capabilities provided by these LLMs, their lack of understanding of visual elements and layouts can be severely limiting in the DocAI context (Liu et al., 2023c). Although multimodal LLMs (Li et al., 2023; Zhu et al., 2023; Liu et al., 2023a; Wu et al., 2023; Ye et al., 2023c) that explicitly include image information can account for visual signals, they often struggle to recognize structures and patterns observed in enterprise documents since most are not trained specifically for visually rich document understanding (VRDU) tasks.

Document Understanding Models. Models such as UDOP (Tang et al., 2023) and LayoutLM (Xu et al., 2020) specifically cater towards document processing tasks. They can account for different modalities including text, image and layout information and are trained using large document corpora. However, these models require task-specific fine-tuning, may lack a flexible interface and cannot understand open-domain instructions. Recent

efforts like mPLUG-DocOwl (Ye et al., 2023a) and UReader (Ye et al., 2023b) build on LLMs and perform DocAI-focused instruction tuning. We differ from these by avoiding expensive visual encoders.

Model Architecture. Disentangled attention mechanisms, where different signals are represented by independent vectors, have been studied before (He et al., 2020). While we use a similar construct, our spatial position based encodings are more complex and applied in a multimodal context. Learning to infill autoregressive language models has been explored in Bavarian et al. (2022), Shen et al. (2023), and Du et al. (2021). Although we share their goal of adding fill-in-the-middle (FIM) capability, we differ in the mechanism by integrating FIM into the visual document contexts and avoiding extremely short segments.

3 DocLLM Framework

3.1 Architecture Overview

DocLLM is constructed upon the foundation of an auto-regressive transformer language model (Touvron et al., 2023; Penedo et al., 2023) following a causal decoder structure. It integrates lightweight visual information by utilizing the spatial positions and dimensions of text tokens obtained using OCR. Instead of simply augmenting the text with bounding box information via additive positional encoding (Xu et al., 2021), separate vectors are used to represent these two distinct modalities and the self-attention mechanism of the transformer architecture is extended to compute their interdependencies in a disentangled manner. Furthermore, the traditional left-to-right next token prediction during self-supervised training is replaced by a block infilling objective that better leverages contextual information. See Figure 2 for an overview.

3.2 Disentangled Spatial Attention

Let $\mathbf{x} = (x_1, ..., x_i, ..., x_T)$ be an input sequence of length T, where x_i is a text token. In classical transformers, using a learned embedding matrix based on the text vocabulary and a learned set of parameters for the token position in the sequence, the input tokens are first encoded into hidden vectors $\mathbf{H} \in \mathbb{R}^{T \times d}$. A self-attention head then computes the attention scores between tokens i and j as:

$$\mathbf{Q}^t = \mathbf{H}\mathbf{W}^{t,q}, \quad \mathbf{K}^t = \mathbf{H}\mathbf{W}^{t,k}, \quad \mathbf{A}^t_{i,j} = \mathbf{Q}^t_i \mathbf{K}^{t\intercal}_j \tag{1}$$

where $\mathbf{W}^q \in \mathbb{R}^{d \times d}$ and $\mathbf{W}^k \in \mathbb{R}^{d \times d}$ are projection matrices, and the superscript t indicates the text modality. The attention scores $\mathbf{A} \in \mathbb{R}^{T \times T}$ along with another projection matrix $\mathbf{W}^v$ are further used to compute the hidden vectors $\mathbf{H}'$, which are in turn used as inputs for a subsequent layer:

$$\mathbf{V}^t = \mathbf{H}\mathbf{W}^{t,v}, \quad \mathbf{H}' = softmax(\frac{\mathbf{A}^t}{\sqrt{d}})\mathbf{V}^t. \tag{2}$$

In DocLLM, the input is represented as $\mathbf{x} = \{(x_i, b_i)\}_{i=1}^T$, where $b_i = $ (left, top, right, bottom) is the bounding box corresponding to x_i. To capture the new modality (i.e. spatial information), we encode the bounding boxes into hidden vectors represented by $\mathbf{S} \in \mathbb{R}^{T \times d}$. We then decompose the attention matrix computation into four different scores, namely *text-to-text*, *text-to-spatial*, *spatial-to-text* and *spatial-to-spatial*. Formally, the new attention mechanism is calculated as:

$$\begin{aligned} \mathbf{Q}^s = \mathbf{S}\mathbf{W}^{s,q}, \qquad \mathbf{K}^s = \mathbf{S}\mathbf{W}^{s,k}, \\ \mathbf{A}_{i,j} = \mathbf{Q}^t_i \mathbf{K}^{t\intercal}_j + \lambda_{t,s}\mathbf{Q}^t_i \mathbf{K}^{s\intercal}_j \\ + \lambda_{s,t}\mathbf{Q}^s_i \mathbf{K}^{t\intercal}_j + \lambda_{s,s}\mathbf{Q}^s_i \mathbf{K}^{s\intercal}_j, \end{aligned} \tag{3}$$

where $\mathbf{W}^{s,q} \in \mathbb{R}^{d \times d}$ and $\mathbf{W}^{s,k} \in \mathbb{R}^{d \times d}$ are newly introduced projection matrices corresponding to the spatial modality, and λs are hyperparameters that control the relative importance of each score. The input hidden vectors for the next layer $\mathbf{H}'$ are computed exactly as before. However, in contrast to equation (2), the newly calculated hidden vectors rely not only on the text semantics but also on the layout information of the text tokens.

It is important to mention that the hidden vectors $\mathbf{S}$ are reused across different layers, while each layer retains the flexibility to employ different projection matrices. We also note that the number of extra parameters required to encode the bounding box information is significantly lower compared to the overhead introduced by image based models (Li et al., 2022). By simply adding $\mathbf{S}$ to $\mathbf{H}$ similar to Xu et al. (2020), we could have avoided using $\mathbf{W}^s$ matrices altogether and further reduced the number of parameters. However, it would have irreversibly coupled the layout information with the text semantics. In contrast, our disentangled representation of these modalities in the attention scores enables selective focus when appropriate (He et al., 2020), thereby providing an optimal balance between model size and effectiveness.

3.3 Pretraining

DocLLM is first pre-trained in a self-supervised fashion on a large number of unlabeled documents.

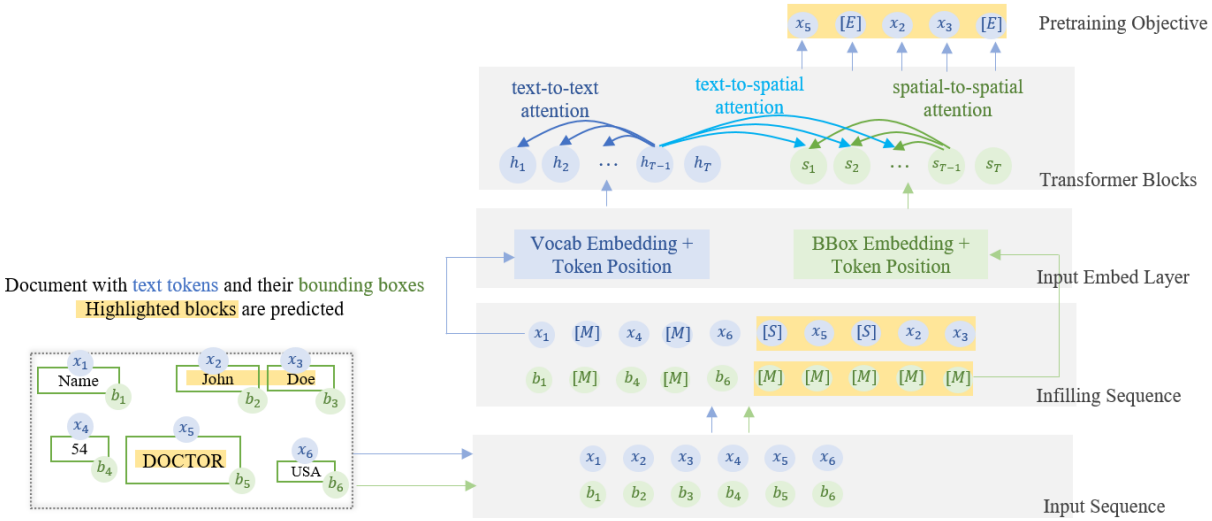


Figure 2: DocLLM model architecture with disentangled spatial attention and infilling objective. *left*: Input document with text tokens x_i and bounding boxes b_i. Some text blocks are randomly masked (two blocks here) and the model predicts the tokens in these text blocks autoregressively. *right*: The infilling sequence is created by replacing the sampled blocks with [M] and prepending them with [S]. The attention mechanism is extended to account for cross-attention between text and spatial modalities.

Visual documents are often sparse and irregular, featuring isolated and disconnected text fragments. It is preferable to consider coarse segments of related tokens during pre-training rather than focusing on individual tokens. Hence we use the broader context provided by multiple tokens, referred as blocks[1], for better comprehension. Most OCR engines can provide block level information, which makes it feasible to identify coherent text blocks such as a heading or an address[2].

Learning to infill text, where the prediction is conditioned on both prefix and suffix tokens rather than only preceding tokens, can be beneficial for document understanding. The infilling objectives enable contextually relevant completions, provide robustness to OCR noise or misaligned tokens, and can better handle relationships between various document fields. Hence we modify the standard pre-training objective to predict blocks of text given preceding and following text blocks. Inspired by (Du et al., 2021), we follow an autoregressive block infilling objective, where text blocks are randomly masked, and the masked blocks are shuffled and reconstructed in a sequential left-to-right fashion.

Formally, let $\mathbf{c} = \{c_1, ..., c_K\}$ be a set of text blocks that partitions an input sequence $\mathbf{x}$ into non-overlapping contiguous tokens such that $c_1 \cup ... \cup c_K = \mathbf{x}$ and $c_k \cap c_{k'} = \emptyset$. Let $\mathbf{z} = \{z_m\}_{m=1}^{M}$ be $M \ll K$ different text blocks randomly sampled from $\mathbf{c}$, where each block $z_m = (z_{m,1}, ..., z_{m,N_m})$ contains a consecutive series of tokens. Further, let $\tilde{\mathbf{x}}$ be a corrupted version of $\mathbf{x}$ where the contiguous tokens corresponding to a sampled text block are replaced with a special mask token [M]. To facilitate the identification of the block to be filled during text generation, each input block is augmented with a special start token [S] while the output block includes an end token [E]. For instance, a block with tokens (x_4, x_5) becomes [M] in $\tilde{\mathbf{x}}$, $([S], x_4, x_5)$ when conditioned upon, and is expected to generate $(x_4, x_5, [E])$ as output autoregressively[3]. Let θ denote all the parameters of the transformer model, including the projection matrices discussed above. The following cross-entropy loss is then minimized for the infilling objective

$$\mathcal{L}_{\text{IF}}(\theta) = -\sum_{m=1}^{M}\sum_{j=1}^{N_m} \log p_\theta(z_{m,j}|\tilde{\mathbf{x}}, \mathbf{z}_{<m}, \mathbf{z}_{m,<j}). \quad (4)$$

3.4 Instruction Tuning

Following recent work in the field of VRDU (Tang et al., 2023; Ye et al., 2023a,b) and prior work in NLP (Wei et al., 2022; Chung et al., 2022), we

[1] In Figure 2, "Name", "John Doe" , and "Doctor" are all examples of blocks

[2] In order to avoid any leakage of useful information, the block information is only used during pre-training, and the model is unaware of the number of tokens in a masked block.

[3] See Figure 2 for an illustration of these configurations.

Table 1: Prompt templates used for instruction-tuning (spatial tokens not included).

Task	Template type	Prompt template	Expected response
VQA	Extraction	`{document} {question}`	`answer` annotation
NLI	MCQ	`{document} "{statement}"`, Yes or No?	`answer` annotation
KIE	Extraction	`{document}` What is the value for the `"{key}"`?	Associated `value` annotation
	MCQ	`{document}` What is `"{value}"` in the document? Possible choices: `{keys}`. *(where* `keys` *is a subset of all the key names in the dataset in random order)*	Associated `key` annotation
	Internal classification	`{document}` What is `"{value}"` in the document?	Associated `key` annotation
CLS	MCQ	`{document}` What type of document is this? Possible choices: `{classes}`. *(where* `classes` *is a subset of all the classes in the dataset in random order)*	`class` annotation
	Internal classification	`{document}` What type of document is this?	`class` annotation

instruction-tune DocLLM on a variety of instructions curated from multiple DocAI datasets using templates. We employ a total of 16 datasets with their corresponding OCRs, spanning four DocAI tasks.

The diversity of supervised fine tuning (SFT) instructions is critical in helping zero-shot generalization (Wei et al., 2022; Chung et al., 2022; Ouyang et al., 2022). Thus, we diversify templates per task when possible, with each template asking a different question, and in some cases, expecting different types of answers. We re-use the templates introduced in Ye et al. (2023a,b) when applicable.

We create the templates following what we believe end users would generally ask about documents (see Table 1). For KIE and CLS, we hypothesize that (1) the extraction instructions can teach DocLLM to correlate names of keys in the prompts with document fields so as to retrieve values, (2) the internal classification instructions can help the model understand what intrinsically characterizes each key or document type, and (3) the multiple choice question (MCQ) instructions can teach the model to leverage its comprehension of key names included as choices in the prompt (resp. document type names) to classify extracted values (resp. entire documents). The templates are as follows[4]:

Visual Question Answering. A single template. Prompt Example: *What is the deadline for scientific abstract submission for ACOG - 51st annual clinical meeting?*

Natural Language Inference. A single template. Prompt Example: *"The UN commission on Korea include 2 Australians.", Yes or No?*

Key Information Extraction. Three templates corresponding to extraction, internal classification, and MCQ instructions. Example prompt for extraction: *What is the value for the "charity number"?*

Document Classification. Two templates corresponding to internal classification and MCQ instructions. Example prompt for MCQ: *What type of document is this? Possible answers: [budget, form, file folder, questionnaire].*

See Appendix A.2 for further details.

4 Experiments

4.1 Datasets

Pre-training. We gather data for pre-training from two primary sources: (1) IIT-CDIP Test Collection 1.0 (Lewis et al., 2006) and (2) DocBank (Li et al., 2020). IIT-CDIP Test Collection 1.0 encompasses a vast repository of over 5 million documents, comprising more than 16 million document pages. This dataset is derived from documents related to legal proceedings against the tobacco industry during the 1990s. DocBank consists of 500K documents, each featuring distinct layouts and a single page per document. We obtain a collection of 16.7 million pages comprising a total of 3.8 billion tokens. See Table 6 in the Appendix for detailed statistics.

Instruction Tuning. To instruction-tune the model for the VQA task, we collect DocVQA (Mathew et al., 2021), WikiTableQuestions (WTQ) (Pasupat and Liang, 2015), VisualMRC (Tanaka et al., 2021), and DUDE (Landeghem et al., 2023). For NLI, we only include TabFact (Chen et al., 2020) in our instruction-tuning data mix, due to lack of additional DocAI NLI datasets available. For KIE, we gather Kleister Charity (KLC) (Stanislawek et al., 2021), CORD (Park et al., 2019), FUNSD (Jaume et al., 2019), DeepForm (Svetlichnaya, 2020), PWC (Kardas et al., 2020), SROIE (Huang et al., 2019), and VRDU ad-buy (Wang et al., 2023) (with random train-test splitting). Finally, we use RVL-CDIP (Harley et al., 2015) to

[4]Examples are derived from DocVQA (Mathew et al., 2021), TabFact (Chen et al., 2020), KLC (Stanislawek et al., 2021), RVL-CDIP (Harley et al., 2015).

build our CLS instruction-tuning data. We also downsample RVL-CDIP in the train split to avoid hindering the other datasets due to size. See Table 7 in the Appendix for detailed statistics.

To the above datasets, we add BizDocs, a collection of ∼1,600 business entity filings curated from state registration websites within the US. BizDocs is annotated for three tasks – VQA, KIE, and CLS – and we therefore include it in the respective instruction-tuning collections[5].

4.2 Evaluation Setup

Model Configuration. We train two variants of DocLLM: DocLLM-1B, which is based on the Falcon-1B architecture (Penedo et al., 2023), and DocLLM-7B, which is based on the Llama2-7B architecture (Touvron et al., 2023)[6]. The maximum sequence length is set to 1,024 for both these models during the entire training process. See Appendix B for a detailed discussion on the model configuration and training hyper-parameters.

Settings. We investigate two experimental settings:
Same Datasets, Different Splits (SDDS): Following previous work (Lee et al., 2023; Davis et al., 2022; Kim et al., 2022; Tang et al., 2023; Ye et al., 2023a,b), we first evaluate DocLLM on the unseen test split (or dev split when labeled test split is not publicly available) of each of the 16 datasets composing the instruction tuning data. The motivation behind this very typical setting is to check how DocLLM performs when tasks and domains supposedly stay the same from train to test.
Same Tasks, Different Datasets (STDD): Following (Wei et al., 2022; Chung et al., 2022; Dai et al., 2023; Zhang et al., 2023a), we also evaluate DocLLM on held-out datasets. More precisely, we instruction-tune the pretrained checkpoint of DocLLM on prompts from 11 of the 16 datasets considered in SDDS, then evaluate DocLLM on the test split of the remaining five datasets. The rationale behind this evaluation setting is to assess the performance of DocLLM when tasks are unchanged but domains and layouts differ from train to test. We believe examining this setting in the DocAI field is relevant because industry use cases usually encountered in practice revolve around VQA, KIE, and CLS, while document characteristics tend to change more often in production. We specifically isolate DocVQA, KLC, and BizDocs for STDD evaluation in order to (1) exclude at least one dataset per task from SFT when possible, (2) leave enough datapoints per task in the training split of the instruction-tuning data, (3) avoid data leakage, and (4) benchmark models on popular yet challenging datasets when possible. Due to the high cost of instruction-tuning, we were not able to run experiments with other held-out datasets.

Baselines. In SDDS and STDD, we benchmark DocLLM against comparably-sized SotA LLMs using ZS prompts that contain the text extracted from each document using an OCR engine (excluding the spatial information) (Touvron et al., 2023; Ouyang et al., 2022). In SDDS, we also report numbers from recent DocAI LLMs evaluated in a similar setting (Ye et al., 2023a,b). As motivated in Section 2, we do not consider DocAI models that require task-specific fine-tuning such as LayoutLMv3 (Huang et al., 2022) or Pix2Struct (Lee et al., 2023), and/or dataset-specific prompts such as UDOP (Tang et al., 2023). We instead focus on LLMs with out-of-the-box instruction following capability[7].

Metrics. Following previous work (Borchmann et al., 2021; Lee et al., 2023; Ye et al., 2023b,a), we evaluate all VQA datasets using Average Normalized Levenshtein Similarity (ANLS) (Biten et al., 2019), with the exception of VisualMRC, for which we use CIDEr[8] (Vedantam et al., 2015) and WTQ, for which we use accuracy. Performance on all CLS and NLI datasets is measured using accuracy. We evaluate all KIE datasets with the F1 score.

4.3 Results

SDDS Setting. Table 2 shows that DocLLM-7B excels in 12 out of 16 datasets, inclusively compared to ZS results of GPT4 and Llama2, and SDDS results of mPLUG-DocOwl and UReader. Among equivalent models (excluding GPT4), our model outperforms in 14 out of 16 datasets. Specifically, DocLLM demonstrates superior performance in layout-intensive tasks such as KIE and CLS. In VQA and NLI, its performance surpasses that of most multimodal language models, although it underperforms compared to GPT4. GPT4 outper-

[5]The BizDocs dataset will be released upon acceptance.
[6]Since LLaMA2 does not come with pre-trained weights at 1B parameters, we use the Falcon-1B architecture for the smaller version of DocLLM.
[7]Refer to Appendix C.4 for a comparison against SotA models regardless of architecture.
[8]This is done to remain consistent with the results reported by other baselines.

Table 2: Performance comparison in the SDDS setting against other multimodal and non-multimodal LLMs; non-multimodal LLMs are Zero-Shot (ZS) prompted while multimodal LLMs are instruction-tuned on the train split of the datasets considered. '*' indicates datasets for which a designated test set was not publicly available.

	Dataset	GPT4+OCR – (T) ZS	Llama2+OCR 7B (T) ZS	mPLUG-DocOwl 7B (T+V) SDDS	UReader 7B (T+V) SDDS	DocLLM-1B 1B (T+L) SDDS	DocLLM-7B 7B (T+L) SDDS
VQA	DocVQA	**82.8**	47.4	62.2	65.4	61.4	69.5
	WTQ *(Accuracy)*	**65.4**	25.0	26.9	29.4	21.9	27.1
	VisualMRC *(CIDEr)*	255.1	115.5	188.8	221.7	245.0	**264.1**
	DUDE*	**54.6**	38.1	-	-	42.6	47.2
	BizDocs	76.4	48.8	-	-	84.5	**86.7**
NLI	TabFact	**77.1**	48.2	60.2	67.6	58.0	66.4
KIE	KLC	45.9	27.8	30.3	32.8	58.9	**60.3**
	CORD	58.3	13.8	-	-	66.9	**67.4**
	FUNSD	37.0	17.8	-	-	48.2	**51.8**
	DeepForm	42.1	20.5	42.6	49.5	71.3	**75.7**
	PWC	18.3	6.8	-	-	25.7	**29.06**
	SROIE	90.6	56.4	-	-	91.0	**91.9**
	VRDU a.-b.*	43.7	18.7	-	-	87.6	**88.8**
	BizDocs	66.1	10.8	-	-	95.4	**96.0**
CLS	RVL-CDIP	68.2	32.8	-	-	90.9	**91.8**
	BizDocs	84.9	40.9	-	-	98.3	**99.4**

Table 3: Performance comparison in the STDD setting on held-out VRDU datasets against non-multimodal LLMs.

Model	Size	Setting	DocVQA VQA	KLC KIE	BizDocs VQA	BizDocs KIE	BizDocs CLS
GPT4+OCR	–	ZS	**82.8**	45.9	**76.4**	66.1	**84.9**
Llama2+OCR	7B	ZS	47.4	27.8	48.4	10.8	40.9
DocLLM-1B	1B	STDD	53.5	40.1	65.5	63.0	20.8
DocLLM-7B	7B	STDD	63.4	**49.9**	73.3	**72.6**	31.1

Table 4: Ablation study on disentangled spatial attention. T and S stands for text and spatial modality respectively.

Cross-Modal Interactions	NTP Accuracy
T2T	35.43
T2S + T2T	38.08
S2T + T2T	38.05
S2S + T2T	**39.12**
T2S + S2S + T2T	39.06
S2T + S2S + T2T	39.07
T2S + S2T + S2S + T2T	39.02

Table 5: Ablation study on the block infilling objective.

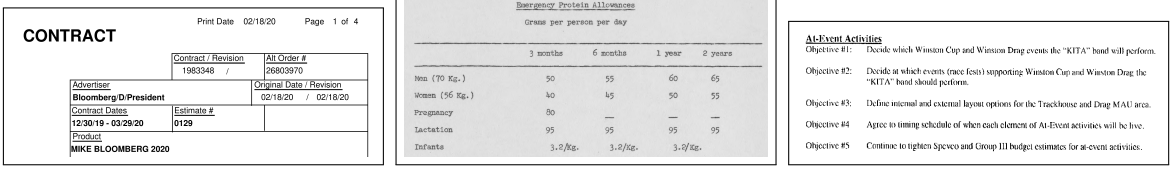
Pretraining Objective	NTP Accuracy
Causal Learning	32.6
Causal Learning + Spatial	36.2
Block Infilling + Spatial	**39.1**

forms DocLLM in VQA, possibly due to the higher complexity of reasoning and abstraction involved in VQA datasets compared to tasks like KIE or CLS[9]. DocLLM-1B demonstrates performance close to that of our larger model, suggesting that the smaller model can derive significant benefits from the architecture of DocLLM.

STDD Setting. Table 3 shows that our model demonstrates superior performance compared to Llama2 across four out of five datasets, and achieves the best score overall for two of them (KIE task again). DocLLM also outperforms mPLUG-DocOwl on DocVQA and both mPLUG-DocOwl and UReader on KLC, despite both baselines having been instruction-tuned on these datasets. However, it is important to note that classification accuracy is notably lower in our model. This discrepancy may stem from the fact that our model has been trained using only one CLS dataset, limiting its ability to generalize effectively to new datasets.

Qualitative Comparisons. Figure 3 shows qualitative examples, comparing the outputs of DocLLM-7B and GPT4. Figure 3a corresponds to a

[9]See Appendix C.2 for further details.



(a) Prompt: What is the value for the "advertiser"?
DocLLM: Bloomberg/D/President
GPT4+OCR: MIKE BLOOMBERG 2020

(b) Prompt: What is written under the heading 'emergency protein allowances'?
DocLLM: Grams per person per day
GPT4+OCR: Men (70 Kg.) 50 55 ...

(c) Prompt: How many objectives are listed under at-event activities?
DocLLM: 4
GPT4+OCR: 5

Figure 3: Qualitative examples of DocLLM-7B performance for KIE (Svetlichnaya, 2020) and VQA (Mathew et al., 2021) tasks. Correct answers are highlighted in blue and incorrect answers are highlighted in red.

KIE instruction, showing that DocLLM can provide correct answers when a question requires some knowledge of the semantic nuances of enterprise documents. DocLLM's spatial reasoning abilities are demonstrated in Figure 3b, where the model correctly locates the heading *'emergency protein allowances'* and identifies the text immediately underneath it. Figure 3c highlights a limitation, with the model failing at a counting task, at which GPT4 succeeds. See Appendix C.1 for more examples.

Ablation Analysis. We conduct ablation studies based on Next Token Prediction (NTP) accuracy to validate the main contributions of DocLLM. We observe that incorporating the spatial modality in the attention mechanism performs better over the classical text-only modality, thereby validating the utility of disentangled spatial attention (See Table 4). Furthermore, block infilling with spatial modality outperforms causal learning, highlighting the value of fill-in-the-middle objectives (See Table 5). Appendix D contains more details.

5 Discussion

Impact. DocLLM enables language models to go beyond plain text settings and offers immediate utility in visually rich document understanding tasks. By accommodating complex layout structures, DocLLM allows documents with rich layouts to be included in the pre-training corpus without requiring extensive preprocessing. The explicit modeling of spatial relationships enables perceiving the documents as inherently structured knowledge.

Flexibility. The support for multi-page documents, implemented through page breaks and document boundaries, enhances the model's ability to comprehend documents with diverse lengths. This overcomes the constraints of small multimodal models that can handle only a single page and multimodal LLMs mainly designed for images.

Limitations. The use of English-language datasets derived from limited enterprise domains (such as IIT-CDIP) may introduce inherent representational biases in VRDU models, including DocLLM. Also, DocLLM may be vulnerable to inaccurate bounding box information produced by an OCR engine. However, several modern off-the-shelf solutions can robustly extract text from documents, mitigating this issue. DocLLM's support for long-form documents is restricted by its context length. Increasing the model size and allowing unbounded context length during inference can address this limitation. Finally, DocLLM may not excel at complex reasoning tasks, especially those requiring a deep understanding of numerical concepts. See Appendix E for additional discussion.

6 Conclusions

We introduced DocLLM, a lightweight extension to traditional LLMs, tailored for generative reasoning over documents with rich layouts. DocLLM eschews expensive image encoders and instead utilizes bounding box information to capture the spatial layout structure of documents. This is achieved through a disentangled attention mechanism that models cross-alignment between text and spatial modalities. Notably, our model addresses the challenges posed by irregular layouts and heterogeneous content using a learning to infill pre-training objective. Tuning the model on a carefully curated instruction dataset provides a flexible interface for interactions. Our evaluation across various document intelligence tasks demonstrates that DocLLM surpasses equivalent models both for in-domain and out-of-domain tasks. In the future, we plan to infuse vision into DocLLM in a lightweight manner.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *CoRR*, abs/2308.12966.

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. 2022. Efficient training of language models to fill in the middle. *arXiv preprint arXiv:2207.14255*.

Ali Furkan Biten, Rubèn Tito, Andrés Mafla, Lluís Gómez, Marçal Rusiñol, Minesh Mathew, C. V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. 2019. ICDAR 2019 competition on scene text visual question answering. In *2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019*, pages 1563–1570. IEEE.

Lukasz Borchmann, Michal Pietruszka, Tomasz Stanislawek, Dawid Jurkiewicz, Michal Turski, Karolina Szyndler, and Filip Gralinski. 2021. DUE: end-to-end document understanding benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

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

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *CoRR*, abs/2305.06500.

Brian L. Davis, Bryan S. Morse, Brian L. Price, Chris Tensmeyer, Curtis Wigington, and Vlad I. Morariu. 2022. End-to-end document recognition and understanding with dessurt. In *Computer Vision - ECCV 2022 Workshops - Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IV*, volume 13804 of *Lecture Notes in Computer Science*, pages 280–296. Springer.

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

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*.

Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. In *13th International Conference on Document Analysis and Recognition, ICDAR 2015, Nancy, France, August 23-26, 2015*, pages 991–995. IEEE Computer Society.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. 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.

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

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. FUNSD: A dataset for form understanding in noisy scanned documents. In *2nd International Workshop on Open Services and Tools for Document Analysis, OST@ICDAR 2019, Sydney, Australia, September 22-25, 2019*, pages 1–6. IEEE.

Marcin Kardas, Piotr Czapla, Pontus Stenetorp, Sebastian Ruder, Sebastian Riedel, Ross Taylor, and Robert Stojnic. 2020. AxCell: Automatic extraction of results from machine learning papers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8580–8594, Online. Association for Computational Linguistics.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, page 498–517, Berlin, Heidelberg. Springer-Verlag.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Arjun Reddy Kunduru. 2023. From data entry to intelligence: Artificial intelligence's impact on financial system workflows. *International Journal on Orange Technologies*, 5(8):38–45.

Jordy Van Landeghem, Rubèn Tito, Lukasz Borchmann, Michal Pietruszka, Pawel Józiak, Rafal Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Anckaert, Ernest Valveny, Matthew B. Blaschko, Sien Moens, and Tomasz Stanislawek. 2023. Document understanding dataset and evaluation (DUDE). *CoRR*, abs/2305.08455.

Chen-Yu Lee, Chun-Liang Li, Timothy Dozat, Vincent Perot, Guolong Su, Nan Hua, Joshua Ainslie, Renshen Wang, Yasuhisa Fujii, and Tomas Pfister. 2022. FormNet: Structural encoding beyond sequential modeling in form document information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3735–3754, Dublin, Ireland. Association for Computational Linguistics.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 18893–18912. PMLR.

D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 2006. 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*, SIGIR '06, page 665–666, New York, NY, USA. Association for Computing Machinery.

Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2021. StructuralLM: Structural pre-training for form 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)*, pages 6309–6318. Association for Computational Linguistics.

Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. 2022. Dit: Self-supervised pre-training for document image transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3530–3539.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020. DocBank: A benchmark dataset for document layout analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 949–960, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Tianyang Liu, Fei Wang, and Muhao Chen. 2023b. Rethinking tabular data understanding with large language models. *CoRR*, abs/2312.16702.

Yuliang Liu, Zhang Li, Hongliang Li, Wenwen Yu, Mingxin Huang, Dezhi Peng, Mingyu Liu, Mingrui Chen, Chunyuan Li, Lianwen Jin, and Xiang Bai. 2023c. On the hidden mystery of OCR in large multimodal models. *CoRR*, abs/2305.07895.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq R. Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2263–2279. Association for Computational Linguistics.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. 2022. Infographicvqa. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022*, pages 2582–2591. IEEE.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. Docvqa: A dataset for VQA on document images. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, pages 2199–2208. IEEE.

Zihang Meng, Licheng Yu, Ning Zhang, Tamara L Berg, Babak Damavandi, Vikas Singh, and Amy Bearman. 2021. Connecting what to say with where to look by modeling human attention traces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12679–12688.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *CoRR*, abs/2307.06435.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: A consolidated receipt dataset for post-ocr parsing.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1470–1480. The Association for Computer Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Tianxiao Shen, Hao Peng, Ruoqi Shen, Yao Fu, Zaid Harchaoui, and Yejin Choi. 2023. Film: Fill-in language models for any-order generation. *arXiv preprint arXiv:2310.09930*.

Tomasz Stanislawek, Filip Gralinski, Anna Wróblewska, Dawid Lipinski, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemyslaw Biecek. 2021. Kleister: Key information extraction datasets involving long documents with complex layouts. In *16th International Conference on Document Analysis and Recognition, ICDAR 2021, Lausanne, Switzerland, September 5-10, 2021, Proceedings, Part I*, volume 12821 of *Lecture Notes in Computer Science*, pages 564–579. Springer.

Stacey Svetlichnaya. 2020. Deepform: Understand structured documents at scale.

Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. Visualmrc: Machine reading comprehension on document images. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13878–13888. AAAI Press.

Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. Unifying vision, text, and layout for universal document processing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 19254–19264. IEEE.

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

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks.

In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. 2023. VRDU: A benchmark for visually-rich document understanding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 5184–5193. ACM.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*.

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. 2021. 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)*, pages 2579–2591, Online. Association for Computational Linguistics.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. 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*, pages 1192–1200.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023a. mplug-docowl: Modularized multimodal large language model for document understanding. *CoRR*, abs/2307.02499.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, Qin Jin, Liang He, Xin Alex Lin, and Fei Huang. 2023b. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *CoRR*, abs/2310.05126.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023c. mplug-owl: Modularization empowers large language models with multimodality. *CoRR*, abs/2304.14178.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023d. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 174–184. ACM.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2024. Investigating the catastrophic forgetting in multimodal large language model fine-tuning. In *Conference on Parsimony and Learning*, pages 202–227. PMLR.

Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. 2023a. Llavar: Enhanced visual instruction tuning for text-rich image understanding. *CoRR*, abs/2306.17107.

Zhenrong Zhang, Jiefeng Ma, Jun Du, Licheng Wang, and Jianshu Zhang. 2023b. Multimodal pre-training based on graph attention network for document understanding. *IEEE Trans. Multim.*, 25:6743–6755.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A Dataset Details

A.1 Preprocessing

Of the datasets used in our study, IIT-CDIP and DocBank do not provide token-level OCR output. Therefore we process both datasets using the Tesseract-OCR engine[10]. For the remaining datasets, we used the OCR output provided by each publisher.

A.2 Instruction Tuning Templates

For the extraction template, we add a "None" answer if a key does not exist in the given document, following Ye et al. (2023a,b). As described in Section 3.4 and Table 1, to increase diversity in the training data, we derive internal classification and MCQ instructions in addition to extraction instructions from the original KIE annotations. However, to stay consistent with benchmarks from previous work (Ye et al., 2023a,b), we only keep the prompts derived from the extraction template in the test split of each KIE dataset. To avoid the cold start problem induced by potentially unseen types of documents in testing or production usage, we only keep the MCQ prompts for the test split of each CLS dataset. Note that when a prompt accepts more than one answer, we create multiple copies of the prompt with one acceptable answer assigned to each. We only perform this "flattening" operation in the training split of the dataset.

A.3 Dataset Statistics

See Table 6 for pretraining dataset details and Table 7 for instruction tuning dataset details.

Table 6: Pretraining dataset statistics.

Dataset	#Docs	#Pages	#Tokens
CDIP	5,092,636	16,293,353	3,637,551,478
DocBank	499,609	499,609	228,362,274
Total	5,592,245	16,792,962	3,865,913,752

Table 7: Instruction tuning dataset statistics.

Task	#Train prompts	#Test prompts
VQA	145,090	24,347
NLI	104,360	12,720
KIE	236,806	38,039
CLS	149,627	21,813
Total	635,883	96,919

B Training Details

DocLLM-1B is composed of 24 layers, each with 16 attention heads and a hidden size of 1,536. DocLLM-7B comprises 36 layers, 32 heads, and a hidden size of 4,096. Using pretrained weights as the backbone for the text modality, we extend the Falcon-1B and Llama2-7B models by adding the disentangled attention and block infilling objective as described in Section 3. We start directly from the pretrained weights of the backbone LLMs in order to continue their pretraining in a multimodal manner and avoid catastrophic forgetting of instruction following abilities (Luo et al., 2023; Zhai et al., 2024).

For DocLLM-1B, we use a pre-training learning rate of 2×10^{-4} with 1,000 warmup steps, employing a cosine scheduler, and Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.96$ and a weight decay of 0.1. For instruction tuning we use a learning rate of 1×10^{-4} with 500 warmup steps and a cosine scheduler, and the same parameters for weight decay and Adam optimizer as the pre-training phase. The Adam epsilon is set to 1×10^{-5}. We pretrain for one epoch, and instruction-tune for a total of 10 epochs.

For DocLLM-7B, pretraining involves a learning rate of 3×10^{-4} with 1,000 warmup steps and cosine scheduler, weight decay of 0.1, and Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.95$. Instruction tuning uses a learning rate of 1×10^{-4} with 500 warmup steps and a cosine scheduler, weight decay of 0.1, and Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.95$. Adam epsilon is set at 1×10^{-6}. We conduct one epoch of pretraining, followed by three epochs of instruction tuning, considering available computing resources.

The DocLLM-7B models are trained with 16-bit mixed precision on 8 24GB A10G GPUs using fully sharded data parallelism, implemented with the Accelerate library.[11] The DocLLM-1B model, on the other hand, is trained on a single 24GB A10G GPU.

Table 8 provides an overview of the model configuration and training hyper-parameters that were used.

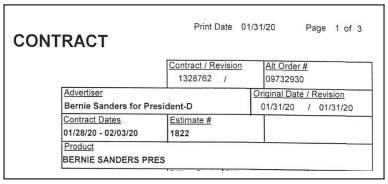
[10] https://github.com/tesseract-ocr/tesseract

[11] https://huggingface.co/docs/accelerate

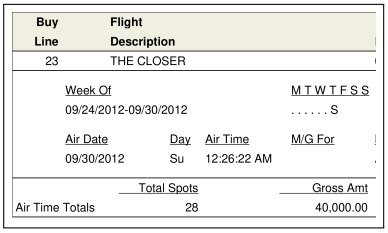
Table 8: Model configuration and training hyperparameters setting for DocLLM-1B and -7B.

	DocLLM-1B		DocLLM-7B	
Backbone	Falcon-1B (Penedo et al., 2023)		Llama2-7B (Touvron et al., 2023)	
#Parameters	1,524,963,328		7,853,019,136	
Layers	24		36	
Attention heads	16		32	
Hidden size	1,536		4,096	
Precision	bfloat16		bfloat16	
Batch size	2		5	
Max context length	1,024		1,024	
	Pretraining	**Instruction tuning**	**Pretraining**	**Instruction tuning**
Learning rate	2×10^{-4}	1×10^{-4}	3×10^{-4}	1×10^{-4}
Warmups	1,000	500	1,000	500
Scheduler type	cosine	cosine	cosine	cosine
Weight decay	0.1	0.1	0.1	0.1
Adam βs	(0.9, 0.96)	(0.9,0.96)	(0.9,0.95)	(0.9,0.95)
Adam epsilon	1×10^{-5}	1×10^{-5}	1×10^{-6}	1×10^{-6}

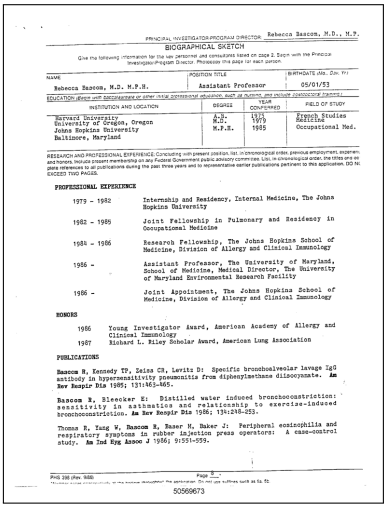
(a) Prompt: What is the doctor's id no.?
DocLLM: 162
GPT4: No information provided



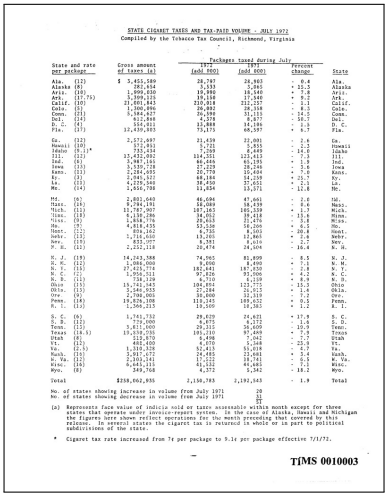
(b) Prompt: What is the value for the "contract num"?
DocLLM: 1328762
GPT4: 09732930



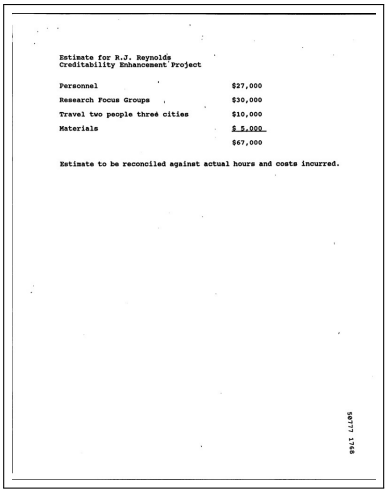
(c) Prompt: What is the value for the "gross amount"?
DocLLM: None
GPT4: 40,000.00



(d) DocLLM: resume
GPT4: form



(e) DocLLM: budget
GPT4: scientific report



(f) DocLLM: budget
GPT4: invoice

Figure 4: Qualitative examples of DocLLM-7B performance versus a SotA baseline (GPT4). Correct answers are highlighted in blue and incorrect answers are highlighted in red. (a): VQA example from the DocVQA dataset (Mathew et al., 2021). (b)-(c): KIE examples from the DeepForm dataset (Svetlichnaya, 2020). (d)-(f): CLS examples from the RVL-CDIP dataset (Harley et al., 2015). The prompt used here was: What type of document is this? Possible answers: [letter, memo, email, file folder, form, handwritten, invoice, advertisement, budget, news article, presentation, scientific publication, questionnaire, resume, scientific report, specification].

C Detailed Performance Analysis

C.1 Qualitative Examples

Figure 4 shows additional qualitative examples from the DocLLM-7B output, where 4a highlights a VQA example from the DocVQA dataset (Mathew et al., 2021), 4b and 4c display two KIE examples from the DeepForm dataset (Svetlichnaya, 2020), and the bottom row shows CLS examples from the RVL-CDIP dataset (Harley et al., 2015).

As Figures 4a and 4e show, DocLLM can provide correct answers when the question requires some knowledge of the semantic nuances of enterprise documents. As an example, in Figure 4e, GPT4 mislabels a tax report issued by a local tax council as a scientific report, possibly due to the numeric contents of the table, whereas DocLLM is able to associate the content and the corresponding issuing authority with a budget report. Figure 4b demonstrate DocLLM's spatial reasoning capability. The rightmost column of Figure 4 shows examples of failure by DocLLM. Each failure case demonstrates a limitation in the design and scope of the model. Figure 4c shows an example where DocLLM is unable to extract the gross amount. This error is due to the fact that the correct answer falls outside of the context window of the model, as it is located on the fourth page of a multi-page document. Lastly, Figure 4f shows an example for which the class predicted by DocLLM, i.e. "budget", is semantically viable, but is nevertheless not the correct class. In future studies, we plan to address some of the above mentioned limitations, and increase the context length of the model.

C.2 DocVQA Deep-Dive

We conduct an in-depth analysis of the performance of DocLLM-7B on the various question categories of DocVQA. As depicted in Table 9, DocLLM exhibits strong performance on "Form" and "Layout" questions, attaining scores of 82.2 and 72.4 respectively. These results underline the model's proficiency in understanding and processing structured document formats and layouts. Conversely, the "Image/Photo", "Figure/Diagram", and "Yes/No" questions have lower scores of 47.9, 41.4, and 43.9 respectively. The absence of integrated vision features might account for DocLLM's lower capacity in recognizing certain visual cues.

Table 9: DocLLM-7B scores for DocVQA categories.

Category	ANLS
Figure/Diagram	41.4
Form	82.2
Table/List	66.2
Layout	72.4
Free text	64.6
Image/Photo	47.8
Handwritten	62.8
Yes/No	43.9
Other	56.8

C.3 GPT4V Performance Comparison

Given the recent roll out of the GPT4V API[12] and the interest it has generated, we also benchmark DocLLM-7B against GPT4V on DocVQA and BizDocs (Table 10). We select these datasets in order to include both SDDS and STDD results in the comparison. Moreover, as BizDocs has not been made public yet, we can be certain that GPT4 and GPT4V have not been trained on it. Due to cost and daily API usage limitations, we were not able to cover additional datasets.

We first observe that GPT4V does not uniformly outperform GPT4+OCR on the datasets considered. Both models show close ZS performance in BizDocs CLS, but GPT4+OCR beats GPT4V in BizDocs VQA while GPT4V tops GPT4+OCR on BizDocs KIE and DocVQA. The additional vision component of GPT4V seems to help in general, especially for datasets such as DocVQA. However, as the characteristics of these model are undisclosed, analyzing their performance differences in depth is difficult. We do note that, despite its lack of visual and spatial features, GPT4+OCR fares well on VQA, KIE, and CLS tasks, and might be able to partially model the spatial relationships in documents based on the natural ordering of OCR tokens.

[12] https://openai.com/blog/new-models-and-developer-products-announced-at-devday

Table 10: DocLLM-7B performance comparison against GPT4+OCR and GPT4V. BizDocs KIE GPT4V results were obtained on a sample of 5K (cost & API limits).

Model	Setting	DocVQA	BizDocs		
		VQA	VQA	KIE	CLS
GPT4+OCR	ZS	82.8	76.4	66.1	84.9
GPT4V	ZS	**88.4**	67.9	70.0	86.0
DocLLM-7B	SDDS	69.5	**86.7**	**96.0**	**99.4**
DocLLM-7B	STDD	63.4	73.3	72.6	31.1

Its robustness to OCR token position permutations is however not guaranteed.

Next, we observe that DocLLM-7B also outperforms GPT4V in addition to GPT4+OCR on BizDocs SDDS. In the STDD evaluation setting, which is closer to out-of-distribution ZS inference, our model still exhibits competitive performance in VQA and KIE – although not consistently exceeding the scores of the likely larger GPT4 models. DocLLM's lack of vision encoder appears to be mostly detrimental on DocVQA, where it particularly struggles on "Image/Photo" and "Figure/Diagram" questions, as seen in Section C.2.

C.4 SotA Performance Comparison

In Table 11, we compare DocLLM-7B against the SotA on the datasets considered in this paper. Note that BizDocs is not included here as it has not been made public yet. Similarly, DUDE and VRDU ad-buy are not considered in this section, since we used validation and bespoke splits respectively to evaluate models on them (see the caption on Table 2). FUNSD and PWC are also excluded from this study, as the prompts we built for these datasets leveraged annotations differently than previous work: our FUNSD KIE questions are based on the annotated key-value links, and our PWC KIE questions are formulated using the annotated set of Machine Learning tasks covered by the dataset.

Table 11 offers a few notable takeaways. First, despite the recent progress in multi-modal document understanding, a foundation model that outranks others across a wide range of tasks and datasets does not currently exist. Most SotA models are single-task fine-tuned models that outperform others in one or a few datasets, as seen here with LayoutT5 (Tanaka et al., 2021), StructuralLM (Li et al., 2021), PASTA+DATER (Ye et al., 2023d), GPT-3.5+DP+PyAgent+MixSC (Liu et al., 2023b), and GraphDoc (Zhang et al., 2023b). The same observation applies to general NLP (Brown et al., 2020; Wang et al., 2022; Chowdhery et al., 2022; Naveed et al., 2023). While UDOP tops all models on three KIE datasets, it remains an expert model that requires dataset-specific prompts and per dataset fine-tuning (on top of its multitask supervised pretraining) in order to reach the performance reported. On table-based datasets such as WTQ and TabFact, SotA models rely on large, text-only LLMs to reason over data using SQL or Pandas – thus reducing their ability to generalize to non-tabular document data. The abstractive reasoning limitations of DocLLM-7B are more apparent on these table-based datasets, but our single model performs competititvely in KIE and CLS (even on KLC and Deepform, despite DocLLM's relatively short context-length).

Second, recent multimodal LLMs such as Qwen-VL-Max[13] and GPT4V[14] show impressive ZS performance in VQA. These generalist models report strong performance on DocVQA and other datasets like ChartQA (Masry et al., 2022) and InfographicVQA (Mathew et al., 2022) (which we do not consider in this paper) thanks to their additional vision encoder[15]. However, the lack of transparency about their size, exact architecture, training procedure, and training data makes it hard to draw any conclusions. On DocVQA, DocLLM-7B outperforms Qwen-VL-10B (Bai et al., 2023). Moreover, as these recent multimodal LLMs were designed to tackle a wide range of tasks (e.g., image captioning) and not just DocAI, their ZS performance on certain tasks considered here (document NLI, KIE, CLS) has not been investigated – making a thorough comparison with our model even more complex.

Finally, despite lower performance compared to the top-performing model in each category, DocLLM still shows superior performance to generalist LLMs of comparable size, as indicated in Table 2. The model also proves robust to out-of-distribution data in ZS, as demonstrated in Table 3.

D Ablation Studies

We conduct ablation studies to validate the three main contributions of DocLLM: (1) disentangled spatial features, (2) the block infilling pre-training objective, and (3) the masking strategy used for decoding. For all ablations, we use Next Token Prediction (NTP) out-of-sample accuracy to compare configurations at the pre-training stage. Due to resource restrictions, each experiment uses a subset of our pre-training corpus: we randomly sample 100,000 chunks and predict on 1,000 unseen documents. A chunk is a collection of documents wherein the total number of tokens across the collection is less than the maximum input context length. The hyperparameters are set consistently

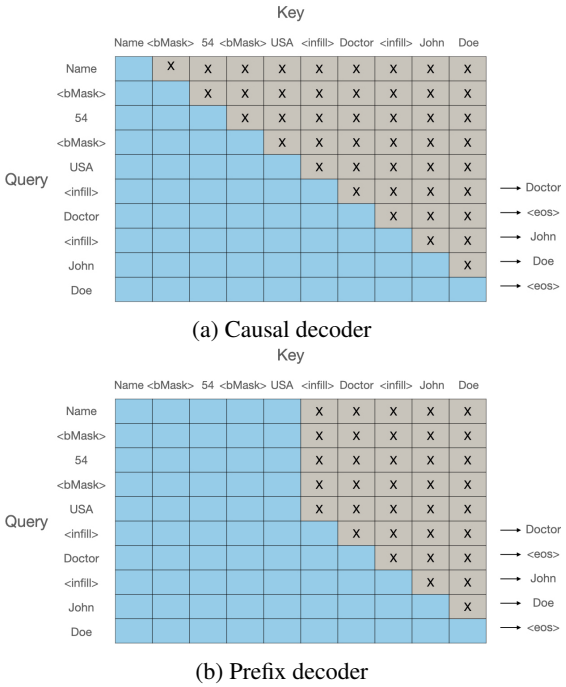
[13] https://qwenlm.github.io/blog/qwen-vl/

[14] https://openai.com/research/gpt-4

[15] In future studies, we hope to equip DocLLM with access to the vision modality too — albeit in a more efficient manner than is typically implemented.

Table 11: DocLLM-7B (SDDS) performance comparison against SotA models.

	Dataset	Model	Modality	SotA	DocLLM-7B
VQA	DocVQA	**Qwen-VL-Max** (qwenlm.github.io/blog/qwen-vl)	T+V	93.1	69.5
	WTQ *(Accuracy)*	**GPT-3.5+DP+PyAgent+MixSC** (Liu et al., 2023b)	T	73.6	27.1
	VisualMRC *(CIDEr)*	**LayoutT5** (Tanaka et al., 2021)	T+V+L	364.2	264.1
NLI	TabFact	**PASTA+DATER** (Ye et al., 2023d)	T	93.0	66.4
KIE	KLC	**UDOP** (Tang et al., 2023)	T+V+L	82.8	60.3
	CORD	**UDOP** (Tang et al., 2023)	T+V+L	97.6	67.4
	DeepForm	**UDOP** (Tang et al., 2023)	T+V+L	85.5	75.7
	SROIE	**GraphDoc** (Zhang et al., 2023b)	T+V+L	98.45	91.9
CLS	RVL-CDIP	**StructuralLM** (Li et al., 2021)	T+L	96.1	91.8



(a) Causal decoder

(b) Prefix decoder

Figure 5: A simplified illustration of attention masks for causal-decoder and prefix-decoder for block infilling.

following Table 8 across all ablation experiments.

Disentangled Spatial Attention. To measure the effect of disentangled spatial attention on cross-modal interactions, we train the models by setting the λ hyperparameter in Eq 4 to 0 or 1. Table 4 enumerates the attention combinations, and the results suggest that keeping only the spatial-to-spatial interaction (i.e. $\lambda_{s,s} = 1$) yields the highest NTP accuracy. The performance differences among other configurations, such as text-to-spatial and spatial-to-text, are subtle. Notably, the vanilla text-only self-attention mechanism yields the lowest NTP accuracy, underlining the importance of incorporating spatial features for understanding documents with rich layouts. For all experiments in Section 4, we therefore set $\lambda_{s,s} = 1$, $\lambda_{s,t} = 0$, and $\lambda_{t,s} = 0$. We opt for simplicity by choosing a hard mode over a soft one while acknowledging the potential advantage of flexibility for the latter.

Autoregressive Block Infilling. To evaluate the effectiveness of the proposed autoregressive block infilling objective especially comparing with the conventional left-to-right causal learning, we benchmark three configurations in our ablation study: (1) causal learning, (2) causal learning with spatial modality, and (3) block infilling with spatial modality. As highlighted in Table 5, autoregressive block infilling exhibits the best performance. Additionally, the performance gain of adding the spatial modality to the causal learning proves the advantage of the spatial modality.

Prefix Decoder and Causal Decoder. For document-conditioned generation, an intuitive choice is to employ a prefix decoder with prefix masking that utilizes bidirectional attention mechanism for the entire document, as illustrated in Figure 5b. We investigate this assumption through experiments where we compare a prefix decoder against the conventional causal decoder. Specifically, we conduct experiments on these two decoders for different settings outlined in the **Disen-**

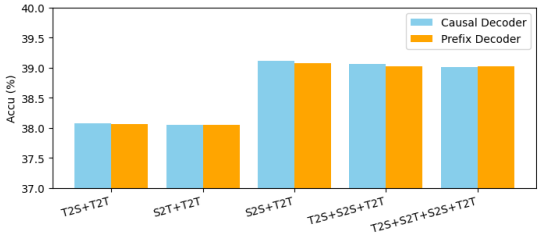


Figure 6: Performance comparison on NTP between causal decoder and prefix decoder.

tangled Spatial Attention ablation to study their resulting performance.

The results in Figure 6 show marginal differences between these two decoders across the five configurations, with the causal decoder having a slight edge over the prefix. The minor difference suggests that both masking methods are comparable in modeling documents. Thus the bidirectional attention enabled by the prefix decoder may not be crucial in this context, and we consequently elect to use a causal decoder for all experiments in section 4.

E Additional Discussion

The main concept for a cohesive block is to ensure meaningful infilling during the pretraining phase, preventing disconnected predictions. However, the choice of OCR engines to obtain such cohesive blocks remains an open area for exploration. Practical comparisons with various OCR engines and/or layout parsers are left as future work, as LayoutLMs underscore the importance of accurate OCR for improved VQA results. They leverage the Microsoft Azure API, demonstrating superior performance compared to TesseractOCR, as indicated in the DocVQA leaderboard[16]. Consequently, researchers are also encouraged to utilize more accurate OCR engines for potential enhancements, if such resources are available.

We have presented a collection of SDDS results alongside zero-shot outcomes. To mitigate prompt influence in the zero-shot results, a rigorous methodology was implemented. This involved the engagement of three independent prompt engineers, each undergoing five rounds of refinement for zero-shot settings, followed by a series of post-processing techniques to enhance result reliability. The best results are thus obtained from each of the three groups. We still acknowledge the potential for refinement and improvement.

We share some internal training experiences, acknowledging the absence of robust validation. First, we observe that a higher weight decay (e.g., 0.1 versus 0.01) generally improves performance in both pretraining and instruction tuning. During the instruction tuning phase, a higher initial learning rate, such as 1e-4 versus 5e-5, leads to enhanced performance. Overall, we've observed that the cosine scheduler tends to outperform linear or constant schedulers across various settings.

[16] https://rrc.cvc.uab.es/?ch=17&com=evaluation&task=1