# GraphMLLM: A Graph-based Multi-level Layout Language-independent Model for Document Understanding

He-Sen Dai[1,2], Xiao-Hui Li[2(✉)], Fei Yin[2], Xudong Yan[3], Shuqi Mei[3], and Cheng-Lin Liu[1,2]

[1] School of Artificial Intelligence,
University of Chinese Academy of Sciences, Beijing, 100049, China
[2] State Key Laboratory of Multimodal Artificial Intelligence Systems,
Institute of Automation of Chinese Academy of Sciences, Beijing, 100190, China
[3] T Lab, Tencent Map,
Tencent Technology (Beijing) Co., Ltd., Beijing, 100193, China
daihesen20@mails.ucas.ac.cn, {xiaohui.li, fyin, liucl}@nlpr.ia.ac.cn,
{owenyan, shawnmei}@tencent.com

**Abstract.** Self-supervised multi-modal document pre-training for document knowledge learning shows superiority in various downstream tasks. However, due to the diversity of document languages and structures, there is still room to better model various document layouts while efficiently utilizing the pre-trained language models. To this goal, this paper proposes a Graph-based Multi-level Layout Language-independent Model (GraphMLLM) which uses dual-stream structure to explore textual and layout information separately and cooperatively. Specifically, GraphMLLM consists of a text stream which uses off-the-shelf pre-trained language model to explore textual semantics and a layout stream which uses multi-level graph neural network (GNN) to model hierarchical page layouts. Through the cooperation of the text stream and layout stream, GraphMLLM can model multi-level page layouts more comprehensively and improve the performance of language-independent document pre-trained model. Experimental results show that compared with previous state-of-the-art methods, GraphMLLM yields higher performance on downstream visual information extraction (VIE) tasks after pre-training on less documents. Code and model will be available at https://github.com/HSDai/GraphMLLM.

**Keywords:** Visual information extraction · Self-supervised pre-training · Multi-level page layouts.

## 1 Introduction

As an important task in Visual Document Understanding (VDU), Visual Information Extraction (VIE) focuses on automated information extraction through Semantic Entity Recognition (SER) and Relationship Extraction (RE) from

Visually-Rich Documents (VRD) including receipts, forms, reports, invoices, etc. It receives widespread attention from both industry and academia for its promising applications.

There have been numerous works of VIE reported in recent years. However, due to the heavy workload of manual annotation, the existing VIE datasets, such as FUNSD [9], XFUND [24], CORD [16] and EPHOIE [20], usually have small scales, severely limiting the performance of deep-learning based VIE methods trained from scratch. To overcome this limitation, many pre-training based methods, such as DocFormer [1], GraphDoc [26], SelfDoc [13], StructuralLM [11], and the LayoutLM [23] series, have been proposed. Different from BERT[4] designed for plain text, pre-training methods for VIE task usually take into consideration the natural multi-modal property of documents and utilize textual, visual and layout information when pre-training their models. Besides, many self-supervised pretext tasks are designed for self-supervised model learning, such as Masked Language Modeling (MLM) [4], Masked Image Modeling (MIM) [2], Masked Visual-Language Model (MVLM) [23], Text Image Alignment (TIA) [22], Text Image Matching (TIM) [22], etc.

The joint learning of multi-modal information can bring significant performance gain on downstream tasks, but it also brings some unexpected disadvantages including the huge data amount required by pre-training and the inflexibility when handling documents of languages not covered by the pre-trained model. Though some works, e.g. LayoutXLM [24], directly use multilingual documents for pre-training to achieve better performance on multilingual dataset XFUND [24], they require even more data for pre-training, yet the pre-trained model still lacks the ability to generalize to unseen languages. Considering that documents with different languages may share similar layouts, LiLT [19] proposes to use dual-stream transformer to decouple text and layout during pre-training, and then re-couples them for downstream task fine-tuning. Through this design, LiLT can be pre-trained on IIT-CDIP [10] which only contains English documents and then adapted to other languages during fine-tuning.

Document understanding can take advantage of the hierarchical nature of document layouts (see Fig. 1), which can provide important guidance for various document understanding tasks. To better excavate document information from multi-level layout structures, some methods including StrucTexT [14], FastStrucTexT [25], LayoutLMv2 [22] and ERNIE-mmLayout [21] have been proposed. These methods either model document hierarchy implicitly through token and segment 1D embedding [14,22] or explicitly exchange and integrate information from different levels and granularities [25,21].

To better exploit the hierarchical structure of layouts for visual document understanding, in this paper we propose a Graph-based Multi-level Layout Language-independent Model (GraphMLLM) that decouples text and layout through a dual-stream structure and explores multi-level layout information more efficiently. By decoupling text and layout, GraphMLLM can directly reuse existing pre-trained language models and greatly reduce the reliance on the amount of pre-training data. Meanwhile, graphs are used to model the multi-level layout

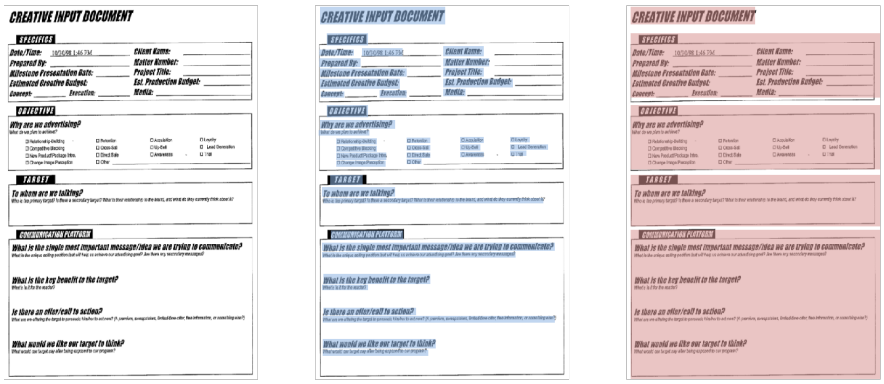


**Fig. 1.** The hierarchical layout structure of documents. Word-level and segment-level layouts in (a) and (b), are typically acquired via OCR engines. Region-level layout is derived through heuristic rules or layout analysis models.

information and interact with semantic modalities through a disentangled attention mechanism, enabling the language model to access different levels of layout information, so as to improve the performance of the entire model.

During pre-training, we use Masked Visual-Language Model, Key Point Location and Cross-modal Alignment Identification as pretext tasks and train our model on the monolingual IIT-CDIP dataset. While during fine-tuning, we conducted experiments on two monolingual datasets FUNSD, CORD and one multilingual dataset XFUND to demonstrate the effectiveness of our model. The experimental results show that despite using fewer data for pre-training, our approach can still outperform other multilingual pre-trained models and obtains competitive results on all tasks compared with state-of-the-art methods.

The contributions of this paper are summarized as follows:

(1) We propose a new multi-modal document pre-training model named GraphMLLM for document understanding, which contains a dual-stream structure to decouple the textual and layout information during pre-training to make it language independent.

(2) We use a multi-level graph neural network (GNN) to model hierarchical document layouts at different granularities, thus exploit the hierarchical structure of documents more efficiently.

(3) The combination of text-layout decoupling and hierarchical layout modeling can significantly reduce the required quantity of pre-training data while still remains high performance on downstream tasks.

(4) We evaluated on three benchmark datasets of different languages, and the experimental results show that GraphMLLM can obtain superior or competitive results compared with state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 introduces the architecture of the proposed model and the pre-training

method. Section 4 and 5 present experimental settings and results, and Section 6 draws concluding remarks.

## 2 Related Work

Here we briefly review existing methods closely related to our work based on the granularity of layout considered: word-level layout based model, segment-level layout based model, and multi-level layout based model.

### 2.1 Word-level Layout Based Model

To perceive layout information, LayoutLM [23] adds token and word position embeddings as initial embeddings with 2D position awareness. In addition to layout information, LayoutLMv2 [22] integrates visual information by gridifing document images (e.g. $7 \times 7$) and achieves *soft alignment* between textual and visual information through pre-training tasks. By focusing on the relationship between texts, BROS [7] improves the attention mechanism by introducing relative position information. DocFormer [1] improves the attention mechanism in integrating textual, layout, and visual features, and proposes a pixel-level image reconstruction task. Furthermore, ERNIE-Layout [17] focuses on document layout information and enables the model to obtain correct reading order through serialization modules and pre-training task for reading order. LayoutXLM [24] emphasizes information extraction from multilingual documents and uses multilingual documents for pre-training. Such word-level layout based models can model fine-grained token-level information but are weak in macroscopic perspective when facing complex layouts.

### 2.2 Segment-level Layout Based Model

Compared to words, segment-level layout (based on text lines, e.g.) is more informative for document understanding. To incorporate higher-level layout information, StructuralLM [11] uses segment-level layout information as the positional embedding of tokens. SelfDoc [13] uses segment-level semantic and layout features, and proposes a multimodal adaptation attention mechanism. Extended to image-centric Document task, LayoutLMv3 [8] adopts segment-level layout information and tokenizes images like the DiT [12] model, without relying on CNN-based visual encoders. Considering local dependencies between segments, GraphDoc [26] uses graph neural networks as the backbone of the pre-trained model. To achieve cross-language transfer capability, LiLT [19] uses a dual-stream Transformer structure with bi-directional attention complementation mechanism (BiACM) to decouple text and layout information. These methods focus on segment-level layout information but ignore the word-level layout information which is also crucial for the comprehension of certain documents such as forms.

### 2.3 Multi-level Layout Based Model

Multi-level layout based models aim to extract layout information of multi-level granularities for better document understanding. StrucTexT [14] adds the token embeddings and word-level layout embeddings as the initial token embeddings, and adds visual features and segment-level layout embeddings together as the initial visual token embeddings, and the word- and segment-level layout embeddings are integrated. As an extension of LayoutLMv2, ERINE-mmLayout [21] introduces additional segment and region information on the basis of LayoutLMv2 without pre-training again, to perceive multi-level layout information and effectively improves the model performance. These works show the effectiveness of integrating multi-level layout information. However, the multi-level layout information is still not utilized sufficiently in that the interactions between inter- and intra-level layouts are not considered in great detail.

In this paper, we try to model multi-level layout information of documents in pre-trained model with better integration and interaction between different levels, so that the model has better cross-language transfer capability at moderate model complexity and data reliance.

## 3 GraphMLLM

Inspired by the disentangled attention mechanism proposed in DeBERTa [6] and the decoupled modeling in LiLT, GraphMLLM (see Fig. 2) adopts a dual-stream structure to encode text features and layout features separately, and interacts information between two modalities through an attention based hierarchical interaction mechanism. In the following, the multi-level graph representation of documents is first introduced, followed by the text flow module, layout flow module and interaction mechanism between these two modalities.

### 3.1 Document Representation

First, all the texts in documents along with their coordinates are extracted using an OCR engine provided by ReSenseTech (other OCR software can be used alternatively)[1], and are transformed into text and layout representations using token embeddings and hierarchical layout embeddings, respectively.

**Text Representation.** Since the reading order obtained by OCR is noisy, we use XY Cut [5] to obtain a proper reading order of texts. Then, like many pre-trained language models, we serialize and tokenize the texts and add special tokens `[CLS]` and `[SEP]` at the beginning and end to obtain the text token sequence: $T = [t_1, ..., t_{N_t}]$, where $N_t$ stands for the token sequence length. By adding token embeddings and 1D position embeddings, 1D-position aware token embeddings are obtained:

$$\boldsymbol{E}_t = \mathrm{LN}(\boldsymbol{E}_{token} + \boldsymbol{E}_{1D_{pos}}), \tag{1}$$

[1] http://www.resensetech.com

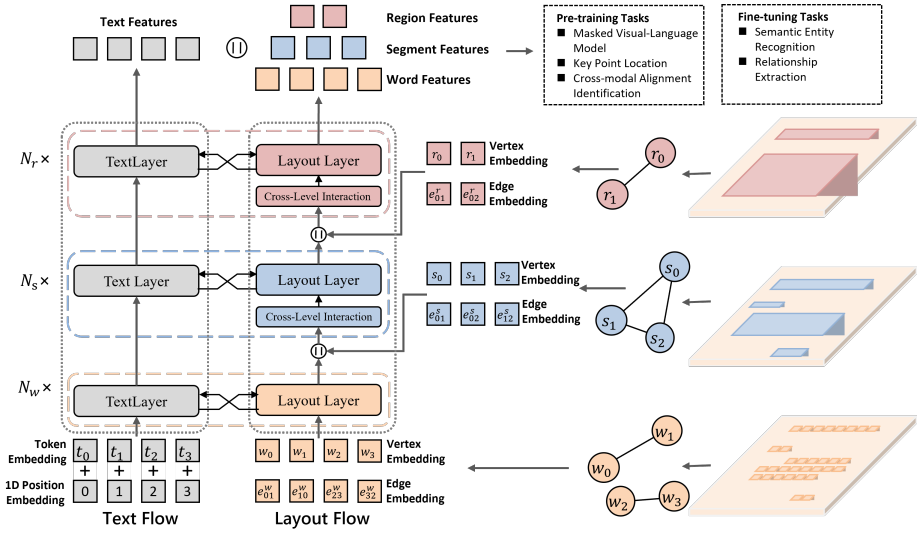


**Fig. 2.** The overall architecture of GraphMLLM, which consists of two streams: the *Text Flow* and the *Layout Flow*. It decouples text and layout information, and uses hierarchical graphs to model multi-level layouts. Higher level layouts are successively added to the model, $N_w$, $N_s$ and $N_r$ are layer numbers of each stage. In each layer, layout features are firstly interacted cross levels, then information from text and layout modalities are interacted through disentangled attention. Best viewed in zoomed-in.

where $\boldsymbol{E}_{token} \in \mathbb{R}^{N_t \times D_t}$ is token embedding matrix, $\boldsymbol{E}_{1D_{pos}} \in \mathbb{R}^{N_t \times D_t}$ is position embedding matrix, $D_t$ is feature dimension and LN is Layer Normalization.

**Multi-Layer Layout Representation.** Document layout information can be represented at three levels: word, segment, and region, represented by $W = \{w_1, ..., w_{N_w}\}$, $S = \{s_1, ..., s_{N_s}\}$ and $R = \{r_1, ..., r_{N_r}\}$ respectively. Here, $N_w, N_s, N_r$ are the number of words, segments, and regions for the given document. The layout is represented as a graph structure $G = (V, E)$, where vertices $V = W \cup S \cup R$ denote layout elements (words, segments and regions), and edges $E = E_{ww} \cup E_{ss} \cup E_{rr}$ denote the connections between vertices of the same level.

Fellowing GraphDoc [26], the vertex and edge embeddings $\boldsymbol{E}_v \in \mathbb{R}^{N_v \times D_v}$, $\boldsymbol{E}_e \in \mathbb{R}^{N_e \times D_e}$ of graph are obtained as follows:

$$\boldsymbol{E}_v = Concat(\boldsymbol{E}_x(x_0, x_1, w), \boldsymbol{E}_y(y_0, y_1, h)), \tag{2}$$

where $x_0, y_0, x_1, y_1$ denote the left, top, right, bottom coordinates, and $h, w$ denote height and width of bounding boxes, $\boldsymbol{E}_x, \boldsymbol{E}_y$ are learnable position embeddings. Similarly,

$$\boldsymbol{E}_e = \boldsymbol{E}_{tl}\boldsymbol{W}_{tl} + \boldsymbol{E}_{tr}\boldsymbol{W}_{tr} + \boldsymbol{E}_{bl}\boldsymbol{W}_{bl} + \boldsymbol{E}_{br}\boldsymbol{W}_{br}, \tag{3}$$

where $\boldsymbol{W}_{tl}, \boldsymbol{W}_{tr}, \boldsymbol{W}_{bl}, \boldsymbol{W}_{br}$ are learnable parameters, and $\boldsymbol{E}_{tl}, \boldsymbol{E}_{tr}, \boldsymbol{E}_{bl}, \boldsymbol{E}_{br}$ are sinusoidal position embeddings of distance of top-left, top-right, bottom-left and bottom-right coordinates of the vertex bounding boxes, which are calculated as follows:

$$\boldsymbol{E}_{dist} = Concat(\mathrm{PE}(x_{dist}), \mathrm{PE}(y_{dist})), \tag{4}$$

where $dist \in \{tl, tr, bl, br\}$, PE is a sinusoidal function [18], and $x_{dist}$ and $y_{dist}$ represent the horizontal and vertical distances of the corresponding coordinates.

For relationship between elements of different levels, we use matrixes $\boldsymbol{M}_{ws} \in \mathbb{R}^{N_w \times N_s}$ and $\boldsymbol{M}_{sr} \in \mathbb{R}^{N_s \times N_r}$ to represent the relationship between word-segment levels and between segment-region levels, respectively. Taking $\boldsymbol{M}_{ws}$ as an example, each $m_{ij}$ in the matrix indicates whether the word $w_i$ belongs (set as 1) to the segment $s_j$ or not (set as 0).

**Text-Layout Alignment Relationship.** The correspondence between text and layout is many-to-many. To facilitate the calculation, we use matrices $\boldsymbol{M}_{tw} \in \mathbb{R}^{N_t \times N_w}$, $\boldsymbol{M}_{ts} \in \mathbb{R}^{N_t \times N_s}$, $\boldsymbol{M}_{tr} \in \mathbb{R}^{N_t \times N_r}$ denote the correspondence of text-to-word-layout, text-to-segment-layout, text-to-region-layout, respectively. Taking $\boldsymbol{M}_{ts}$ as an example, each $m_{ij}$ in the matrix indicates whether the text $t_i$ belongs (set as 1) to the segment $s_j$ or not (set as 0).

### 3.2 Text Flow

For extracting semantic features from text sequence, the backbone of text flow adopts a pre-trained language model consisting of several Transformer encoder layers [18]. Specifically, we input the token embeddings $\boldsymbol{E}_t$ into the text flow to derive semantic features with contextual information. As depicted in Fig. 3 and Fig. 4, for the $k$-th layer of the text flow:

$$\boldsymbol{H}_t^{k*} = \mathrm{LN}(\mathrm{MHA_t}(\boldsymbol{H}_t^k, \boldsymbol{H}_v^k) + \boldsymbol{H}_t^k), \tag{5}$$

$$\boldsymbol{H}_t^{k+1} = \mathrm{LN}(\mathrm{FFN}(\boldsymbol{H}_t^{k*}) + \boldsymbol{H}_t^{k*}), \tag{6}$$

where $\boldsymbol{H}_t^k \in \mathbb{R}^{N_t \times D_t}$ and $\boldsymbol{H}_v^k \in \mathbb{R}^{N_v \times D_v}$ are the token and layout features input to the $k$-th text layer, $\boldsymbol{H}_t^{k+1} \in \mathbb{R}^{N_t \times D_t}$ is the output of the $k$-th text layer, FFN is the Feed-Forward Network [18], $\mathrm{MHA_t}$ is the multi-head muti-modal self-attention mechanism, which will be described in detail in Section 3.4.

### 3.3 Layout Flow

In the layout flow, we use multi-level graph attention network to compute hidden representations of document layouts focusing on neighbouring features. We employ a bottom-up strategy to increasingly extract higher-level layout features. Specifically, the first $k_w$ layers are designed to extract solely word-level layout features. Following this, the next $k_s$ layers incorporate segment-level layout features into the model. Finally, the last $k_r$ layers introduce region-level layout information to achieve a comprehensive representation of the document layout. The single-layer implementation of layout flow and inter-layer interaction are described separately in the following parts of this subsection.

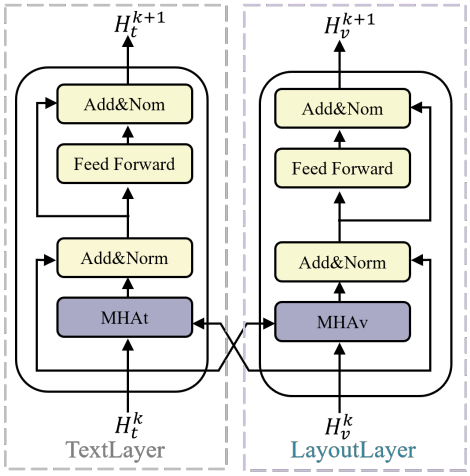


**Fig. 3.** Text-Layout Layer.

**Layout Layer** Following GraphDoc [26], we use graph attention layers to compute the hidden representation of layout tokens, by attending over its neighbors following a self-attention strategy. As depicted in Fig. 3 and Fig. 4, for the $k$-th layer of the layout flow:

$$\boldsymbol{H}_v^{k*} = \mathrm{LN}(\mathrm{MHA_v}(\boldsymbol{H}_v^k, \boldsymbol{H}_t^k, \boldsymbol{E}_e) + \boldsymbol{H}_v^k), \tag{7}$$

$$\boldsymbol{H}_v^{k+1} = \mathrm{LN}(\mathrm{FFN}(\boldsymbol{H}_v^{k*}) + \boldsymbol{H}_v^{k*}), \tag{8}$$

where $\boldsymbol{H}_v^k \in \mathbb{R}^{N_v \times D_v}$ and $\boldsymbol{H}_t^k \in \mathbb{R}^{N_t \times D_t}$ are the layout and text features input to the $k$-th layer, and $\boldsymbol{E}_e \in \mathbb{R}^{N_e \times D_v}$ is the initialized edge embedding, and $\mathrm{MHA_v}$ is the graph attention mechanism, which will be described in Section 3.4.

**Cross-level Interaction** In order to enhance the multi-level layout representation ability of the layout flow, we consider interactions between different levels. Taking as an example the word-level layout feature $\boldsymbol{H}_w$ and segment-level feature $\boldsymbol{H}_s$ in one layer of the middle $k_s$ layout layers:

$$\boldsymbol{H}_s = \mathit{AvePooling}(\boldsymbol{H}_w, \boldsymbol{M}_{ws}) + \boldsymbol{H}_s, \tag{9}$$

$$\boldsymbol{H}_w = \boldsymbol{M}_{ws}\boldsymbol{H}_s + \boldsymbol{H}_w, \tag{10}$$

where *AvePooling* performs the average pooling operation among all words belonging to the same segments. Taking node $s_i$ as an example, its corresponding feature is denoted as $\boldsymbol{h}_{s_i}$:

$$\boldsymbol{h}_{s_i} = \frac{1}{|\mathcal{E}(s_i)|} \sum_{w_j \in \mathcal{E}(s_i)} \boldsymbol{h}_{w_j} + \boldsymbol{h}_{s_i}, \tag{11}$$

where $\boldsymbol{h}_{s_i} \in \mathbb{R}^{D_v}$, $\boldsymbol{h}_{w_j} \in \mathbb{R}^{D_v}$, and $\mathcal{E}(s_i) = \{w_k | m_{ki} = 1, m_{ki} \in M_{wr}\}$.

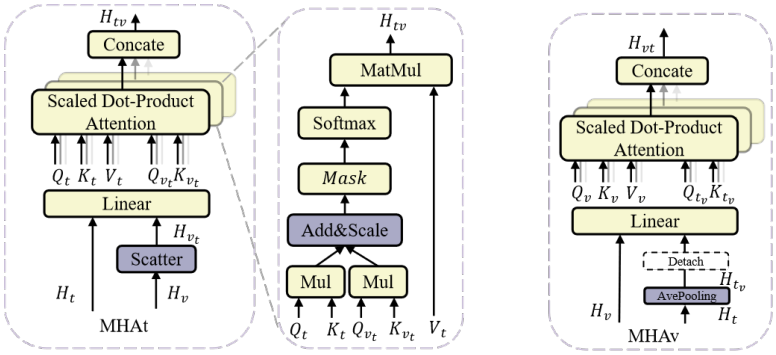


**Fig. 4.** Disentangled Interaction Mechanism.

### 3.4 Cross-Modal Interaction Mechanism

As discussed earlier, it is possible to decouple layout and text features and interact only through attention scores. Here we give its details as follows.

**Text Flow Attention Mechanism.** $\text{MHA}_\text{t}$ is a disentangled attention mechanism in which attention weights among tokens are computed using disentangled attention matrices on their contents and layouts.

First, the layout features of tokens $\boldsymbol{H}_{v_t} \in \mathbb{R}^{N_t \times D_t}$ are obtained through operation:

$$\boldsymbol{H}_{v_t} = Concate(\boldsymbol{M}_{tv}\boldsymbol{H}_v, \boldsymbol{E}_{1D_{lay}}), \tag{12}$$

where $\boldsymbol{M}_{tv} \in \{\boldsymbol{M}_{tw}, \boldsymbol{M}_{ts}, \boldsymbol{M}_{tr}\}$ according to the layout type, $\boldsymbol{E}_{1D_{lay}} \in \mathbb{R}^{N_t \times D_t}$ is 1D position embedding matrix similar to $\boldsymbol{E}_{1D_{pos}}$.

Then, we map token features $\boldsymbol{H}_t$ and layout features $\boldsymbol{H}_{v_t}$ to $\boldsymbol{Q}_t$, $\boldsymbol{K}_t$, $\boldsymbol{V}_t$, $\boldsymbol{Q}_{v_t}$, $\boldsymbol{K}_{v_t}$, $\boldsymbol{V}_{v_t}$:

$$\boldsymbol{Q}_t, \boldsymbol{K}_t, \boldsymbol{V}_t = \boldsymbol{H}_t\boldsymbol{W}_{Qt}, \boldsymbol{H}_t\boldsymbol{W}_{Kt}, \boldsymbol{H}_t\boldsymbol{W}_{Vt}, \tag{13}$$

$$\boldsymbol{Q}_{v_t}, \boldsymbol{K}_{v_t}, \boldsymbol{V}_{v_t} = \boldsymbol{H}_{v_t}\boldsymbol{W}_{Q_{vt}}, \boldsymbol{H}_{v_t}\boldsymbol{W}_{K_{vt}}, \boldsymbol{H}_{v_t}\boldsymbol{W}_{V_{vt}}, \tag{14}$$

where $\boldsymbol{W}_{*t} \in \mathbb{R}^{D_t \times D_t}$ and $\boldsymbol{W}_{*vt} \in \mathbb{R}^{D_{v_t} \times D_t}$ are learnable parameters.

Then, the contextualized representation output is obtained by taking a weighted sum of the values based on the attention weights:

$$\boldsymbol{A}_{tv} = \boldsymbol{Q}_t\boldsymbol{K}_t^T + \boldsymbol{Q}_{v_t}(\boldsymbol{K}_{v_t})^T, \tag{15}$$

$$\boldsymbol{H}_{tv} = Softmax(\frac{\boldsymbol{A}_{tv}}{\sqrt{D_t}})\boldsymbol{V}_t, \tag{16}$$

where $\boldsymbol{A}_{tv} \in \mathbb{R}^{N_t \times N_t}$ is the attention score matrix.

**Layout Flow Attention Mechanism.** $\mathrm{MHA_v}$ is also a disentangled attention mechanism that differs from $\mathrm{MHA_t}$ in that it relies only on local context.

First, the text features corresponding to each layout element $\boldsymbol{H}_{t_v} \in \mathbb{R}^{N_v \times D_t}$ are obtained by averaging pooling:

$$\boldsymbol{H}_{t_v} = AvePooling(\boldsymbol{H}_t, \boldsymbol{M}_{tv}), \tag{17}$$

Similarly to Eq. 13 and Eq. 14, we map text and layout features to get $\boldsymbol{Q}_{t_v}, \boldsymbol{K}_{t_v}, \boldsymbol{V}_{t_v}$ and $\boldsymbol{Q}_v, \boldsymbol{K}_v, \boldsymbol{V}_v$. Then, we use the following operations to obtain the attention matrix $\boldsymbol{A}_{vt} \in \mathbb{R}^{N_v \times N_v}$ and layout features $\boldsymbol{H}_{vt} \in \mathbb{R}^{N_v \times D_v}$ after fusing the neighbor information:

$$\boldsymbol{A}_{vt} = \boldsymbol{Q}_v \boldsymbol{K}_v^T + Reshape(\boldsymbol{Q}_v \boldsymbol{E}_e^T) + \boldsymbol{Q}_{t_v} \boldsymbol{K}_{t_v}^T, \tag{18}$$

$$\boldsymbol{A}_g = Mask(\boldsymbol{A}_{vt}, \boldsymbol{M}_v), \tag{19}$$

$$\boldsymbol{H}_{vt} = Softmax(\frac{\boldsymbol{A}_g}{\sqrt{D_v}})\boldsymbol{V}_v, \tag{20}$$

where $Mask$ is the masking operation [18] according to the matrix $\boldsymbol{M}_v$ which is the adjacency matrix of layout elements according to layout graph $G$, $Reshape$ is reshaping operation to match the shape of $\boldsymbol{A}_{vt}$.

### 3.5 Gradient Detach Operation

Following LiLT [19], to ensure the linguistic independence of the text flow, the gradient back propagation from the layout flow to the text flow needs to be terminated during pre-training. Specifically, during pre-training Eq. 17 is replaced as the following operation:

$$\boldsymbol{H}_{t_v} = Detach(AvePooling(\boldsymbol{H}_t, \boldsymbol{M}_{tv})) \tag{21}$$

where $Detach$ is the gradient detach operation, i.e., the gradient back propagation is not continued from here. The gradient detach operation can mitigate the gradient impact of the layout flow on the text flow, thus enhancing the cross-linguistic capability of model.

## 4 Experiment Setting

### 4.1 Pre-training Tasks

**Masked Visual-Language Model.** Masked Visual-Language Model (MVLM) [23] is a pre-training task for model to learn linguistic representations. During pre-training, 15% of the tokens are randomly masked, of which 80% are replaced by special tokens "[MASK]", 10% are replaced by random tokens sampled from the entire vocabulary, and the last 10% remain unchanged. The goal of this task is to predict the tokens masked in the text.

**Key Point Location.** Key Point Location (KPL) [19] is a pre-training task for model to learn the layout representation using surrounding layout information. During training, 15% of the text bounding boxes are randomly masked, of which 80% are replaced by [0,0,0,0], 10% are replaced by random boxes sampled from the same batch, and the last 10% remain unchanged. The target is to predict the key points (top left, center, bottom right) of each bounding boxes belonging to certain regions (the document is divided equally into 49 regions).

**Cross-modal Alignment Identification.** Cross-modal alignment identification (CAI) [19] is a pre-training task for aligning tokens and bounding boxes. In pre-training, the token-box pairs of encoded tokens are collected, and the training goal is to predict whether each pair has undergone a replacement operation. Since all three pre-training tasks are classification tasks, we use fully connected layers for classification and their losses are all calculated using cross-entropy loss.

### 4.2 Downstream Tasks

**Semantic Entity Recognition.** The goal of Semantic Entity Recognition (SER) [24] is to extract semantic entities from a set of tokens. Specifically, given a document $D$ with a sequence of tokens $T = [t_0, t_1, ..., t_n]$ and target tags $C = \{c_0, c_1, ..., c_m\}$, it is required to predicted semantic entities:

$$\mathcal{E} = \{([x_0^0, ..., x_0^{n_0}], c_0), ..., ([x_k^0, ..., x_k^{n_k}], c_k)\} \tag{22}$$

where $x_k^{n_k} \in T$ and $n_k$ is the length of the $k$-th extracted entity.

**Relationship Extraction.** Relationship Extraction (RE) [24] is to extract the relationships between entities. Specifically, given a set of entities of document $D$ and the semantic relation labels $R = \{r_0, r_1, ..., r_m\}$, it is required to predict a set of semantic relations:

$$\mathcal{L} = \{([head_0, tail_0], r_0), ..., ([head_k, tail_k], r_k)\} \tag{23}$$

where $head_k$ and $tail_k$ are two semantic entities and $r_k$ is the relation between them. In this work, we mainly focus on the key-value relation extraction..

## 5 Experiments

### 5.1 Datasets

**IIT-CDIP:** IIT-CDIP [10] is a large-scale dataset of scanned English document images, containing over 6 million documents and over 11 million scanned document images. This dataset is used for self-supervised pre-training.
**FUNSD:** FUNSD [9] is a scanned English form dataset for the form understanding task. It is divided into a training set containing 149 samples and a

**Table 1.** SER results on English datasets of FUNSD and CORD. "#Docs" represents the number of documents utilized for pre-training, measured in millions (M). "W", "S" and "R" denote word-level, segment-level and region-level layouts, respectively. **Bold** implicates the best results and underline the second best results.

| Model | #Docs | Layout | FUNSD | | | CORD | | |
|---|---|---|---|---|---|---|---|---|
| | | | Precision | Recall | F1 | Precision | Recall | F1 |
| $\text{BERT}_{\text{BASE}}$ [4] | — | — | 0.5469 | 0.6710 | 0.6026 | 0.8833 | 0.9107 | 0.8968 |
| $\text{RoBERTa}_{\text{BASE}}$ [15] | — | — | 0.6349 | 0.6975 | 0.6648 | — | — | — |
| $\text{LayoutLM}_{\text{BASE}}$ [23] | 11M | W | 0.7597 | 0.8155 | 0.7866 | 0.9437 | 0.9508 | 0.9472 |
| $\text{LayoutLMv3}_{\text{BASE}}$ [8] | 11M | S | — | — | **0.9029** | — | — | 0.9656 |
| GraphDoc [26] | 0.32M | S | — | — | 0.8795 | — | — | **0.9693** |
| $\text{LayoutXLM}_{\text{BASE}}$ [24] | 30M | W | — | — | 0.794 | — | — | — |
| LiLT [19] | 11M | S | 0.8721 | **0.8965** | 0.8841 | 0.9598 | 0.9616 | 0.9607 |
| GraphMLLM | 2M | W | 0.7591 | 0.7955 | 0.7769 | 0.9313 | 0.9431 | 0.9372 |
| GraphMLLM | 2M | WS | 0.8623 | 0.8830 | 0.8725 | 0.9515 | 0.9536 | 0.9525 |
| GraphMLLM | 2M | WSR | 0.8616 | 0.8840 | 0.8727 | 0.9537 | 0.9558 | 0.9548 |
| GraphMLLM | 11M | WSR | **0.8835** | 0.8870 | 0.8852 | **0.9620** | **0.9656** | 0.9638 |

test set containing 50 samples. Each document contains four types of entities: `question`, `answer`, `heading`, and `other`.

**XFUND:** XFUND [24] is a multilingual document understanding dataset extended from the FUNSD dataset. The languages of documents are extended from English (EN) to seven other languages, including Chinese (ZH), Japanese (JA), Spanish (ES), French (FR), Italian (IT), German (DE), and Portuguese (PT). Each language includes 199 forms, among which 149 forms are used for training and the other 50 forms are used for testing.

**CORD:** CORD [16] is a receipt dataset with a training set containing 800 samples, a validation set containing 100 samples, and a test set containing 100 samples. The dataset defines 30 fields under 4 categories and the task aims to label each word to the right field.

For pre-training on the IIT-CDIP dataset, the OCR engine is utilized to extract texts along with their bounding boxes. While for fine-tuning on FUNSD, CORD and XFUND, the official OCR annotations are used.

### 5.2 Evaluation Metrics

For the SER task and RE task, we use entity-level F1 score and pair-level F1 score as the evaluation metrics, respectively. Using the same settings as LayoutXLM [24] and LiLT [19], we evaluate the performance of the pre-trained model in three main settings: 1) fine-tuning and testing on a specific language; 2) fine-tuning on English data and then testing on multilingual data (zero-shot learning); 3) fine-tuning on all language data and testing on individual language data.

### 5.3 Implementation Details

For word-level and segment-level layout, we use the k-Nearest Neighbours algorithm to build the graph, with k values set to 100 and 50, respectively. While for regional-level layout, we initially employs a rule-based method which clustered

**Table 2.** Language-specific fine-tuning F1 accuracy on FUNSD and XFUND.

| Task | Model | Pretrain Docs | | FUNSD | XFUND | | | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Language | Size | EN | ZH | JA | ES | FR | IT | DE | PT | |
| SER | LayoutXLM | Multilingual | 30M | 0.794 | 0.8924 | 0.7921 | 0.755 | 0.7902 | 0.8082 | 0.8222 | 0.7903 | 0.8056 |
| | LiLT | English only | 11M | 0.8415 | 0.8938 | 0.7964 | 0.7911 | 0.7953 | 0.8376 | 0.8231 | 0.822 | 0.8251 |
| | LiLT | English only | 2M | — | — | — | — | — | — | — | — | 0.7963 |
| | GraphMLLM | English only | 2M | 0.8403 | **0.9080** | **0.8034** | 0.7954 | 0.8374 | 0.8458 | 0.8481 | 0.8301 | 0.8386 |
| | GraphMLLM | English only | 11M | **0.8553** | 0.9041 | 0.7971 | **0.8222** | **0.8578** | **0.8666** | **0.8581** | **0.8444** | **0.8507** |
| RE | LayoutXLM | Multilingual | 30M | 0.5483 | 0.7073 | 0.6963 | 0.6896 | 0.6353 | 0.6415 | 0.6551 | 0.5718 | 0.6432 |
| | LiLT | English only | 11M | 0.6276 | 0.7297 | 0.7037 | 0.7195 | 0.6965 | 0.7043 | 0.6558 | 0.5874 | 0.6781 |
| | GraphMLLM | English only | 2M | 0.6462 | **0.7734** | **0.7178** | 0.6832 | 0.6781 | **0.7172** | **0.6744** | 0.5888 | 0.6849 |
| | GraphMLLM | English only | 11M | **0.7116** | 0.7657 | 0.7173 | **0.7327** | **0.7142** | 0.7123 | 0.6685 | **0.6100** | **0.7040** |

**Table 3.** Cross-lingual zero-shot transfer F1 accuracy on FUNSD and XFUND.

| Task | Model | Pretrain Docs | | FUNSD | XFUND | | | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Language | Size | EN | ZH | JA | ES | FR | IT | DE | PT | |
| SER | LayoutXLM | Multilingual | 30M | 0.7940 | 0.6019 | 0.4715 | 0.4565 | 0.5757 | 0.4846 | 0.5252 | 0.5390 | 0.5561 |
| | LiLT | English only | 11M | 0.8415 | 0.6152 | 0.5184 | 0.5101 | 0.5923 | 0.5371 | 0.6013 | 0.6325 | 0.6061 |
| | GraphMLLM | English only | 2M | 0.8403 | 0.6102 | 0.5118 | 0.5104 | 0.6030 | 0.5446 | 0.5854 | **0.6387** | 0.6056 |
| | GraphMLLM | English only | 11M | **0.8553** | **0.6404** | **0.5266** | **0.5374** | **0.6507** | **0.5953** | **0.6356** | 0.6353 | **0.6346** |
| RE | LayoutXLM | Multilingual | 30M | 0.5483 | 0.4494 | 0.4408 | 0.4708 | 0.4416 | 0.4090 | 0.3820 | 0.3685 | 0.4388 |
| | LiLT | English only | 11M | 0.6276 | 0.4764 | 0.5081 | 0.4968 | 0.5209 | 0.4697 | 0.4169 | 0.4272 | 0.4930 |
| | GraphMLLM | English only | 2M | 0.6462 | 0.5667 | 0.5811 | 0.5453 | 0.5852 | 0.5022 | 0.4912 | 0.4330 | 0.5439 |
| | GraphMLLM | English only | 11M | **0.7116** | **0.6395** | **0.6405** | **0.6169** | **0.6814** | **0.5919** | **0.5680** | **0.5255** | **0.6219** |

regions based on the distance between segments. However, we found significant inconsistency between documents, thus we ultimately resorted to utilizing bounding boxes containing all text to provide the global layout feature of the whole document. We use a 12-layer, 12-heads text-layout model, where the hidden layer dimensions for text layers and layout layers are set as 768 and 192, respectively. The parameters $k_w$, $k_s$, and $k_r$ of each layout level are all set to 4.

GraphMLLM is pre-trained using Adam with the learning rate $2e^{-5}$, weight decay $1e^{-2}$, and $(beta1, beta2) = (0.9, 0.999)$. The learning rate is linearly warmed up over the first 10% steps then linearly decayed. We set the batch size as 80 and train GraphMLLM for 2 epochs on the partial/full IIT-CDIP dataset using 4 NVIDIA A6000 48GB GPUs. Parameters of text flow are initialized with $\text{RoBERTa}_{\text{BASE}}$ [15], while parameters of layout flow are initialized from random. For multilingual downstream tasks, we load $\text{InfoXLM}_{\text{BASE}}$ [3] parameters to initialize the text flow, making GraphMLLM multi-language capable.

### 5.4 Main Results

**Language-specific Fine-tuning** First, we conducted experiments for GraphMLLM using different levels of layout information. The SER results of English dataset fine-tuning in Table 1 show that GraphMLLM using multi-level layouts performs better than using word-level layout alone. This justifies that using multi-level layout features can effectively improve the model's ability to understand documents. With only text and layout information as inputs, GraphMLLM can achieve competitive results using less training data than previous methods.

To validate the cross-language capability of GraphMLLM, we also evaluate it on XFUND. Table 2 shows that GraphMLLM with less pre-training data meets

**Table 4.** Multitask fine-tuning F1 accuracy on FUNSD and XFUND.

| Task | Model | Pretrain Docs | | FUNSD | XFUND | | | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Language | Size | EN | ZH | JA | ES | FR | IT | DE | PT | |
| SER | LayoutXLM | Multilingual | 30M | 0.7924 | 0.8755 | 0.7964 | 0.7798 | 0.8173 | 0.8210 | 0.8322 | 0.8241 | 0.8201 |
| | LiLT | English only | 11M | 0.8574 | 0.9047 | 0.8088 | 0.8340 | 0.8577 | 0.8792 | 0.8769 | 0.8493 | 0.8585 |
| | GraphMLLM | English only | 2M | 0.8737 | 0.9113 | 0.8079 | 0.8523 | 0.8854 | 0.8806 | **0.8881** | 0.8603 | 0.8700 |
| | GraphMLLM | English only | 11M | **0.8920** | **0.9178** | **0.8194** | **0.8573** | **0.9013** | **0.9033** | 0.8830 | **0.8699** | **0.8805** |
| RE | LayoutXLM | Multilingual | 30M | 0.6671 | 0.8241 | 0.8142 | 0.8104 | 0.8221 | 0.8310 | 0.7854 | 0.7044 | 0.7823 |
| | LiLT | English only | 11M | 0.7407 | 0.8241 | 0.8345 | 0.8335 | 0.8466 | 0.8458 | 0.7878 | 0.7643 | 0.8125 |
| | GraphMLLM | English only | 2M | 0.8298 | **0.8946** | 0.8456 | **0.8533** | **0.8860** | **0.8641** | **0.8315** | **0.7836** | 0.8486 |
| | GraphMLLM | English only | 11M | **0.8867** | 0.8911 | **0.8756** | 0.8472 | 0.8791 | 0.8468 | 0.8301 | 0.7828 | **0.8549** |

or exceeds the performance of previous multilingual models, which illustrates the superior cross-language capability and efficient data utilization of GraphMLLM. In comparison to LiLT, which solely relies on single-granularity layout information, GraphMLLM exhibits superior performance even with less pre-training documents. This underscores the significance of multi-level graph-based structure in modeling document layout.

**Zero-shot Transfer Learning** Table 3 presents the results of cross-language zero-shot transfer learning. According to the results, the GraphMLLM model has a outstanding zero-shot transfer capability without applying multiple language documents for pre-training, and it outperforms previous counterpart models. Due to the multi-level layout flow, the GraphMLLM model is able to effectively model the layout structure of documents.

**Multi-task Fine-tuning** Table 4 shows the experimental results of GraphMLLM fine-tuned using data from eight languages. GraphMLLM achieves optimal results, indicating that the model is capable of efficiently processing multiple language documents simultaneously and benefits from this ability. Compared with the results in Table 2, multi-task fine-tuning can further improve the performance on dataset of each language, implicitly showing that GraphMLLM can benefit from more fine-tuning data even though they have different languages.

### 5.5 Ablation Studies

We pre-trained GraphMLLM with 2M documents randomly selected from the IIT-CDIP dataset for ablation experiments. The experiments were conducted mainly on the multilingual datasets FUNSD and XFUND, using the language-specific task setting.

**Table 5.** Ablation study on the effect of multi-level page layouts.

| Task | Layout | FUNSD | XFUND | | | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| | | EN | ZH | JA | ES | FR | IT | DE | PT | |
| SER | W | 0.7243 | 0.8995 | 0.7984 | 0.6879 | 0.7362 | 0.7283 | 0.7526 | 0.7377 | 0.7596 |
| | WS | 0.8367 | 0.9067 | 0.8019 | 0.7979 | 0.8333 | 0.8409 | 0.8496 | 0.8245 | 0.8364 |
| | WSR | 0.8403 | 0.9080 | 0.8034 | 0.7954 | 0.8374 | 0.8458 | 0.8481 | 0.8301 | 0.8386 |

**Table 6.** Ablation study on the effect of dual-stream inter-modal interaction.

| Task | Inter-modal | FUNSD | XFUND | | | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| | | EN | ZH | JA | ES | FR | IT | DE | PT | |
| SER | w/o | 0.6996 | 0.8918 | 0.7833 | 0.6424 | 0.7146 | 0.6885 | 0.7009 | 0.7007 | 0.7277 |
| | w/ | 0.8403 | 0.9080 | 0.8034 | 0.7954 | 0.8374 | 0.8458 | 0.8481 | 0.8301 | 0.8386 |

The results in Table 5 show that adding segment-level layout information significantly improves the performance. This is due to the fact that the granularity of word-level layout is too small to capture the high-level layout features. However, the improvement from adding region-level layout is somewhat marginal, likely because the page-level region layout is too coarse to effectively represent structural information about the document. More robust region extraction methods can be considered in the future to obtain more accurate and meaningful regions.

To verify the effect of inter-modal interaction in our dual-stream, we also conducted a simple comparison experiment, results showing in Table 6. We can see without inter-modal interaction, GraphMLLM's performance will be severely degraded, showing that the interaction between different modalities is essential for the performance.

## 6 Conclusion

In this paper, we present GraphMLLM, a multi-modal document pre-training model that can integrate multi-level layout information for document understanding tasks. Following the idea of decoupling text and layout information, GraphMLLM utilizes a dual-stream structure and models multi-level layout features through hierarchical graph attention networks. The text flow can reuse existing pre-trained language model, which can effectively reduce the quantity of dataset required for pre-training. After pre-training with monolingual documents, GraphMLLM can generalize to multilingual downstream tasks by leveraging off-the-shelf multilingual language models, as long as the OCR engine can provide multilingual text and hierarchical layout information. Experimental results on multiple datasets demonstrate the effectiveness and superiority of our proposed approach.

Despite its effectiveness, GraphMLLM still has two main limitations. The first one is the dependence on OCR results, especially on segment-level layouts. Different OCR engines may generate different results, and the OCR quality may affect the model's performance significantly. The second one is the absence of visual features. Document images have rich visual features, which can be integrated into the model in the future to enhance the model's capability to perform end-to-end visual document understanding tasks.

**Acknowledgements** This work has been supported by the National Key Research and Development Program Grant 2020AAA0109700, and the National Natural Science Foundation of China (NSFC) Grant U23B2029.

## References


1. Appalaraju, S., Jasani, B., Kota, B.U., Xie, Y., Manmatha, R.: Docformer: End-to-end transformer for document understanding. In: International Conference on Computer Vision. pp. 973–983 (2021)
2. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. In: International Conference on Learning Representations (2021)
3. Chi, Z., Dong, L., Wei, F., Yang, N., Singhal, S., Wang, W., Song, X., Mao, X.L., Huang, H., Zhou, M.: InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In: 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 3576–3588 (2021)
4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4171–4186 (2019)
5. Gu, Z., Meng, C., Wang, K., Lan, J., Wang, W., Gu, M., Zhang, L.: Xylayoutlm: Towards layout-aware multimodal networks for visually-rich document understanding. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4583–4592 (2022)
6. He, P., Liu, X., Gao, J., Chen, W.: Deberta: decoding-enhanced bert with disentangled attention. In: The 9th International Conference on Learning Representations (2021)
7. Hong, T., Kim, D., Ji, M., Hwang, W., Nam, D., Park, S.: BROS: A pre-trained language model focusing on text and layout for better key information extraction from documents. In: The 36th AAAI Conference on Artificial Intelligence. pp. 10767–10775 (2022)
8. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: Layoutlmv3: Pre-training for document AI with unified text and image masking. In: The 30th ACM International Conference on Multimedia. pp. 4083–4091 (2022)
9. Jaume, G., Ekenel, H.K., Thiran, J.: FUNSD: A dataset for form understanding in noisy scanned documents. In: The 2nd International Workshop on Open Services and Tools for Document Analysis. pp. 1–6 (2019)
10. Lewis, D.D., Agam, G., Argamon, S., Frieder, O., Grossman, D.A., Heard, J.: Building a test collection for complex document information processing. In: The 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 665–666 (2006)
11. Li, C., Bi, B., Yan, M., Wang, W., Huang, S., Huang, F., Si, L.: StructuralLM: Structural pre-training for form understanding. In: The 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 6309–6318 (2021)
12. Li, J., Xu, Y., Lv, T., Cui, L., Zhang, C., Wei, F.: Dit: Self-supervised pre-training for document image transformer. In: The 30th ACM International Conference on Multimedia. pp. 3530–3539 (2022)
13. Li, P., Gu, J., Kuen, J., Morariu, V.I., Zhao, H., Jain, R., Manjunatha, V., Liu, H.: Selfdoc: Self-supervised document representation learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5652–5660 (2021)
14. Li, Y., Qian, Y., Yu, Y., Qin, X., Zhang, C., Liu, Y., Yao, K., Han, J., Liu, J., Ding, E.: Structext: Structured text understanding with multi-modal transformers. In: The 21st ACM Multimedia Conference on Multimedia. pp. 1912–1920 (2021)

15. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
16. Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., Lee, H.: Cord: a consolidated receipt dataset for post-ocr parsing. In: Workshop on Document Intelligence at NeurIPS 2019 (2019)
17. Peng, Q., Pan, Y., Wang, W., Luo, B., Zhang, Z., Huang, Z., Cao, Y., Yin, W., Chen, Y., Zhang, Y., Feng, S., Sun, Y., Tian, H., Wu, H., Wang, H.: ERNIE-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. In: Findings of the Association for Computational Linguistics: EMNLP 2022. pp. 3744–3756 (2022)
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30. pp. 5998–6008 (2017)
19. Wang, J., Jin, L., Ding, K.: LiLT: A simple yet effective language-independent layout transformer for structured document understanding. In: The 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 7747–7757 (2022)
20. Wang, J., Liu, C., Jin, L., Tang, G., Zhang, J., Zhang, S., Wang, Q., Wu, Y., Cai, M.: Towards robust visual information extraction in real world: New dataset and novel solution. In: The AAAI Conference on Artificial Intelligence. pp. 2738–2745 (2021)
21. Wang, W., Huang, Z., Luo, B., Chen, Q., Peng, Q., Pan, Y., Yin, W., Feng, S., Sun, Y., Yu, D., et al.: Ernie-mmlayout: Multi-grained multimodal transformer for document understanding. arXiv preprint arXiv:2209.08569 (2022)
22. Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., Zhou, L.: LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In: The 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 2579–2591 (2021)
23. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: Layoutlm: Pre-training of text and layout for document image understanding. In: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1192–1200 (2020)
24. Xu, Y., Lv, T., Cui, L., Wang, G., Lu, Y., Florencio, D., Zhang, C., Wei, F.: Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. arXiv preprint arXiv:2104.08836 (2021)
25. Zhai, M., Li, Y., Qin, X., Yi, C., Xie, Q., Zhang, C., Yao, K., Wu, Y., Jia, Y.: Fast-structext: An efficient hourglass transformer with modality-guided dynamic token merge for document understanding. arXiv preprint arXiv:2305.11392 (2023)
26. Zhang, Z., Ma, J., Du, J., Wang, L., Zhang, J.: Multimodal pre-training based on graph attention network for document understanding. IEEE Transactions on Multimedia **25**, 6743–6755 (2023)