DocLLM: A LAYOUT-AWARE LANGUAGE MODEL FOR MULTIMODAL DOCUMENT UNDERSTANDING

Dongsheng Wang*, **Natraj Raman***, **Mathieu Sibue***
Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, Xiaomo Liu
JPMorgan AI Research
{first.last}@jpmchase.com

ABSTRACT

Enterprise documents such as forms, invoices, receipts, reports, contracts, and other similar 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.

***K*eywords** DocAI · VRDU · LLM · GPT · Spatial Attention

1 Introduction

Documents with rich layouts, including invoices, receipts, contracts, orders, and forms, constitute a significant portion of enterprise corpora. The automatic interpretation and analysis of these documents offer considerable advantages [1], which has spurred the development of AI-driven solutions. These visually rich documents feature complex layouts, bespoke type-setting, and often exhibit variations in templates, formats and quality. Although Document AI (DocAI) has made tremendous progress in various tasks including extraction, classification and question answering, there remains a significant performance gap in real-world applications. In particular, accuracy, reliability, contextual understanding and generalization to previously unseen domains continues to be a challenge [2].

Document intelligence is inherently a multi-modal problem with both the text content and visual layout cues being critical to understanding the documents. It requires solutions distinct from conventional large language models such as GPT-3.5 [3], Llama [4], Falcon [5] or PaLM [6] that primarily accept text-only inputs and assume that the documents exhibit simple layouts and uniform formatting, which may not be suitable for handling visual documents. Numerous vision-language frameworks [7, 8] that can process documents as images and capture the interactions between textual and visual modalities are available. However, these frameworks necessitate the use of complex vision backbone architectures [9] to encode image information, and they often make use of spatial information as an auxiliary contextual signal [10, 11].

In this paper we present DocLLM, a light-weight 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

*These authors contributed equally to this work.

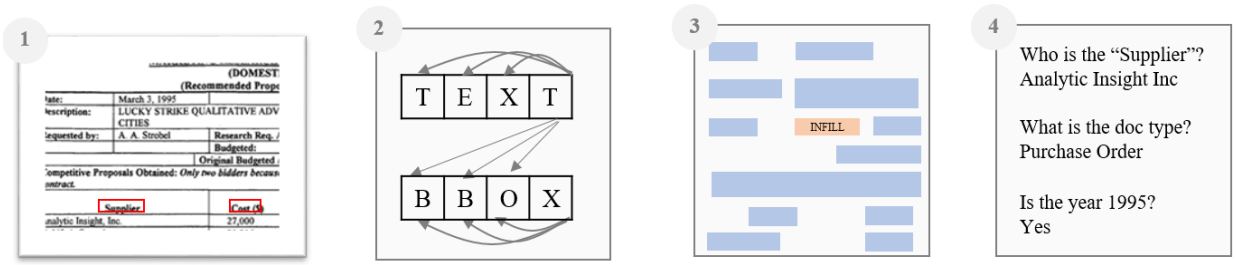


Figure 1: Key elements of DocLLM. (1) Input documents contain text tokens and their bounding boxes. (2) Attention mechanism of LLMs are extended to capture dependencies between text semantics and spatial layouts. (3) Infilling text blocks is used as pre-training objective. (4) Task adaptation is performed on a newly collated dataset of instructions.

intrinsically multi-modal. 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 any 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, as it does not rely on a complex vision encoder. 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 [12] or summing the two [13]. In contrast, we treat the spatial information as a distinct modality and compute its inter-dependency with the text modality in a disentangled manner [14]. In detail, we extend the self-attention mechanism of standard transformers to include new attention scores that capture cross-modal relationships. This is motivated by the observation that there is often a correlation between the content, position and size of the fields in a form. Representing their alignments at various abstraction levels across the transformer layers can enhance document understanding.

A common characteristic of visual documents is their heterogeneous content, irregular layouts, and disjointed text segments. When working with such documents, employing a classical next token prediction objective during the self-supervised pre-training phase can be restrictive. In particular, the preceding tokens may not always be relevant due to the diverse arrangements of text, which can be positioned horizontally, vertically, or even in a staggered manner. 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 [15], our solution is tailored for visual documents with an emphasis on semantically coherent blocks.

We adapt the pre-trained knowledge of DocLLM for several document intelligence tasks by fine-tuning it on instruction data curated from several datasets. These tasks encompass key information extraction, natural language inference, visual question-answering and document classification. Our instruction-tuning data covers both single and multi-page documents. Layout hints such as field separators, titles and captions can be integrated during instruction-tuning to facilitate learning the logical structure of the documents. We observe that the modifications introduced by DocLLM result in a performance improvement ranging from 15% to 61% for the Llama2-7B model in four out of five previously unseen datasets.

Fig. 1 summarizes the framework. Our contributions include:

1. A light-weight 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. An instruction-tuning dataset specially curated towards visual document intelligence tasks.
5. Comprehensive experiments and valuable insights into the model behavior.

2 Related Work

2.1 LLMs

The remarkable success of ChatGPT has generated substantial research interest in LLMs across academia and industry. Subsequently, numerous LLMs have been introduced starting from text-based LLMs [16, 17, 4, 18] to multimodal LLMs [19, 20, 21, 22, 23]. In this section, we review these recent advances in LLMs and discuss their connection to and distinctions from our work.

Text-based LLMs. The introduction of the transformer model in 2017 [24] has been foundational for the pre-trained models such as BERT [25], GPT [26], and T5 [27], each designed with specific pre-training objectives. The emergence of ChatGPT and GPT-4 marked a notable shift, characterized by a substantial increase in both model parameters and training data size. This enhancement has resulted in remarkable zero-shot generalization capabilities, allowing these models to excel in tasks previously unseen. Such success of LLMs has prompted the development of additional LLMs such as OPT [28], BLOOM [18], PaLM [17], and Llama [4]. Particularly, Llama2 [4] is an open-source LLM that achieves comparable or better performance to both open and closed-sourced models, including ChatGPT, PaLM and Falcon, with enhanced safety strategies. Llama2 employs the standard Transformer architecture with pre-normalization [28], SwiGLU activation function [29], and rotary positional embeddings [30]. The pre-training data consists of two trillion tokens from publicly available sources.

Multimodal LLMs. Multimodal LLMs extend the scope of text to diverse modalities, with a focus on visual input. These models can be categorized into two tropes: general-purpose multimodal LLMs [19, 20, 21, 22, 23] and models that are tailored for visually-rich document understanding [31, 32, 33, 34, 12]. The general-purpose multimodal LLMs exhibit promising performance in identifying and reasoning with image information. However, they have not yet been vigorously evaluated on VRDU tasks. As an example, the GPT-4 Technical Report [16] highlights diverse multimodal test cases, such as explaining meme picture distinctiveness, but very few examples are included for visual document use cases. Prior to the advent of large language models, fine-tune-based models relying on vision only were less effective than layout (and vision) modality models in processing visual documents. For example, models like UDOP [12] and LayoutLM [13] outperform vision-only models such as Donut [35] and Pix2Struct [34] in VRDU tasks. But such models require task- and dataset-specific fine-tuning, and are thus excluded in our analysis. The more recent mPLUG-DocOwl [31] and UReader [32], built upon LLMs, undergo instruction finetuning on a diverse set of VRDU, visual, and textual datasets, and exhibit impressive zero-shot generalization capabilities. Hence, we include those as baselines in our evaluation in Section 4.

Despite the remarkable performance of LLMs, unimodal models aren't equipped to process multimodal input, and multimodal LLMs rely on complex and memory intensive open-domain vision encoders. Our proposed model, `DocLLM`, addresses these challenges by explicitly modeling spatial layouts and text semantics, enabling effective comprehension of visual documents. Notably, `DocLLM` offers an extension to the unimodal architecture by adding the spatial signal to text semantics, avoiding the expensive vision encoder, resulting in a more compact model and efficient processing time.

2.2 LLM Architectures

Autoregressive Infilling. There are two main autoregressive infilling approaches: "fill-in-the-middle" (FIM) where a single span is sampled, and "blank infilling" with multiple spans.

The OpenAI FIM approach [36] uses the template `(prefix, middle, suffix)` to divide a document into three segments. Next, these segments are reorganized into `(prefix, suffix, middle)`, enabling the model to predict the middle segment. This process relies on three special tokens, `[PRE]`, `[SUF]`, and `[MID]`, which structure a document as: `[PRE]` prefix `[SUF]` suffix `[MID]` middle. The `[MID]` token denotes the start for prediction, while the other two special tokens guide the model on where to infill. This method demonstrates that autoregressive models can learn to infill text where the middle part is missing. Fill-in Language Model (FiLM) [37] is a subsequent development that enables flexible generation at arbitrary positions, unconstrained by a predefined generation order. In contrast, approaches like GLM [15] sample multiple spans for infilling. For each blank to be infilled, a pair of special tokens is used: `[blank_mask]` and `[start_to_fill]`. The multiple spans not only require special tokens but also global indicators to distinguish which middle span the model should infill. This global indicator is implemented with 1D token positions, ensuring that each pair of the two special tokens, i.e., `[blank_mask]` and `[start_to_fill]`, share the same positions. We adopt a similar infilling object with the goal to prevent disconnected next-token predictions while avoiding breaking sparse documents into very short segments, e.g., word pieces and/or phrase pieces.

Disentangled attention. Disentangled attention is introduced in the DeBERTa model [38], where token embeddings and relative positional encodings were kept separate rather than summed together, and each used independently when computing attention weights using disentangled matrices. The motivation behind this was to facilitate the learning of

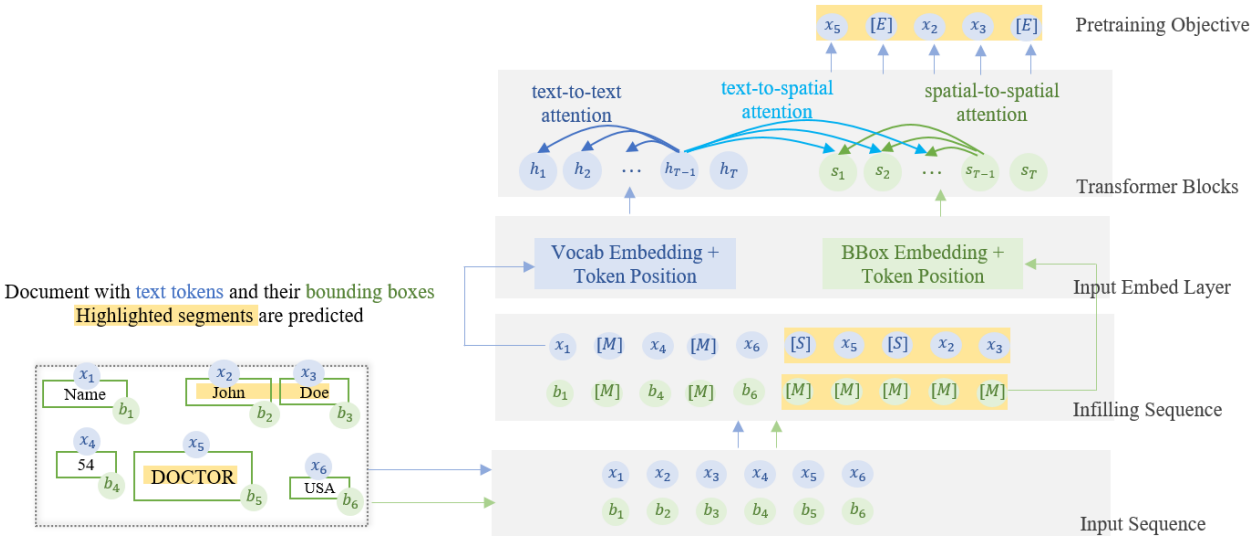


Figure 2: DocLLM model architecture with disentangled spatial attention and infilling objective. *left*: Input document with text tokens x_i and bounding boxes b_i. Some text segments are randomly masked (two segments here) and the model predicts the tokens in these text segments autoregressively. *right*: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

decoupled attention alignments based on content and position separately. This innovation proved effective as it allowed DeBERTa to outperform RoBERTA-large and T5 on NLU benchmarks, as well as to surpass the human baseline on SuperGLUE [39]. In our work, given considerably more complex position encodings used in visually rich documents, disentanglement becomes ever more important to our model's performance.

3 DocLLM Framework

In this section, we discuss the architecture of DocLLM and outline the pre-training and instruction tuning procedures. Figure 2 presents an overview of the model architecture.

3.1 Model Architecture

DocLLM is constructed upon the foundation of an auto-regressive transformer language model [4] following a causal decoder structure. It is composed of stacked transformer blocks, where each block contains a multi-head self-attention layer and a fully connected feed forward network. Standard language models are typically unimodal, accepting only a sequence of text tokens as input. In contrast, DocLLM is a multi-modal system that integrates lightweight visual information by utilizing the spatial positions and dimensions of text tokens obtained using OCR. Simply augmenting the text with bounding box information via additive positional encoding may not capture the intricate relationships between text semantics and spatial layout, especially for visually rich documents [10]. Consequently, we treat the spatial information about the text tokens as a distinct modality. In particular, we use separate vectors to represent these two modalities and extend the self-attention mechanism of the transformer architecture to compute their inter-dependencies in a disentangled manner, as explained in the following section. Furthermore, instead of the traditional left-to-right next token prediction during self-supervised training, we employ a text infilling objective that better leverages contextual information.

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:

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

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:

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

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:

$$\mathbf{Q}^s = \mathbf{S}\mathbf{W}^{s,q}, \qquad \mathbf{K}^s = \mathbf{S}\mathbf{W}^{s,k} \tag{3}$$

$$\mathbf{A}_{i,j} = \mathbf{Q}_i^t\mathbf{K}_j^{t\intercal} + \lambda_{t,s}\mathbf{Q}_i^t\mathbf{K}_j^{s\intercal} + \lambda_{s,t}\mathbf{Q}_i^s\mathbf{K}_j^{t\intercal} + \lambda_{s,s}\mathbf{Q}_i^s\mathbf{K}_j^{s\intercal}, \tag{4}$$

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 [7]. By simply adding $\mathbf{S}$ to $\mathbf{H}$ similar to [13], 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 [38], 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. The self-supervised pre-training objective in autoregressive language models [26] is generally to maximize the log-likelihood of the next token prediction in a sequence based on the context provided by preceding tokens. Let θ denote all the parameters of the transformer model, including the projection matrices discussed above. The following cross-entropy loss is then typically minimized during the pre-training step:

$$\mathcal{L}_{\text{AR}}(\theta) = -\sum_{i=1}^{T} \log p_\theta(x_i|\mathbf{x}_{j<i}) \tag{5}$$

Visual documents are often sparse and irregular, featuring isolated and disconnected text fragments. In such cases, it is preferable to consider coarse segments of related tokens during pre-training rather than focusing on individual tokens. A segment may represent a coherent chunk of information, similar to a text block, or it can simply be a linear sequence, similar to a text span. In Figure 2, "Name", "John Doe" , and "Doctor" are all examples of blocks. In general, the broader context provided by multiple tokens in a block can lead to better comprehension.

Furthermore, learning to infill text, where the prediction is conditioned on both prefix and suffix tokens rather than only preceding tokens, can be beneficial. 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.

Most OCR engines can provide block level information, which makes it feasible to identify coherent text blocks such as a heading or an address[1]. Inspired by [15], 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. Block information and block infilling are solely utilized for the pre-training phase, not in instruct-tuning or downstream tasks.

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 $c_k \cap c_{k'} = \emptyset$. These text blocks are typically identified from OCR information. Let

[1]Note that in order to avoid any leakage of useful information, the block information is only used for the masking objective during pre-training, and is not provided to the model as input. Concretely, masking is performed at the block level, but the model is not provided with information about the number of tokens in a given masked block. Please refer to Figure 2 for an illustrated example.

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: `{choices}`." *(where* `choices` *is a subset of all the keys 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: `{choices}`." *(where* `choices` *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

$\mathbf{z} = \{z_m\}_{m=1}^{M}$ 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 $[\mathrm{M}]$. To facilitate the identification of the block to be filled during text generation, each input block is augmented with a special start token $[\mathrm{S}]$ while the output block includes an end token $[\mathrm{E}]$. For instance, a block with tokens (x_4, x_5) becomes $[\mathrm{M}]$ in $\tilde{\mathbf{x}}$, $([\mathrm{S}], x_4, x_5)$ when conditioned upon, and is expected to generate $(x_4, x_5, [\mathrm{E}])$ as output autoregressively (see Figure 2 for a detailed illustration of these configurations). The following cross-entropy loss is then minimized for the infilling objective.

$$\mathcal{L}_{\mathrm{IF}}(\theta) = -\sum_{m=1}^{M}\sum_{j=1}^{N_m} \log p_\theta(z_{m,j}|\tilde{\mathbf{x}}, \mathbf{z}_{<m}, \mathbf{z}_{m,<j}) \tag{6}$$

3.4 Instruction Tuning

Following recent work in the field of VRDU [12, 31, 32] and prior work in NLP [40, 41], we instruction-tune `DocLLM` on a variety of instructions derived from DocAI datasets using various templates. Due to the high cost and time intensity of manual data collection, we leave the construction of a VRDU instruction-tuning dataset with crowdsourced instructions and preferences to future work. We employ a total of 16 datasets with their corresponding OCRs, spanning four DocAI tasks: visual question answering (VQA), natural language inference (NLI), key information extraction (KIE), and document classification (CLS).

The diversity of supervised fine tuning (SFT) instructions is critical in helping zero-shot generalization [40, 41, 42]. 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 [31, 32] when applicable, and consider a broader selection of datasets in our instruction-tuning data mix.

We create the templates following what we believe end users would generally ask about documents (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). We introduce the templates in detail as follows.

Visual Question Answering. We collect DocVQA [43], WikiTableQuestions (WTQ) [44], VisualMRC [45], DUDE [46], and BizDocs[2], to compose the VQA instruction-tuning data mix. We use one instruction template to build our SFT inputs for VQA, as shown in table 1. An example prompt derived from DocVQA would read: "`{document}` *What is the deadline for scientific abstract submission for ACOG - 51st annual clinical meeting?*"

Natural Language Inference. We only include TabFact [47] in our instruction-tuning data mix for NLI task, due to lack of additional DocAI NLI datasets available. The instruction template is shown in table 1. An example prompt derived from TabFact would read: "`{document}` *\"The UN commission on Korea include 2 Australians.\", Yes or No?*"

Key Information Extraction. We gather Kleister Charity (KLC) [48], CORD [49], FUNSD [50], DeepForm [51], PWC [52], SROIE [53], VRDU ad-buy [54] (with random train-test splitting), and BizDocs to build the KIE instruction-tuning data, where we leverage three instruction templates: extraction, internal classification, and MCQ, as shown in 1. For the

[2]BizDocs is a collection of business entity filings that is due to be released publicly.

Table 2: Pre-training dataset statistics.

	No. of Docs	No. of Pages	No. of Total 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 3: Instruction-tuning dataset statistics.

Tasks	No. of Training	No. of Testing
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

extraction template, we add the "None" answer if the key does not exist in the given document. To increase diversity in the SFT training data, we also derive internal classification and MCQ instructions from original KIE annotations. To stay consistent with benchmarks from previous work [31, 32], we only keep the prompts derived from the extraction template in the test split of each KIE dataset. An example extraction instruction derived from KLC would read: "`{document}` *What is the value for the \"charity number\"?*"

Document Classification. We aggregate RVL-CDIP [55] and BizDocs to build our CLS instruction-tuning data. We used two types of instruction templates for this task: internal classification and MCQ, as shown in 1. To avoid the cold start problem induced by potentially unseen types of documents in testing or even in production usage, we only keep the MCQ prompts for the test split of each CLS dataset. We also downsample RVL-CDIP in the train split to avoid hindering the other datasets. An example MCQ instruction derived from RVL-CDIP would read: "`{document}` *What type of document is this? Possible answers: [budget, form, file folder, questionnaire].*"

4 Experiments

4.1 Datasets

We gather data for pre-training from two primary sources: (1) IIT-CDIP Test Collection 1.0 [56] and (2) DocBank [57]. 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. The relevant statistics for the datasets utilized in the pre-training are detailed in Table 2. We obtain a collection of 16.7 million pages comprising a total of 3.8 billion tokens.

We have introduced the datasets used to conduct instruction tuning on Section 3.4. These datasets encompass four common DocAI tasks: VQA, NLI, KIE, and CLS. Note that when a prompt includes a list of possible answers, we create multiple copies of the prompt with one possible answer assigned to each. We only perform this "flattening" operation in the training split of the dataset. Detailed statistics for these tasks are presented in Table 3.

4.2 Model Setup and Training Details

Table 4 provides key settings and hyperparameters for two variants of `DocLLM`: `DocLLM-1B`, which is based on the Falcon-1B architecture [5], and `DocLLM-7B`, which is based on the Llama2-7B architecture [4][3]. `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 pre-trained 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.

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 [58] with $\beta_1 = 0.9, \beta_2 = 0.96$ and a weight decay of 0.1. For instruction tuning we use a learning

[3]Since Llama2 does not come with pre-trained weights at 1B parameters, we use the Falcon-1B architecture for the smaller version of `DocLLM`.

Table 4: Model configuration and training hyperparameters setting for DocLLM-1B and -7B.

	DocLLM-1B		DocLLM-7B	
Backbone	Falcon-1B [5]		Llama2-7B [4]	
Layers	24		36	
Attention heads	16		32	
Hidden size	1536		4096	
Precision	bfloat16		bfloat16	
Batch size	2		5	
Max context length	1,024		1,024	
	Pre-train	**Instruct-tune**	**Pre-train**	**Instruct-tune**
Learning rate	2×10^{-4}	1×10^{-4}	3×10^{-4}	1×10^{-4}
Warmups	1000	500	1000	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}

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 pre-train for one epoch, and instruct-tune for a total of 10 epochs.

For DocLLM-7B, pre-training 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 pre-training, followed by three epochs of instruct-tuning, considering available computing resources.

The maximum sequence length, or context length, is consistently set to 1,024 for both versions during the entire training process. 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.[4] The DocLLM-1B model, on the other hand, is trained on a single 24GB A10g GPU.

4.3 Downstream Evaluation

Experimental settings. We investigate two experimental settings:

- **Same Datasets, Different Splits** (SDDS): Following previous work in VRDU [34, 59, 33, 12, 31, 32], we first evaluate DocLLM on the unseen test split (or dev split when test split is unavailable) 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 [40, 41, 60, 61], we also evaluate DocLLM on held-out datasets. More precisely, we instruction-tune the pre-trained checkpoint of DocLLM on prompts from 11 of the 16 datasets considered in SDDS, then evaluate DocLLM on the test split of the remaining three 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 (certain datasets were obtained from the same sources), 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 additional experiments with different held-out datasets.

Baselines. In SDDS and STDD, we benchmark DocLLM against comparably-sized and SOTA LLMs using Zero-Shot (ZS) prompts that contain the text extracted from each document using an OCR engine (excluding the spatial information) [4, 42]. In SDDS, we also report numbers from recent DocAI LLMs evaluated in a similar setting [31, 32].

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

Table 5: 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. '-' marks not available.

	Dataset	**GPT-4+OCR** ∼1T (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	**95.9**
CLS	RVL-CDIP	68.2	32.8	-	-	90.9	**91.8**
	BizDocs	84.9	40.9	-	-	98.3	**99.4**

As motivated in section 2, we do not consider DocAI models that require task-specific fine-tuning [33, 59, 34] and/or dataset specific prompts [12], and instead focus on LLMs with out-of-the-box instruction following capability.

Metrics. Following previous work [62, 34, 32, 31], we evaluate all VQA datasets using Average Normalized Levenshtein Similarity (ANLS) [63], with the exception of VisualMRC, for which we use CIDEr [64] and WTQ, for which we use accuracy[5]. Performance on all CLS and NLI datasets is measured using accuracy. We evaluate all KIE datasets with the F1 score.

Results. In the SDDS setting, as shown in the Table 5, we observe 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 GPT-4. GPT-4 outperforms `DocLLM` in VQA, possibly due to the higher complexity of reasoning and abstraction involved in VQA datasets compared to tasks like KIE or CLS. `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`.

In the STDD setting, 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 classification dataset, limiting its ability to generalize effectively to new datasets.

5 Ablation Studies

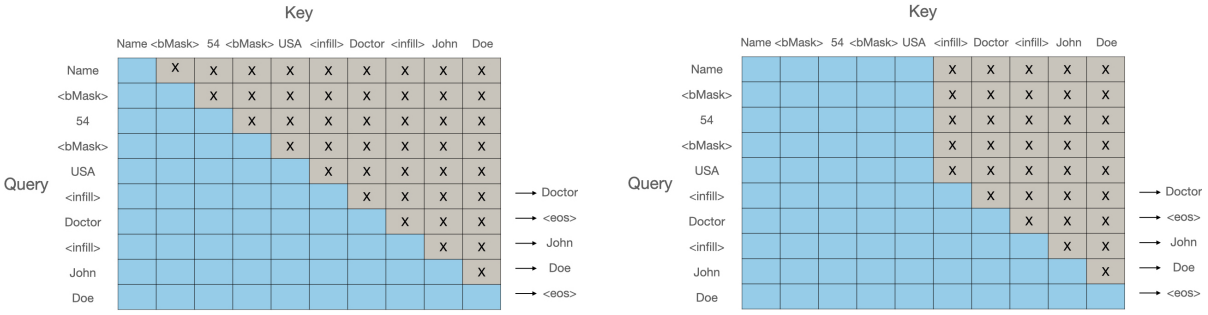
We conduct ablation studies to validate the three contributions of `DocLLM`: (1) disentangled spatial features, (2) the block infilling pre-training objective, and (3) the masking strategy used for decoding.

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 pack of documents concatenated one by one with the total length less than maximum input length. The hyperparameters are set consistently following Table 4 across all ablation experiments.

[5]This is done to remain consistent with the results reported by other SotA models.

Table 6: Performance comparison on three held-out VRDU datasets in the STDD setting against non-multimodal LLMs.

Model	Size	Setting	DocVQA VQA	KLC KIE	BizDocs VQA	BizDocs KIE	BizDocs CLS
GPT-4+OCR	∼1T	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



(a) Causal decoder

(b) Prefix decoder

Figure 3: A simplified illustration of attention masks for causal-decoder and prefix-decoder for block infilling.

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 6 to 0 or 1 . Table 7 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 8, 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 to make the whole document bidirectional visible in the attention, as illustrated in Figure 3b. We investigate this assumption through experiments where we compare a prefix decoder against the conventional

Table 7: Ablation study on disentangled spatial attention. `T` stands for the text modality, `S` stands for the spatial modality, and their cross-modal interactions represent as `X2X`, e.g., text-to-spatial → `T2S`.

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 8: 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**

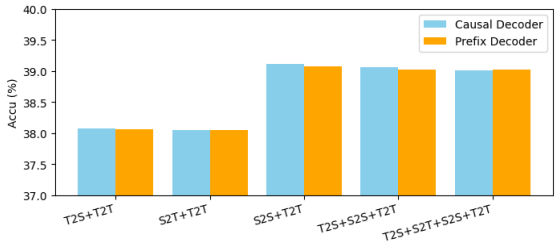


Figure 4: Performance comparison on NTP between causal decoder and prefix decoder.

causal decoder. Specifically, we conduct contrast experiments on these two decoders for different settings outlined in the **disentangled spatial attention** to study their resulting performance.

The results in Figure 4 show marginal differences between these two decoder 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.

6 Discussion and Findings

In addition to its immediate utility in visually rich document understanding tasks, we posit that DocLLM offers an opportunity to change the landscape of generative pre-training by enabling language models to go beyond next token prediction in plain text settings. By accommodating complex layout structures, DocLLM allows for e-books, e-publications, and other documents with rich layouts to be incorporated into the pre-training corpus without requiring extensive preprocessing. The spatial-aware reading approach enables the model to perceive the document as inherently structured knowledge.

Moreover, the multi-page awareness, of both page breaks and document boundaries, enhances the model's ability to comprehend documents of various lengths. This addresses the limitations of previous smaller multi-modal models (which are mainly for single-page documents) and the existing multimodal LLMs (which are primarily designed for images). In supervised instruction tuning, we can adhere to the established practices used in other works, based on desired outputs such as text or images.

The main concept for a cohesive block is to ensure meaningful infilling during the pre-training 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.[6] 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 involves 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 pre-training and instruction-

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

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.

7 Conclusions

In this paper, we introduced DocLLM, a lightweight extension to traditional large language models, tailored for generative reasoning over documents with rich layouts. Unlike existing multimodal LLMs, DocLLM strategically omits costly image encoders, instead prioritizing bounding box information to effectively capture the spatial layout structure of documents. This is achieved through a disentangled attention approach, decomposing the attention mechanism in classical transformers, and enhancing with cross-alignment between text and spatial modalities in structured documents. Notably, our model addresses the challenges posed by irregular layouts and heterogeneous content by employing a pre-training objective that focuses on learning to infill block texts. We fine-tuned the pre-trained model using a comprehensive instruction dataset. Our evaluation across various document intelligence tasks demonstrates that DocLLM surpasses equivalent models on known tasks for 14 datasets out of 16 and exhibits robust generalization to previously unseen datasets in 4 out of 5 settings, affirming its efficacy in extracting meaningful information from a wide range of visual documents. In future work, we plan to infuse vision into DocLLM in a lightweight manner.

Acknowledgments

This paper was prepared for information purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co and its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. J.P. Morgan makes no representation and warranty whatsoever and disclaims all liability for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

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

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

[3] 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. Language models are few-shot learners, 2020.

[4] 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.

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

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

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

[8] 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.

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

[10] 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 Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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, August 2021. Association for Computational Linguistics.

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

[12] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 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, 2023.

[13] 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*, pages 1192–1200, 2020.

[14] Zihang Meng, Licheng Yu, Ning Zhang, Tamara L Berg, Babak Damavandi, Vikas Singh, and Amy Bearman. 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, 2021.

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

[16] OpenAI. Gpt-4 technical report, 2023.

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

[18] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

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

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

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

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

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

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[25] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[26] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[27] 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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[28] Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019.

[29] Noam Shazeer. Glu variants improve transformer, 2020.

[30] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.

[31] 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. mplug-docowl: Modularized multimodal large language model for document understanding. *CoRR*, abs/2307.02499, 2023.

[32] 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. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *CoRR*, abs/2310.05126, 2023.

[33] 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 *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, page 498–517, Berlin, Heidelberg, 2022. Springer-Verlag.

[34] 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 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *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, 2023.

[35] 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, 2022.

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

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

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

[39] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[40] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 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, 2022.

[41] 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. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416, 2022.

[42] 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. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[43] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 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, 2021.

[44] Panupong Pasupat and Percy Liang. 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, 2015.

[45] Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 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, 2021.

[46] 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. Document understanding dataset and evaluation (DUDE). *CoRR*, abs/2305.08455, 2023.

[47] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 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, 2020.

[48] Tomasz Stanislawek, Filip Gralinski, Anna Wróblewska, Dawid Lipinski, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemyslaw Biecek. Kleister: Key information extraction datasets involving long documents with complex layouts. In Josep Lladós, Daniel Lopresti, and Seiichi Uchida, editors, *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, 2021.

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

[50] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 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, 2019.

[51] Stacey Svetlichnaya. Deepform: Understand structured documents at scale. 2020.

[52] Marcin Kardas, Piotr Czapla, Pontus Stenetorp, Sebastian Ruder, Sebastian Riedel, Ross Taylor, and Robert Stojnic. AxCell: Automatic extraction of results from machine learning papers. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8580–8594, Online, November 2020. Association for Computational Linguistics.

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

[54] Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. VRDU: A benchmark for visually-rich document understanding. In Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye, editors, *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, 2023.

[55] Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis. 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, 2015.

[56] 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*, SIGIR '06, page 665–666, New York, NY, USA, 2006. Association for Computing Machinery.

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

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

[59] Brian L. Davis, Bryan S. Morse, Brian L. Price, Chris Tensmeyer, Curtis Wigington, and Vlad I. Morariu. End-to-end document recognition and understanding with dessurt. In Leonid Karlinsky, Tomer Michaeli, and Ko Nishino, editors, *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, 2022.

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

[61] 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. *CoRR*, abs/2306.17107, 2023.

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

[63] Ali Furkan Biten, Rubèn Tito, Andrés Mafla, Lluís Gómez, Marçal Rusiñol, Minesh Mathew, C. V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. 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, 2019.

[64] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 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, 2015.