DocFormerv2: Local Features for Document Understanding

Srikar Appalaraju[1] *, **Peng Tang**[1], **Qi Dong**[1], **Nishant Sankaran**[1], **Yichu Zhou**[2] †, **R. Manmatha**[1]

[1]AWS AI Labs
[2]School of Computing at University of Utah
{srikara, tangpen, qdon, nishsank, manmatha}@amazon.com, flyaway@cs.utah.edu

Abstract

We propose DocFormerv2, a multi-modal transformer for Visual Document Understanding (VDU). The VDU domain entails understanding documents (beyond mere OCR predictions) e.g., extracting information from a form, VQA for documents and other tasks. VDU is challenging as it needs a model to make sense of multiple modalities (visual, language and spatial) to make a prediction. Our approach, termed DocFormerv2 is an encoder-decoder transformer which takes as input - vision, language and spatial features. DocFormerv2 is pre-trained with unsupervised tasks employed asymmetrically i.e., two novel document tasks on encoder and one on the auto-regressive decoder. The unsupervised tasks have been carefully designed to ensure that the pre-training encourages local-feature alignment between multiple modalities. DocFormerv2 when evaluated on *nine* challenging datasets shows state-of-the-art performance on **all** over strong baselines - On TabFact (+4.3%), InfoVQA (+1.4%), FUNSD (+1.0%). Furthermore, to show generalization capabilities, on three VQA tasks involving scene-text, DocFormerv2 outperforms previous comparably-sized models and even does better than much larger models (such as GIT2, PaLI and Flamingo) on these tasks. Extensive ablations show that due to its novel pre-training tasks, DocFormerv2 understands multiple modalities better than prior-art in VDU.

Introduction

Documents have become ubiquitous carriers of information, including forms, tables, invoices, and other structured documents. Many such documents require visual and layout understanding to make sense (just the text string is insufficient). Visual Document Understanding (VDU) is the task of leveraging machine learning techniques to comprehend such scanned documents, such as PDFs or images. Popular VDU tasks include Document and Tables VQA (Mathew et al. 2020; Chen et al. 2019), sequence labeling for key-value identification in forms (Jaume, Ekenel, and Thiran 2019), entity extraction (Seunghyun et al. 2019), and document classification (Harley, Ufkes, and Derpanis 2015). While modern deep-learning based OCR models (Litman et al. 2020) have proven to be effective in extracting text from documents, the naive approach of linearizing the OCR-text and feeding it to a language model is sub-optimal. This is because the content of a document is presented according to a visual layout and structure that must be taken into account for accurate understanding. Naively linearizing the text from left-to-right will result in sub-optimal performance as the semantic meaning alters based on layout, as shown in Figure 1 - Table 5,4 has experiments demonstrating this. Instead, VDU requires a multi-modal approach that can comprehend text and visual features in the context of a document's 2D layout.

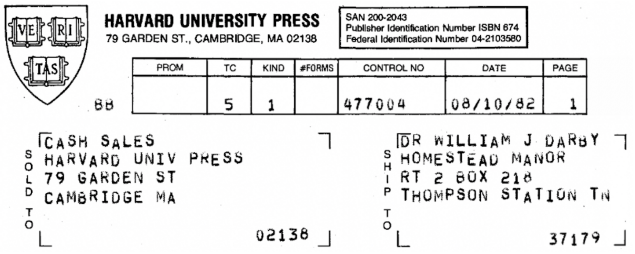

Figure 1: Visual Document Understanding: Snippet of a document receipt from DocVQA (Mathew, Karatzas, and Jawahar 2021). VDU tasks could include a model asked to predict "SOLD TO" address (VQA) or predict all relations ("SOLD TO" → <address>, "SHIP TO" → <address>) or asked to infer info from table (at the top).

Multi-modal training in general entails feature alignment. Specific to vision-language learning this means aligning a piece of text with an arbitrary span of pixels in visual space (Ho et al. 2022; Kim, Son, and Kim 2021; Radford et al. 2021; Wang et al. 2022a; Alayrac et al. 2022; Biten et al. 2022; Appalaraju et al. 2021; Hao et al. 2023; Appalaraju et al. 2020; Li et al. 2022; Chen et al. 2022b). How those features are aligned makes a big difference. In VDU, a majority of the tasks require *local and layout-relative* understanding of the document. For example, in document VQA, semantic labeling or entity extraction, a model needs to make sense of text in-relation to where the text is placed in a document. E.g.: "1" when placed at the top-right/bottom-left of a document is to be interpreted as a page-number vs as a number when placed anywhere else.

Based on this domain understanding of VDU and its

*Corresponding author.
†Work conducted during an internship at Amazon.

Model	Year	Conf.	Arch.	Input Mod.
LayoutLMv1	2020	KDD	E	T + S
DocFormerv1	2021	ICCV	E	T + V + S
LayoutLMv2	2021	ACL	E	T + V + S
SelfDoc	2021	CVPR	E	-
LayoutLMv3	2022	ACM	E	T + V + S
BROS	2022	AAAI	E	T + S
XYLayoutLM	2022	CVPR	E	T + V + S
FormNet	2022	ACL	E	-
ERNIE-Layout	2022	EMNLP	E	T + V + S
LiLT	2022	ACL	E	T + S
XDoc	2022	EMNLP	E	T
TILT	2021	ICDAR	E + D	T + V + S
DocFormerv2	2024	AAAI	E + D	T + V + S

Table 1: VDU Related Work: In this table, a summary of VDU prior art is presented with their architecture (E: Encoder, D: Decoder), the input (T: text, V: vision, S: spatial features), the vision features branch and core idea behind the work.

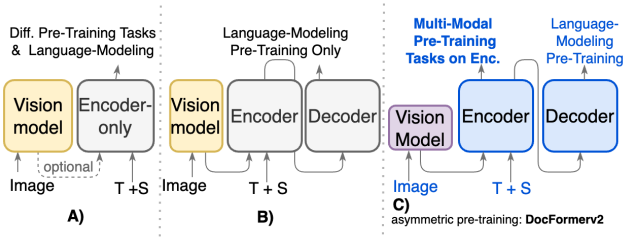

Figure 2: VDU Paradigms: Broad state of Visual Document Understanding (VDU) approaches. In a) E-only LayoutLM (Xu et al. 2020a) and variants. b) E+D but only language-task TILT (Powalski et al. 2021). c) Ours.

challenges, we present DocFormerv2 (DFv2) which is an encoder-decoder multi-modal transformer. In this work, we meticulously devise two novel unsupervised pre-training tasks with the objective of incorporating local semantic information of a document into the model. These pre-training tasks impart the ability to the model to accurately locate relevant information within the document. We also depart from VDU prior-art(Powalski et al. 2021; Tang et al. 2022) as we introduce a novel asymmetrical method of pre-training. i.e., multi-task pre-training on encoder (two tasks) and decoder (one task). We propose two novel pre-training tasks on encoder with the intent to enrich the encoder with local semantic information. The tasks aid in by fusing and aligning multi-modal input and generating efficient representations for the decoder. We show that these pre-training tasks are necessary for effective VDU (see ablation sec.). Furthermore, we demonstrate that a simplified linear visual layer is sufficient to encapsulate visual features, simplifying the architecture from previous VDU research (Xu et al. 2020b; Li et al. 2021c; Powalski et al. 2021) which required specific visual encoders (Dosovitskiy et al. 2020; Liu et al. 2021; He et al. 2016).

Experimentally we demonstrate that DocFormerv2 achieves state-of-the-art performance on five VDU tasks. In addition, we demonstrate the versatility of DocFormerv2 by utilizing its pre-trained model and fine-tuning it on text-VQA tasks from a completely different domain. Our approach yields superior performance on three distinct text-VQA datasets, surpassing comparable models and in some datasets much bigger models like GIT2 (Wang et al. 2022a), PaLI (Chen et al. 2022b) and Flamingo (Alayrac et al. 2022). Therefore, the primary contributions of this paper are as follows:

- Asymmetrical method of pre-training for VDU: Two novel tasks on the encoder which encourage local multi-modal feature collaboration (*Token-to-Line* task and *Token-to-Grid* task) and one on the decoder (see Approach sec).
- Simplified Visual branch: DocFormerv2 is end-to-end trainable and it does not rely on a pre-trained object detection network for visual features simplifying its architecture. On five varied downstream VDU tasks, DocFormerv2 achieves state-of-the-art results (See experiments sec.).
- We also show DocFormerv2 versatility by fine-tuning it on a totally different domain - text-VQA datasets without changing the pre-training. DocFormerv2 beats strong baselines and achieves state-of-the-art numbers on three text-VQA datasets amongst similar model sizes. Selectively, on Text-VQA it out-performs much larger models like PaLI-3B +6.8%, PaLI-15B +1.5% and Flamingo(Alayrac et al. 2022) (+9.9%) (106x DocFormerv2 size in the num. of parameters) by absolute accuracy (see TextVQA experiments).

Furthermore, we conducted comprehensive ablation experiments to demonstrate the advantages of our pre-training tasks, the model's resilience to input noise, and the efficacy of the simplified visual branch.

Related Work

VDU research has attracted considerable attention over the past few years (Wang et al. 2022b; Xu et al. 2020a; Fujinuma et al. 2023; Xu et al. 2020b; Appalaraju et al. 2021; Li et al. 2021c; Powalski et al. 2021; Li et al. 2021b; Huang et al. 2022; Appalaraju et al. 2023; Hong et al. 2020; Gu et al. 2022a; Tang et al. 2023b; Gu et al. 2022b; Lee et al. 2022; Wang, Jin, and Ding 2022; Chen et al. 2022a; Tang et al. 2022; Łukasz Borchmann et al. 2021; Peng et al. 2022; Li et al. 2021a; Tang et al. 2023a). Prominent published research papers in this area are catalogued in Table 1 - the research focus has been lopsided towards encoder-only models. While TILT (Powalski et al. 2021) proposed a encoder-decoder transformer for VDU, they only train it on one pre-training task (masked language modeling) and also use a bulky visual CNN. Our approach DocFormerv2 , simplifies the architecture by not using a separate visual module (CNN or Transformer based) and has multiple unsupervised pre-training tasks. See supplemental for more on prior art.

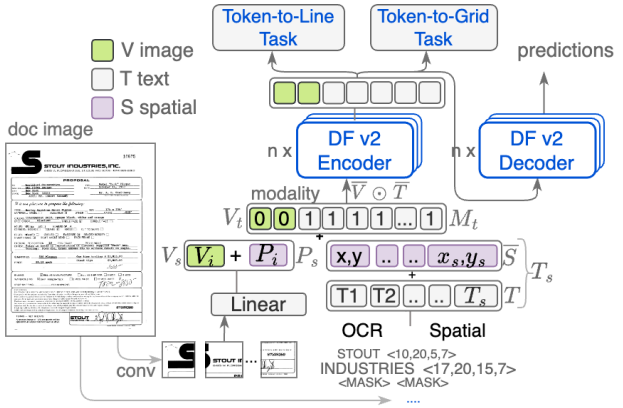

Figure 3: DocFormerv2 Pre-train Architecture: After pre-train, the two prediction heads (token-to-line and grid) on encoder are removed, rest of the architecture remains the same for down-stream tasks. Read section for more details on T_s and V_s. All components are end-to-end trainable. Best viewed in color.

Approach

Architecture

DocFormerv2 (DFv2) is a multi-modal encoder-decoder transformer architecture (see fig. 3). Three variations of DFv2 are designed - small, base and large variants (see supplemental material for details). DFv2 takes multi-modal inputs, the image of the document I, text T extracted by an OCR model along with OCR bounding box co-ordinates as spatial features $\overline{S}$. DFv2 has a unified multi-modal encoder where the multi-modal features fuse and align with the help of novel pre-training tasks (see Fig. 3).

Visual features: DFv2 has a simplified visual branch contrary to most VDU prior-art (fig. 2). DFv2 consumes a flattened image sequence as visual input. Specifically, let $v \in \mathbb{R}^{3\times h\times w}$ be the image of a document. A simple $V = linear(conv_{2\times 2}(v))$ is used to create an image embedding. The weights are randomly initialized for pre-training. As documents tend to have lots of white-space, the linear down-sampling layer gives an opportunity for the model to only keep relevant visual features. Based on our ablation experiments (see supplemental material), this simple approach gives better results than using expensive image encoders such as Swin, ViT (Liu et al. 2021; Dosovitskiy et al. 2020; Ronneberger, Fischer, and Brox 2015) or bulky object-detection networks like FRCNN variants (Ren et al. 2015) as was used in VDU prior-art (Powalski et al. 2021; Appalaraju et al. 2021; Xu et al. 2021). Since transformer layers are permutation-invariant, a learnable 2D-positional encoding P_s is also computed. Finally, $V_s = V + P_s$

Language features: Let t be the predicted text extracted via an OCR model for a document image. DFv2 uses a sentence-piece sub-word tokenizer (Kudo and Richardson 2018) to get tokens t_{tok}. A maximum sequence limit s is applied during training and testing, so if the number of OCR tokens is greater than s, the rest is ignored. If the sequence length is less than s, the sequence is padded. The OCR tokens t_{tok} are sent to a learnable embedding layer W_t to create a text embedding $T = W_t(t_{tok})$.

Spatial features: For each OCR word t_i, the OCR model predicts its bounding-box location in the normalized form $b_i = (x_1, y_1, x_3, y_3)$. This information is encoded using four learnable spatial embedding layers - W_x for encoding a word horizontal spatial information x_i, W_y for the vertical coordinate y_i, W_h for word height h_i and W_w for the width w_i. The spatial features not only encode the location of a word in the document but also provides cues about a word's font-size and thereby its importance in a document (via h_i and w_i). Specifically, spatial features $\overline{S} = W_x(x_1, x_3) + W_y(y_1, y_3) + W_h(y_3 - y_1) + W_w(x_3 - x_1)$. Finally, $T_s = T + \overline{S}$.

Other features: T_s and V_s features are different modalities (fig. 3). As the model has no idea it is being fed multi-modal input, another learnable embedding W_m is used to provide cues to the model about the multi-modal input. A modality-embedding W_m learns nuances of different modalities, which generates M_v embedding for visual modality and M_t for text. Finally, $\overline{V} = V_s + M_v$ and $\overline{T} = T_s + M_t$, whereby $\overline{T}$ and $\overline{V}$ are concatenated ($\overline{V} \odot \overline{T}$) in the sequence dimension to form the input sequence to the DFv2 encoder.

Unsupervised Document Pre-training

In DocFormerv2 we follow the now well established practice of unsupervised pre-training followed by downstream task fine-tuning. Furthermore, with the intent of eliciting the maximum benefit from unsupervised pre-training, we designed the pre-training tasks as a close proxy for downstream tasks. We now describe the two novel pre-training tasks employed on the encoder and the language modeling task on decoder. All three tasks are performed at the same time and the final loss is a linear combination of all three losses for each iteration.

Encoder Token-to-Line Task: We share the intuition that for VDU tasks local feature semantic alignment is important. Most of the related information for key-value prediction in a form or VQA is either on the same line or adjacent lines of a document e.g., see fig. 4, in order to predict the value for `"TOTAL"` (box a), the model has to look in the same line (to its right - `"$4.32"` box d). We teach the model the relative position information between tokens. For implementation, we randomly pick two language tokens and ask the model to predict the number of lines between them. Furthermore, as a document could have an arbitrary number of lines of text, the task is quantized. i.e., there are only three labels: `{0, 1, 2}`. All token pairs that are more than 2 lines apart are labelled as 2 because distant tokens are not likely related and the model should learn to ignore them. Assume that a, b, c, d (fig. 4) are lines. Let F be the DFv2 encoder head function trying to predict a label for this task. then:

$$F(a, d) = 0;\ F(a, b) = 1;\ F(b, c) = 2 \qquad (1)$$

Based on the ablation (table 8), this task gives +2.2% benefit on DocVQA task. The loss for this task is tracked as L_{tol}.

Figure 4: Token-to-Line

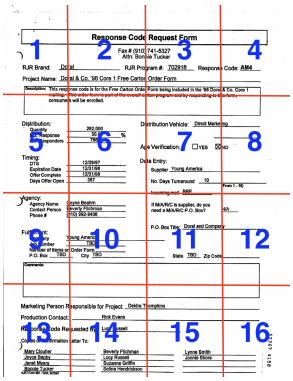

Figure 5: Token-to-Grid: 4x4

Encoder Token-to-Grid Task: Different semantic information is concentrated in different regions of the document. For example, a) In a financial document, the top block contains the header, the middle block contains information to be filled and the bottom block typically contains footer elements/instructions. b) Page numbers are typically at the top or the bottom. c) In a receipt/invoice the company name is typically at the top. The content of a document is presented according to a visual layout and structure that must be taken into account for accurate understanding. Based on this intuition, this task pairs language semantics with the location (visual, spatial or both) in a document. Specifically, the document is virtually divided into a m x n grid. Each OCR token is assigned a grid number and DFv2 is tasked with predicting the grid number for each token. For each OCR token t_i, its top-left location (x_1, y_1) is used to determine its grid-number g_i. Grids are in raster-scan order, so if a particular token falls on the boundary of multiple grids, the scan-order is used to disambiguate. If a token falls on the boundary of normalized image co-ordinates, they are ignored for prediction. See fig. 5 for viz. Specifically, we have:

$$g_i = \lfloor \frac{x_1}{\Delta_x} \rfloor + \lfloor \frac{y_1}{\Delta_y} \rfloor \cdot m,$$

where Δ_x and Δ_y are the widths and heights of each grid, respectively, and m is the number of grids in a row. The loss L_{tog}.

Decoder Language Modeling: Since VDU predictions are in the language domain, language understanding forms an important component of DFv2 pre-training. We do the denoising masked language modeling popularized by T5 (Raffel et al. 2019). During pre-training, not only are the input tokens randomly `MASKED` it's spatial features (mentioned in §) are also masked. Masking the spatial features $\overline{S}$ for the masked tokens makes the grid prediction and line prediction hard because the model does not have 2D-position information of the masked tokens. It has to infer from other available context. The loss for this operation is denoted L_{dlm}.

Final pre-training loss: The final loss is a linear combination of all three pre-training losses i.e., $L_{final} = k*L_{tol}+l*L_{tog}+m*L_{dlm}$, where k, l, m are empirically determined.

Downstream Tasks: Once pre-training is done, we remove the token-to-line and token-to-grid linear prediction heads. The rest of the pre-trained model is fine-tuned on the respective downstream train data.

Experiments

Implementation details: Following prior-art (Appalaraju et al. 2021; Powalski et al. 2021; Biten et al. 2022; Xu et al. 2020a, 2021; Huang et al. 2022) we use the Industrial Document Library (IDL)[1] dataset for pre-training. The IDL is a collection of industry documents hosted by UCSF. It hosts millions of documents publicly disclosed from various industries like tobacco, drug, food etc. The data from the website amounts to about 13M documents, translating to about 70M pages of various document images. We further extracted OCR for each document. Data was cleaned and about 6M documents were pruned, the resulting 64M document images and OCR-text (with spatial co-ordinates) is used for unsupervised pre-training. The data distribution for IDL 64M is presented in supplemental section.

Downstream experiments: The model is fine-tuned on the provided training set and numbers are reported on the corresponding validation/test set. No dataset specific hyperparameter tuning is done. This is an advantage of our approach and we believe that the numbers may be higher if dataset specific fine-tuning is done. Details about fine-tuning datasets are in the supplemental section. We used Pytorch (Paszke et al. 2019) and the Huggingface library (Thomas et al. 2019).

Evaluation Metrics: A dataset specific evaluation metric is adopted. For DocVQA(Mathew et al. 2020), InfoVQA(Mathew et al. 2022), ST-VQA(Biten et al. 2019b), Average Normalized Levenshtein Similarity (ANLS) (Biten et al. 2019a) is used. ANLS measures the similarity between the predicted results and ground truth and ranges from (0,100). For FUNSD(Jaume, Ekenel, and Thiran 2019), CORD(Seunghyun et al. 2019) F1-score is used. For TextVQA (Singh et al. 2019) and OCR-VQA(Mishra et al. 2019) accuracy is used. In all metrics, higher the better.

Table VQA

WikiTable and **TabFact** (Chen et al. 2019; Łukasz Borchmann et al. 2021): These datasets study table understanding and fact verification with semi-structured evidence over tables collected from Wikipedia. Entailed and refuted statements corresponding to a single row or cell were prepared by the authors of TabFact. This task poses challenges due to the complex linguistic and spatial reasoning involved. In Table 2, we can see that DocFormerv2 out-performs prior art by a large margins (+1.1%) and (+4.3%) resp.

Document VQA

DocVQA (Mathew et al. 2020) and InfographicsVQA (Mathew et al. 2022) are datasets for the document VQA task. DocVQA (Mathew et al. 2020) focuses on VQA for real-world industry documents and requires that the model understand images, texts, tables, forms. InfographicsVQA (Mathew et al. 2022) focuses on VQA for infographics and

[1] https://www.industrydocuments.ucsf.edu/

Model	**WikiTable** Acc. (%)	**TabFact** Acc. (%)
methods based on only text / (text + spatial) features:		
$T5_{large}$	33.3	58.9
$T5_{large}$+U	38.1	76.0
$T5_{large}$+2D	30.8	58.0
$T5_{large}$+2D+U	43.3	78.6
methods based on image + text + spatial features:		
$LayoutLMv3_{large}$	45.7	78.1
UDOP	<u>47.2</u>	<u>78.9</u>
$DocFormerv2_{large}$	**48.3**$^{(+1.1\%)}$	**83.2**$^{(+4.3\%)}$

Table 2: Comparison on Table VQA Datasets: DocFormerv2 outperforms the previous state of the art on WikiTableQuestions and TabFact datasets. bold is SOTA and underline indicates the prev SOTA. See supp. for viz. p.s. had to remove citations due to AAAI width and font rules.

requires that the model understand plots/graphs, texts, layout, figures. A model needs to reason multi-modally to generate an answer for this data. Please see the supplemental for data statistics and samples.

Sequence Labeling Task

We study the performance of DocFormerv2 on the semantic entity-labeling task (i.e., group tokens which belong to the same class). We test the model on FUNSD dataset (Jaume, Ekenel, and Thiran 2019), which is a forms dataset containing 199 noisy documents (149 images for train, 50 images for test). There are four classes: *question, answer, header*, and *other*. We measure entity-level performance using F1 score (Table 4). The input sequence to Docformerv2 includes individual texts as prompts and all document texts as context, and the decoder sequence contains the entity texts and predicted labels. Docformerv2 achieves 88.89% F1 score (Table 4), and outperforms the existing methods without using entity box priors in pretraining and finetuning (grayed models in the table).

Following common practice (Łukasz Borchmann et al. 2021; Powalski et al. 2021; Xu et al. 2020b), we train DocFormerv2 on the combination of the training and validation sets and do evaluation on the test set for each dataset. In addition, we also follow (Powalski et al. 2021; Xu et al. 2020b) to train DocFormerv2 on an extra document VQA dataset with 850k question-answer pairs and then fine-tune on DocVQA/InfographicsVQA for higher accuracy.

DocFormerv2 outperforms (Table 3) the previous state of the art for document VQA even without using any extra document VQA pre-training data. After pre-training on the extra data, DocFormerv2 surpasses the previous state of the art by 0.79% on DocVQA and 1.4% on InfographicsVQA, which confirms the effectiveness of our approach.

Model	**DocVQA** test ANLS (%)	**InfoVQA** test ANLS (%)
methods based on only image:		
$Donut_{base}$	67.5	11.5
$Pix2Struct_{large}$	76.6	40.0
methods based on only text / (text + spatial) features:		
$T5_{large}$	70.4	36.7
$T5_{large}$+U	76.3	37.1
$T5_{large}$+2D	69.8	39.2
$T5_{large}$+2D+U	81.0	46.1
methods based on image + text + spatial features:		
$LayoutLMv3_{large}$	83.4	45.1
UDOP	84.7	<u>47.4</u>
$LayoutLMv2^{\dagger}_{large}$	86.7	-
$TILT^{\dagger}_{large}$	<u>87.05</u>	-
$DocFormerv2_{large}$	87.2	-
$DocFormerv2^{\dagger}_{large}$	**87.84**$^{(+0.79\%)}$	**48.8**$^{(+1.4\%)}$

Table 3: Comparison on Document VQA datasets: Our work, DocFormerv2 outperforms the previous state of the art. † indicates training with extra document VQA data.

Entity Extraction Task

We evaluate DocFormerv2 for the entity extraction task on the CORD dataset. CORD (Seunghyun et al. 2019) consists of 1000 receipts (800/100/100 images for train/val/test). It defines 30 fine-grained fields under 4 coarse-grained categories. To extract all entities, in the input sequence, we add a question of "*What are entities of <CLASS>?*" in front of all text context tokens. The output of the decoder includes all entities which are separated by a separator token. Following the standard evaluation metric for entity extraction, we measure entity-level performance using F1 score. Docformerv2 (Table 5) achieves 97.7% F1 score, and outperforms existing methods. Docformerv2 enables multiple entities decoding in an auto-regressive way which shows that the model is able to learn both intra-entity and inter-entity structures. Note that it is unfair to directly compare Docformerv2 with LayoutLMv3(LaMv3), because LaMv3 uses segment-level layout positions, while the other works use word-level layout positions [2]. More importantly, the task studied in Table 5 is entity extraction: predicting words and classes of all entities, against this problem setting if one uses segment-level boxes as inputs.

Generalization Experiments - Scene-Text VQA

In this section, we show the strength of DocFormerv2 on a different task - Scene-Text VQA. Unlike document understanding which focuses on document images, the Scene-Text VQA task answers questions for *natural images* with scene

[2]LaMv3 (Huang et al. 2022) highlighted that using segment-level positions may benefit the semantic entity labeling task, so the two types of work are not directly comparable.

Model	Precision	Recall	F1
methods based on only image:			
$\text{Dessurt}_{\text{base}}$	-	-	65.0
methods based on only text / (text + spatial) features:			
$\text{BERT}_{\text{base}}$	54.69	61.71	60.26
$\text{RoBERTa}_{\text{base}}$	63.49	69.75	66.48
$\text{UniLMv2}_{\text{base}}$	63.49	69.75	66.48
$\text{LayoutLMv1}_{\text{base}}$	76.12	81.55	78.66
$\text{BROS}_{\text{base}}$	80.56	81.88	81.21
$\text{BERT}_{\text{large}}$	61.13	70.85	65.63
$\text{RoBERTa}_{\text{large}}$	67.80	73.91	70.72
$\text{UniLMv2}_{\text{large}}$	67.80	73.91	70.72
$\text{LayoutLMv1}_{\text{large}}$	75.36	80.61	77.89
$\text{StructuralLM}_{\text{large}}$	83.52	86.81	85.14
FormNet	85.21	84.18	84.69
methods based on image + text + spatial features:			
$\text{LayoutLMv1}_{\text{base}}$	76.77	81.95	79.27
$\text{LayoutLMv2}_{\text{base}}$	80.29	85.39	82.76
$\text{LayoutLMv2}_{\text{large}}$	83.24	85.19	84.20
$\text{DocFormer}_{\text{base}}$	80.76	86.09	83.34
$\text{DocFormer}_{\text{large}}$	82.29	86.94	84.55
SelfDoc	-	-	83.36
UDoc	-	-	87.93
StrucTexT✣	85.68	80.97	83.09
$\text{LayoutLMv3}_{\text{base}}$ ❋	77.39	81.65	79.46
$\text{LayoutLMv3}_{\text{large}}$ ❋	81.35	83.75	82.53
$\text{LayoutLMv3}_{\text{base}}$ ❍	89.55	91.65	90.29
$\text{LayoutLMv3}_{\text{large}}$ ❍	92.19	92.10	92.08
UDOP❍	-	-	91.62
$\text{DocFormerv2}_{\text{base}}$	89.15	87.6	88.37
$\text{DocFormerv2}_{\text{large}}$	89.88	87.92	**88.89**(+1.0%)

Table 4: FUNSD comparison: DocFormerv2 does better than models its size and compares well with even larger models. ✣ does not use standard train/test split, and the results are not directly compared with others. ❍ use OCR lines (not word box) as 2D position for words, and use entity boxes as 2D position for each word during fine-tuning and test, and thus the results are not directly comparable. ❋ are results by using the word boxes as 2D position for each word as other competitors do.

Model	Precision	Recall	F1
methods based on only text / (text + spatial) features:			
$\text{BERT}_{\text{base}}$	88.33	91.07	89.68
$\text{UniLMv2}_{\text{base}}$	89.87	91.98	90.92
SPADE	-	-	91.50
$\text{LayoutLMv1}_{\text{base}}$	94.37	95.08	94.72
$\text{BROS}_{\text{base}}$	95.58	95.14	95.36
$\text{BERT}_{\text{large}}$	88.86	91.68	90.25
$\text{RoBERTa}_{\text{large}}$	-	-	93.80
$\text{UniLMv2}_{\text{large}}$	91.23	92.89	92.05
$\text{LayoutLMv1}_{\text{large}}$	94.32	95.54	94.93
FormNet	98.02	96.55	97.28
methods based on image + text + spatial features:			
$\text{LayoutLMv2}_{\text{base}}$	94.53	95.39	94.95
$\text{LayoutLMv2}_{\text{large}}$	95.65	96.37	96.01
$\text{TILT}_{\text{base}}$ ❍	-	-	95.11
$\text{TILT}_{\text{large}}$ ❍	-	-	96.33
$\text{DocFormer}_{\text{base}}$	96.52	96.14	96.33
$\text{DocFormer}_{\text{large}}$	97.25	96.74	96.99
UDoc	-	-	96.86
$\text{LayoutLMv3}_{\text{base}}$ ❋	92.92	94.31	93.61
$\text{LayoutLMv3}_{\text{large}}$ ❋	96.78	96.78	96.78
$\text{LayoutLMv3}_{\text{base}}$ ❍	-	-	96.56
$\text{LayoutLMv3}_{\text{large}}$ ❍	-	-	97.46
UDOP❍	-	-	97.58
$\text{DocFormerv2}_{\text{base}}$	97.51	96.10	96.80
$\text{DocFormerv2}_{\text{large}}$	97.71	97.70	**97.70**(+0.89%)

Table 5: CORD dataset comparison: We present entity-level Precision, Recall, F1 on test set. ❍ use OCR lines (not word box) as 2D position for words, and use entity boxes as 2D position for each word during fine-tuning and testing, and thus the results are not directly comparable. ❋ are results by using the word boxes as 2D position for each word as the other competitors do.

text. We fine-tune our *document* pre-trained models on three Text-VQA datasets. We emphasize that no image-text pre-training was performed on DocFormerv2, it was merely fine-tuned on the respective Text-VQA training dataset. Three popular Text-VQA datasets are used - OCR-VQA (Mishra et al. 2019), TextVQA (Singh et al. 2019) and ST-VQA (Biten et al. 2019b), each with strong baselines from the vision-language community (as is standard practice by Text-VQA we mean any scene text VQA dataset while TextVQA refers to a specific dataset). Please see the supplemental for a dataset breakdown. For OCR-VQA, we fine-tune our models on the training set and do evaluation on the validation and test sets. For TextVQA and ST-VQA, following the previous state-of-the-art methods (Biten et al. 2022; Yang et al. 2021), we fine-tune our models on the combination of the TextVQA and ST-VQA training sets and do evaluation on the validation and test sets of each dataset. Tables 6, 7, 9 show that our large size model outperforms the comparably sized previous state-of-the-art method LaTr (Biten et al. 2022) by +3.4%, +2.4% and +2.2% on the OCR-VQA, TextVQA, and ST-VQA test sets respectively. These results show that our method generalizes beyond document understanding tasks.

Analysis: In an unexpected turn of events, on OCR-VQA, our model $\text{DFv2}_{\text{large}}$ outperforms GIT2, despite the latter being a significantly larger model with 5.1B parameters compared to our 750M, and using a massive 12.9B data corpus for pre-training compared to our 64M.

On TextVQA, DocFormerv2 does better than several vision-language models which are much bigger and have been pre-trained on much more data. On the test set, it is (+9.9%) better than Flamingo (which at 80B has 106x the number of parameters as ours). On the validation set, it is better than PaLI-3B and 15B (+2.2%, +6.8%) respectively. GIT2 and PaLI-17B do perform better than it. (GIT2 also uses 8 VQA datasets to train). DocFormerv2 gets this per-

Model	Val Acc. (%)	Test Acc. (%)
Blk+CNN+W2V	-	48.3
M4C	63.5	63.9
LaAP	63.8	64.1
$\text{LaTr}_{\text{base}}$	67.5	67.9
GIT_{base}	57.3	57.5
$\text{GIT}_{\text{large}}$	62.4	62.9
GIT	67.8	68.1
GIT2✣	-	70.3
$\text{DocFormerv2}_{\text{base}}$	69.7	70.3
$\text{DocFormerv2}_{\text{large}}$	71.1	**71.5**$^{(+3.4\%)}$

Table 6: Comparison on OCR-VQA: DocFormerv2 is better than the previous SOTA by (+3.4%). Bold indicates best and underline indicates the previous state of the art. GIT2 ✣: uses extra VQA data (aggregation of 8 VQA datasets).

formance without any natural image-text pre-training. We present this as evidence that DocFormerv2 is a good approach to solving this problem with a much smaller model and much less data.

Ablation Experiments

Ablation of DFv2 novel pre-training tasks: Table 8 shows DFv2 ablation on the proposed novel pre-training tasks and multi-modal training. The denoising language modeling task and spatial features mentioned in Approach sec. are applied to all architectures. Note, this ablation was performed on DFv2 -small with 1M doc pre-training.

Robustness to OCR errors. This study examines the robustness of DocFormerv2 and LayoutLMv2[3] to OCR errors. Artificial noise simulating character typos is injected into text from the FUNSD dataset, capped at one error per word. While both models utilize visual features, DocFormerv2's generative decoder demonstrates substantial resilience, experiencing only a -1.68% performance drop even with 20% OCR errors, compared to a -9.84% decline for the encoder-only LayoutLMv2. This highlights the advantage of our generative decoder approach in handling text noise. See Fig. 6.

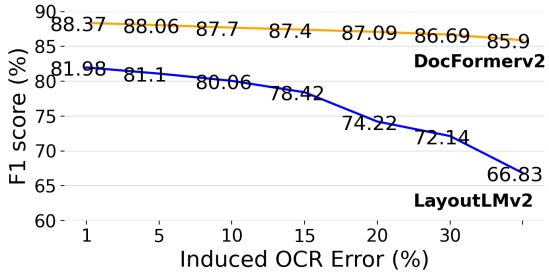

Figure 6: Induced OCR Error Ablation. F1 score perf eval on FUNSD for varying orders of injected OCR errors.

[3]Note that both LayoutLMv2 and ours use visual features, and thus it is fair to compare the robustness in multi-modality context.

Model	Val Acc. (%)	Test Acc. (%)
M4C	47.8	-
LaAP	41.0	41.4
SA-M4C	45.4	44.6
SMA	44.6	45.5
SceneGate	42.4	44.0
SC-Net	44.8	45.7
LOGOS	51.5	51.1
TAP + TAG	53.6	53.7
TAP	54.7	54.0
TAP Two-Stage	55.9	55.3
Flamingo-80B★	57.1	54.1
PreSTU	-	56.3
$\text{LaTr-0.3B}_{\text{base}}$	58.0	58.9
$\text{LaTr-0.3B}^{\dagger}_{\text{base}}$	59.5	59.6
$\text{LaTr-0.85B}^{\dagger}_{\text{large}}$	61.0	61.6
$\text{GIT-0.13B}_{\text{base}}$❍	18.8	-
$\text{GIT-0.4B}_{\text{large}}$❍	37.5	-
GIT-0.7B❍	59.9	59.8
GIT2-5.1B✣	68.4	67.3
PaLI-3B❍	58.8	-
PaLI-15B❍	64.1	-
PaLI-17B❍	70.5	73.1
$\text{DocFormerv2-0.2B}^{\dagger}_{\text{base}}$	61.6	60.0
$\text{DocFormerv2-0.75B}^{\dagger}_{\text{large}}$	**65.6**	**64.0**$^{(+2.4\%)}$

Table 7: Comparison on TextVQA: † indicates the model used the combination of ST-VQA and TextVQA training sets to train the model. GIT2 ✣: extra data used (aggregation of 8 VQA datasets) ★: video-text data. ❍: proprietary image-text data. The Flamingo, GIT2, and PaLI models are much bigger (# parameters $\geq$ 3x $\text{DocFormerv2}_{\text{large}}$ parameters) and use large amounts of external data. $\text{DocFormerv2}_{\text{large}}$ still outperforms Flamingo (+9.9%), PalI-3B (+6.8%) and PalI-15B (+1.5%) models.

Model Ablation	Datasets DocVQA (ANLS)
baseline B	69
B + V	70.5 (+1.5)
B + V + L	71.2 (+2.2)
B + V + G	71.7 (+2.7)
B + V + L + G	73.0 (+4.0)

Table 8: DocFormerv2 Pre-training Tasks Ablation: Impact of three pre-training tasks on four downstream tasks over baseline. B: baseline, V: only with Visual features §, L: with Token-to-Line prediction pre-training §, G: with Token-to-Grid prediction pre-training §.

Pre-training Impact or Better Approach? To isolate the impact of pre-training data size on DFv2's performance, we

Model	Val ANLS (%)	Test ANLS (%)
M4C	47.2	46.2
LaAP	49.7	48.5
SA-M4C	51.2	50.4
SceneGate	52.5	51.6
LOGOS	58.1	57.9
TAP	59.8	59.7
TAP + TAG	62.0	60.2
PreSTU	-	65.5
LaTr-0.3B$_{\text{base}}$	67.5	66.8
LaTr-0.3B$^{\dagger}_{\text{base}}$	68.3	68.4
LaTr-0.85B$^{\dagger}_{\text{large}}$	70.2	69.6
GIT-0.13B$_{\text{base}}$	20.7	-
GIT-0.4B$_{\text{large}}$	44.6	-
GIT-0.7B	69.1	69.6
DocFormerv2-0.2B$^{\dagger}_{\text{base}}$	70.1	68.4
DocFormerv2-0.75B$^{\dagger}_{\text{large}}$	**72.9**	**71.8**$^{(+2.2\%)}$

Table 9: Comparison on ST-VQA: On ST-VQA DocFormerv2 outperforms comparable sized models like GIT and LaTr but large margin (+2.2%) in-spite of being pre-trained on less data. † indicates the combination of the ST-VQA and TextVQA training sets is used.

Model	#data	Datasets FUNSD	CORD
LayoutLMv2$_{\text{base}}$	11M	82.7	94.9
DocFormerv2$_{\text{base}}$	11M	86.1 $^{(+3.4\%)}$	96.2 $^{(+1.3\%)}$
DocFormerv2$_{\text{base}}$	64M	87.9$^{(+5.2\%)}$	96.8$^{(+1.9\%)}$
DocFormerv2$^{\dagger}_{\text{base}}$	64M	88.3 $^{(+5.6\%)}$	96.8 $^{(+1.9\%)}$

Table 10: DocFormerv2 Pre-training Data Ablation: Impact of training with different # of pre-training data on various down-stream tasks. The F1 scores are reported. † indicates the combination of the ST-VQA and TextVQA training sets is used.

conducted an ablation study with both models trained on the same, smaller dataset (11M documents). As shown in Table 10, even with limited data, DFv2 still outperforms LayoutLMv2, indicating the effectiveness of its novel asymmetric pre-training approach. Additionally, DFv2 demonstrates further improvement with more data (64M), solidifying its superiority for VDU tasks.

Correct grid size for Token-to-Grid pre-training? In §, we presented the novel Token-to-Grid pre-training task. In this pre-training ablation §8 this task was observed to provide benefits. Here the appropriate virtual grid-size is empirically determined. From Fig. 7, 4x4 grid seems optimal. Smaller or asymmetric grid structures (4x1) seem to cause harm. On the other end, if the grid is too granular (12x12, 8x8), the performance seems to hurt as well. All models pre-trained on DFv2 $_{\text{small}}$ and 1M documents from IDL, with the

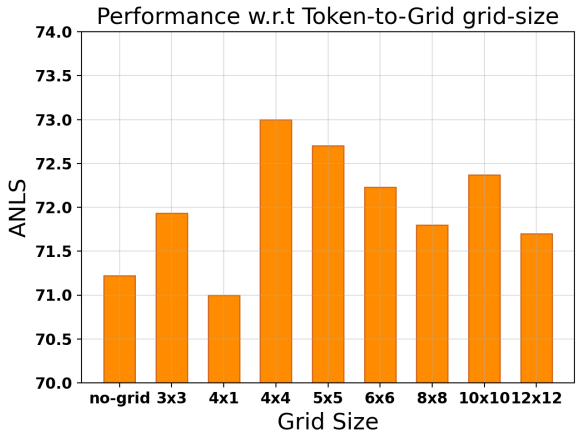

Figure 7: Token-to-Grid Ablation. How different grid sizes used for the Token-to-Grid pre-training task affects model performance on DocVQA. 4x4 seems best and was used for all final pre-training.

Vision and Token-to-line enabled.

More ablations Please find more ablation experiments in supplemental [4] highlighting more experiments of our approach.

Conclusion

Our work DocFormerv2 highlights the importance of two novel pre-training tasks and the efficacy of enriching encoder representations with local semantic information via pre-training tasks. We perform experiments on eight varied datasets (five on VDU and three on scene-text VQA) achieving state-of-the-art numbers on all datasets. Based on ablations, we also show the various design choices and its impact on downstream performance

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; Ring, R.; Rutherford, E.; Cabi, S.; Han, T.; Gong, Z.; Samangooei, S.; Monteiro, M.; Menick, J.; Borgeaud, S.; Brock, A.; Nematzadeh, A.; Sharifzadeh, S.; Binkowski, M.; Barreira, R.; Vinyals, O.; Zisserman, A.; and Simonyan, K. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. *ArXiv*, abs/2204.14198.

Appalaraju, S.; Jasani, B.; Kota, B. U.; Xie, Y.; and Manmatha, R. 2021. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 993–1003.

Appalaraju, S.; Tang, P.; Dong, Q.; Sankaran, N.; Zhou, Y.; and Manmatha, R. 2023. DocFormerv2: Local Features for Document Understanding - Full Paper and Supplemental. *arXiv preprint arXiv:2306.01733*.

Appalaraju, S.; Zhu, Y.; Xie, Y.; and Fehérvári, I. 2020. Towards Good Practices in Self-supervised Representation

[4] https://arxiv.org/abs/2306.01733

Learning. *Neural Information Processing Systems (NeurIPS Self-Supervision Workshop 2020)*.

Biten, A. F.; Litman, R.; Xie, Y.; Appalaraju, S.; and Manmatha, R. 2022. Latr: Layout-aware transformer for scene-text vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16548–16558.

Biten, A. F.; Tito, R.; Mafla, A.; Gomez, L.; Rusinol, M.; Mathew, M.; Jawahar, C.; Valveny, E.; and Karatzas, D. 2019a. Icdar 2019 competition on scene text visual question answering. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1563–1570.

Biten, A. F.; Tito, R.; Mafla, A.; Gomez, L.; Rusinol, M.; Valveny, E.; Jawahar, C.; and Karatzas, D. 2019b. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4291–4301.

Chen, J.; Lv, T.; Cui, L.; Zhang, C.; and Wei, F. 2022a. XDoc: Unified Pre-training for Cross-Format Document Understanding. In *Conference on Empirical Methods in Natural Language Processing*.

Chen, W.; Wang, H.; Chen, J.; Zhang, Y.; Wang, H.; LI, S.; Zhou, X.; and Wang, W. Y. 2019. TabFact: A Large-scale Dataset for Table-based Fact Verification. *ArXiv*, abs/1909.02164.

Chen, X.; Wang, X.; Changpinyo, S.; Piergiovanni, A. J.; Padlewski, P.; Salz, D. M.; Goodman, S.; Grycner, A.; Mustafa, B.; Beyer, L.; Kolesnikov, A.; Puigcerver, J.; Ding, N.; Rong, K.; Akbari, H.; Mishra, G.; Xue, L.; Thapliyal, A. V.; Bradbury, J.; Kuo, W.; Seyedhosseini, M.; Jia, C.; Ayan, B. K.; Riquelme, C.; Steiner, A.; Angelova, A.; Zhai, X.; Houlsby, N.; and Soricut, R. 2022b. PaLI: A Jointly-Scaled Multilingual Language-Image Model. *ArXiv*, abs/2209.06794.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Fujinuma, Y.; Varia, S.; Sankaran, N.; Appalaraju, S.; Min, B.; and Vyas, Y. 2023. A Multi-Modal Multilingual Benchmark for Document Image Classification. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 14361–14376. Singapore: Association for Computational Linguistics.

Gu, J.; Kuen, J.; Morariu, V. I.; Zhao, H.; Barmpalios, N.; Jain, R.; Nenkova, A.; and Sun, T. 2022a. Unified Pre-training Framework for Document Understanding. *ArXiv*, abs/2204.10939.

Gu, Z.; Meng, C.; Wang, K.; Lan, J.; Wang, W.; Gu, M.; and Zhang, L. 2022b. XYLayoutLM: Towards Layout-Aware Multimodal Networks For Visually-Rich Document Understanding. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4573–4582.

Hao, X.; Zhu, Y.; Appalaraju, S.; Zhang, A.; Zhang, W.; Li, B.; and Li, M. 2023. MixGen: A New Multi-Modal Data Augmentation. In *IEEE WACV 2023 - Pre train Workshop*, volume abs/2206.08358.

Harley, A. W.; Ufkes, A.; and Derpanis, K. G. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 991–995.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

Ho, C.-H.; Appalaraju, S.; Jasani, B.; Manmatha, R.; and Vasconcelos, N. 2022. YORO-Lightweight End to End Visual Grounding. In *European Conference on Computer Vision - ECCV CAMP Workshop*.

Hong, T.; Kim, D.; Ji, M.; Hwang, W.; Nam, D.; and Park, S. 2020. BROS: A Pre-trained Language Model for Understanding Texts in Document. *https://openreview.net/forum?id=punMXQEsPr0*.

Huang, Y.; Lv, T.; Cui, L.; Lu, Y.; and Wei, F. 2022. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. *arXiv preprint arXiv:2204.08387*.

Jaume, G.; Ekenel, H. K.; and Thiran, J.-P. 2019. FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, 2: 1–6.

Kim, W.; Son, B.; and Kim, I. 2021. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. In *ICML*.

Kudo, T.; and Richardson, J. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Conference on Empirical Methods in Natural Language Processing*.

Lee, C.-Y.; Li, C.-L.; Dozat, T.; Perot, V.; Su, G.; Hua, N.; Ainslie, J.; Wang, R.; Fujii, Y.; and Pfister, T. 2022. FormNet: Structural Encoding beyond Sequential Modeling in Form Document Information Extraction. In *Annual Meeting of the Association for Computational Linguistics*.

Li, C.; Bi, B.; Yan, M.; Wang, W.; Huang, S.; Huang, F.; and Si, L. 2021a. StructuralLM: Structural Pre-training for Form Understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 6309–6318.

Li, C.; Fehérvári, I.; Zhao, X.; Macêdo, I.; and Appalaraju, S. 2022. SeeTek: Very Large-Scale Open-set Logo Recognition with Text-Aware Metric Learning. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 587–596.

Li, J.; Xu, Y.; Cui, L.; and Wei, F. 2021b. MarkupLM: Pre-training of Text and Markup Language for Visually Rich Document Understanding. In *Annual Meeting of the Association for Computational Linguistics*.

Li, P.; Gu, J.; Kuen, J.; Morariu, V. I.; Zhao, H.; Jain, R.; Manjunatha, V.; and Liu, H. 2021c. SelfDoc: Self-Supervised Document Representation Learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5648–5656.

Litman, R.; Anschel, O.; Tsiper, S.; Litman, R.; Mazor, S.; and Manmatha, R. 2020. SCATTER: selective context attentional scene text recognizer. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11962–11972.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9992–10002.

Mathew, M.; Bagal, V.; Tito, R.; Karatzas, D.; Valveny, E.; and Jawahar, C. 2022. InfographicVQA. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1697–1706.

Mathew, M.; Karatzas, D.; and Jawahar, C. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2200–2209.

Mathew, M.; Karatzas, D.; Manmatha, R.; and Jawahar, C. V. 2020. DocVQA: A Dataset for VQA on Document Images. arXiv:2007.00398.

Mishra, A.; Shekhar, S.; Singh, A. K.; and Chakraborty, A. 2019. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, 947–952. IEEE.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Peng, Q.; Pan, Y.; Wang, W.; Luo, B.; Zhang, Z.; Huang, Z.; Hu, T.; Yin, W.; Chen, Y.; Zhang, Y.; et al. 2022. ERNIE-Layout: Layout Knowledge Enhanced Pre-training for Visually-rich Document Understanding. *arXiv preprint arXiv:2210.06155*.

Powalski, R.; Borchmann, Ł.; Jurkiewicz, D.; Dwojak, T.; Pietruszka, M.; and Pałka, G. 2021. Going full-tilt boogie on document understanding with text-image-layout transformer. In *International Conference on Document Analysis and Recognition*, 732–747.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*.

Raffel, C.; Shazeer, N. M.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *ArXiv*, abs/1910.10683.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39: 1137–1149.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *ArXiv*, abs/1505.04597.

Seunghyun, P.; Seung, S.; Bado, L.; Junyeop, L.; Jaeheung, S.; Minjoon, S.; and Hwalsuk, L. 2019. CORD: A Consolidated Receipt Dataset for Post-OCR Parsing.

Singh, A.; Natarajan, V.; Shah, M.; Jiang, Y.; Chen, X.; Batra, D.; Parikh, D.; and Rohrbach, M. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8317–8326.

Tang, P.; Appalaraju, S.; Manmatha, R.; Xie, Y.; and Mahadevan, V. 2023a. Multiple-Question Multiple-Answer Text-VQA. *arXiv preprint arXiv:2311.08622*.

Tang, P.; Zhu, P.; Li, T.; Appalaraju, S.; Mahadevan, V.; and Manmatha, R. 2023b. DEED: Dynamic Early Exit on Decoder for Accelerating Encoder-Decoder Transformer Models. *arXiv preprint arXiv:2311.08623*.

Tang, Z.; Yang, Z.; Wang, G.; Fang, Y.; Liu, Y.; Zhu, C.; Zeng, M.; Zhang, C.-Y.; and Bansal, M. 2022. Unifying Vision, Text, and Layout for Universal Document Processing. *ArXiv*, abs/2212.02623.

Thomas, W.; Lysandre, D.; Victor, S.; Julien, C.; Clement, D.; Anthony, M.; Pierric, C.; Tim, R.; Rémi, L.; Funtowicz, M.; et al. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wang, J.; Jin, L.; and Ding, K. 2022. LiLT: A Simple yet Effective Language-Independent Layout Transformer for Structured Document Understanding. In *Annual Meeting of the Association for Computational Linguistics*.

Wang, J.; Yang, Z.; Hu, X.; Li, L.; Lin, K.; Gan, Z.; Liu, Z.; Liu, C.; and Wang, L. 2022a. GIT: A Generative Image-to-text Transformer for Vision and Language. *arXiv preprint arXiv:2205.14100*.

Wang, Z.; Zhou, Y.; Wei, W.; Lee, C.-Y.; and Tata, S. 2022b. A Benchmark for Structured Extractions from Complex Documents. *ArXiv*, abs/2211.15421.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020a. 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*, 1192–1200.

Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; Che, W.; et al. 2020b. LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding. *arXiv preprint arXiv:2012.14740*.

Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; Che, W.; et al. 2021. LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2579–2591.

Yang, Z.; Lu, Y.; Wang, J.; Yin, X.; Florencio, D.; Wang, L.; Zhang, C.; Zhang, L.; and Luo, J. 2021. Tap: Text-aware pre-training for text-vqa and text-caption. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8751–8761.

Łukasz Borchmann; Pietruszka, M.; Stanisławek, T.; Jurkiewicz, D.; Turski, M. P.; Szyndler, K.; and Gralinski, F. 2021. DUE: End-to-End Document Understanding Benchmark. In *NeurIPS Datasets and Benchmarks*.