TAP-VL: Text Layout Aware Pretraining for Enriched Vision-Language Models

Jonathan Fhima [*] Technion, Israel
Elad Ben Avraham AWS AI Labs
Oren Nuriel AWS AI Labs
Yair Kittenplon AWS AI Labs
Roy Ganz AWS AI Labs
Aviad Aberdam AWS AI Labs
Ron Litman [†] AWS AI Labs

Abstract

Vision-Language (VL) models have garnered considerable research interest; however, they still face challenges in effectively handling text within images. To address this limitation, researchers have developed two approaches. The first method involves utilizing external Optical Character Recognition (OCR) tools to extract textual information from images and prepend it to the textual inputs. The second strategy is OCR-free and focuses on employing extremely high-resolution images to improve text recognition capabilities. In this paper, we focus on enhancing the first strategy by introducing a novel method, named TAP-VL, which treats OCR information as a distinct modality and seamlessly integrates it into any VL model. TAP-VL employs a lightweight transformer-based OCR module to receive OCR with layout information, compressing it into a short fixed-length sequence which serves as an input for the LLM. To this end, we conduct model-agnostic pretraining of the OCR module on unlabeled documents, followed by its integration into any VL architecture through short fine-tuning. Extensive experiments demonstrate consistent performance improvements when applying TAP-VL to top-performing VL models, across scene-text and document-based benchmarks.

1. Introduction

Large Vision-Language (VL) models have emerged as a key research area in the field of artificial intelligence, leading to significant progress in multimodal reasoning [4, 8, 12, 18, 20, 23–25, 29, 33–35, 38]. Such architectures bridge the gap between visual and textual data by integrating a vision encoder and a Large Language Model (LLM) via a translation module. This module projects the visual encodings into the text embeddings space. As VL models can generate content based on both visual and textual information, they play a pivotal role in a diverse set of applications and tasks, including image captioning (CAPS) [15] and visual question answering (VQA) [5]. While open-source VL models have shown impressive performance across various tasks, many still face challenges when dealing with with OCR-oriented tasks such as TextVQA [50], TextCaps [49], and DocVQA [43]. There are two main strategies to address this challenge: (1) integrating an external OCR system to extract OCR tokens and use them as additional input, and (2) employing very high-resolution images combined with extensive pre-training to enhance text recognition. Each approach has its own advantages and limitations, and both are active areas of research. In this paper, we focus on the first approach.

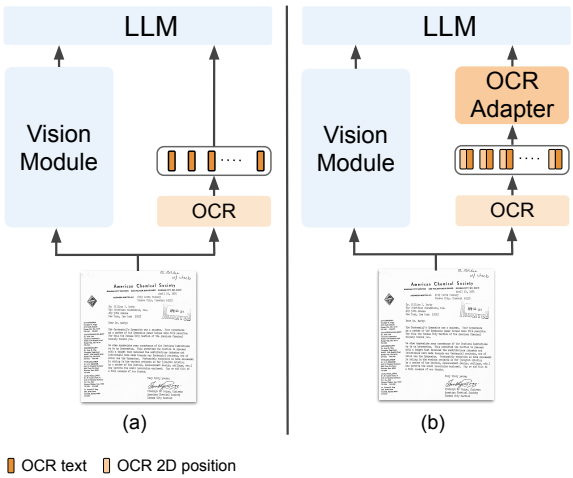


Figure 1. **Layout-aware OCR Adapter**. (a) Previous methods extract OCR data and input it into the LLM as plain text. (b) TAP-VL introduces a plug-and-play OCR adapter that leverages layout information and can be seamlessly integrated with any vision-language LLM (VLLM).

The prevailing paradigm for incorporating OCR into VL systems involves prepending raw OCR-extracted words into

[*]Work done during an Amazon internship
[†]Corresponding author: litmanr@amazon.com

the LLM (left side of Fig. 1). While this strategy enhances performance on OCR-oriented benchmarks, it exhibits critical shortcomings. Firstly, it relies solely on OCR tokens, neglecting crucial spatial layout information proven highly beneficial in OCR-oriented tasks [6, 10, 24, 27]. Moreover, when applied to domains with text-rich images, inserting lengthy OCR sequences into the LLM results in significant computational overhead due to the quadratic complexity of the attention mechanism.

In this study, we address these limitations and introduce TAP-VL, a technique for seamlessly integrating OCR information into any VL model through short fine-tuning (right side of Fig. 1). Conceptually, our method treats OCR as a distinct modality and thus employs an OCR module, similar to the use of a dedicated vision module for encoding visual input. Following this, we introduce a transformer-based lightweight OCR-Q, to generate meaningful representations conditioned on user queries. The OCR encoder captures vital spatial layout information, while the OCR-Q condenses lengthy OCR details into a fixed-size sequence length representation. This condensed representation serves as input for the LLM alongside visual and textual data (right side of Fig. 2). TAP-VL employs these condensed representations to integrate OCR with spatial information into the VL model. By incorporating 2D positional data in addition to the OCR-extracted word tokens, the model can interpret relationships between different textual elements and understand hierarchical structures that are essential for accurate information extraction and holistic document understanding.

Initially, we introduce a standalone, model-agnostic layout-aware pretraining, as depicted on the left side of Fig. 2. This phase operates independently of the VL model, enhancing efficiency and enabling a focused exploration of OCR understanding without introducing a distribution shift to the VL model. Aimed at distilling and extracting the most relevant OCR information, we propose a designated layout-aware pretraining that leverages the abundant unlabeled document data with rich layouts and text [10, 11]. Specifically, we pretrain the OCR-Q in a three-objectives scheme, drawing inspiration from previous works [6, 10, 34]. In more detail, our approach consists of the following layout-aware tasks: (1) OCR-Grounded Mask Denoising, which predicts masked spans based on the noisy OCR input; (2) OCR-Mask Contrastive Learning, which aims to align OCR and word representations within the same document while distinguishing between representations from different documents, and (3) OCR-Mask Matching,which aligns noisy OCR text with missing spans. Combining such objectives propels the model to acquire a deep layout and OCR understanding while providing a compact representation.

Following this, we integrate the same pretrained model into various leading VL models via a short multi-task fine-tuning procedure. Specifically, we examine prominent VL models such as InstructBLIP [20], LLaVA [38] and Qwen-VL [8]. Our extensive experimentation demonstrates the efficacy of TAP-VL across document understanding and scene-text VL benchmarks, resulting in substantial enhancements compared to diverse baseline methods across all assessed benchmarks, including a zero-shot scenario.

In addition, we propose TAP-VL$_{\text{Light}}$, a light-weight version of TAP-VL that solely utilizes our compressed OCR representations without providing the LLM with the raw OCR tokens. This approach is specifically efficient in tasks related to document understanding with dense-text images. Notably, we demonstrate that applying TAP-VL$_{\text{Light}}$ not only significantly reduces the computational costs compared to the relevant baselines (reduces the FLOPs by up to a factor of seven), but also leads to substantial performance improvements. Notably, we showcase TAP-VL$_{\text{Light}}$'s ability to extrapolate to multi-page scenarios without any specific multi-page training. In the most challenging case of multi-page document understanding, TAP-VL$_{\text{Light}}$ achieves performance improvements of up to 4.8%, while substantially reducing the computational costs.

In summary, our contributions include:

- Introducing TAP-VL, a novel approach for seamlessly integrating OCR information into any pretrained VL model, enabling effective reasoning over both textual and spatial information.
- Proposing a unique layout-aware model-agnostic pretraining strategy, utilizing unlabeled document data to acquire rich, condensed OCR features.
- Demonstrating the effectiveness of our method in enhancing performance across various state-of-the-art VL architectures, showcasing its ability to elevate performance across multiple benchmarks in scene-text and document understanding tasks, including challenging zero-shot multi-page setting.
- We present TAP-VL$_{\text{Light}}$, a lightweight version of TAP-VL, capable of handling multi-page documents without any specific training. TAP-VL$_{\text{Light}}$ decreases FLOPs by up to four times while still achieving superior performance compared to approaches relying on the uncompressed OCR sequence.

2. Related work

2.1. Vision-Language Models

VL models have undergone significant evolution, transitioning from task-specific to more generalized approaches, facilitated by LLMs [4, 16–18, 25, 47]. These modern models demonstrate adaptability across diverse tasks and impressive generalization capabilities [18, 20, 34, 38]. Architecturally, these models typically involve three fundamental parts. First, a vision architecture extracts mean-

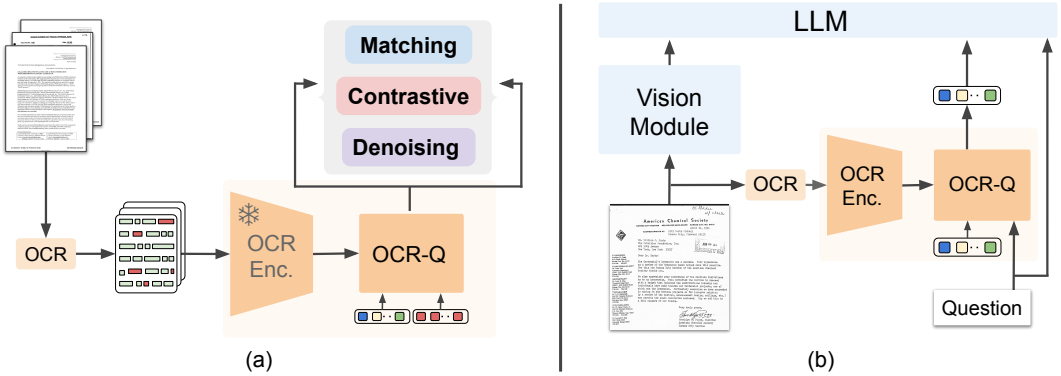


Figure 2. **TAP-VL Approach**. (a) Our model-agnostic layout-aware pretraining framework for creating condensed rich OCR embeddings conditioned on text. (b) TAP-VL fully integrated, enhancing any VL model on OCR-oriented tasks.

ingful information from images, often employing a frozen vision-transformer as the vision encoder. Second, a translation module bridges the gap between vision and language, transforming visual features into a representation comprehensible and processable by a language model. This module may consist of a simple linear layer or MLP [38], or a cross-attention-based transformer architecture [7, 20, 34]. Lastly, the projected visual information and textual instructions, typically in the form of questions or prompts, are input into an LLM to execute the task. More specifically, BLIP-2[34] suggest incorporating a Querying-Transformer (Q-Former) to efficiently add visual cues from a vision encoder to an LLM and InstructBLIP [20] adapts it to follow instructions. LLaVA [38, 39] introduce a simple projection layer to translate between the vision modality to the language one and utilize GPT-4 generated multi-modal data. Qwen-VL [8] propose a single cross-attention layer and perform a full multi-modal fine-tuning phase to align modalities. Additionally, other works have explored extending these methodologies to encompass multiple modalities, such as audio and video [13, 47, 53, 56].

2.2. Integrating OCR information into VL Models

Before the emergence of large vision language models, numerous methods attempted to address OCR-oriented vision challenges. TAP [55] introduced a pretraining objective to align different representations better. In [24], OCR information was integrated into the decoder of an encoder-decoder framework through auxiliary losses. LaTr [10] proposed a layout-aware transformer, enabling reasoning over textual and layout cues using an unsupervised pretraining. However, these works do not consider recent advances in VL models [8, 20] and cannot fully leverage their capabilities.

An alternative approach could target this without assuming an OCR extraction system as the previous approaches do. In this scenario, the VL model would solely rely on visual cues. For instance, OCR-free methods like Qwen-VL and LLaVAR [8, 57] observed that open-source all-purpose vision encoders perform poorly in this regard and proposed a training paradigm comprising explicit OCR-oriented tasks. Although performance improved, they still do not surpass methods utilizing OCR systems. Hence, most VL methods [18, 20] incorporate these OCR systems.

The conventional method of integrating OCR information into these models involves using the raw text extracted by OCR as part of the input prompt to the LLM [18, 20, 38]. This is feasible as modern VL models can understand that these words are associated with the image, requiring minimal modifications and training to seamlessly integrate OCR-derived text. However, this approach overlooks the fact that OCR extraction also provides word bounding boxes, with layout information being crucial [6, 10, 24]. Additionally, inputting the entire OCR sequence into an LLM is computationally intensive, particularly with OCR-dense images, such as in document understanding tasks, potentially leading to poor performance. Therefore, we introduce TAP-VL, a method that effectively and efficiently incorporates OCR and layout information into any VL model.

3. Method

In this section we introduce TAP-VL, which encompasses an OCR module, and a novel layout-aware pretraining paradigm, empowering the VL model with OCR comprehension. The OCR module's architectural design is formulated by treating OCR as an extra, independent modality, addressing its inherent complexity. The layout-aware pretraining aims to teach the OCR module to produce a concise yet rich representation of the OCR. Importantly, this phase operates independently of the VL model, enhancing efficiency and ensuring compatibility with various VL architectures. Following layout-aware pretraining, we integrate our OCR module into any VL architecture through parameter-efficient fine-tuning. This integration leads to significant improvements in OCR benchmarks. Moreover, we propose an additional design choice that can reduce the computational footprint significantly compared to existing approaches, while maintaining comparable performance advantages. In the following sections, we detail our proposed OCR module, the layout-aware model-agnostic pretraining,

and the integration into any VL model.

3.1. Model Architecture

To enhance the OCR understanding of VL architectures, we propose an OCR module composed of two pivotal components: an OCR encoder and an OCR compressor, which we term OCR-Q. The OCR encoder produces embeddings based on tokens and their 2D positions, encompassing essential layout information. The OCR-Q is a transformer-based module designed to produce a compact representation of the OCR based on the query. It is comprised of two transformer sub-modules which share the same self-attention layers: (i) an OCR transformer module interacting with the encoded OCR embeddings and (ii) a text transformer module, which handles the free-text input (such as a user's question). Specifically, the OCR-Q transforms the OCR embeddings via learnable queries and textual prompt into a fixed number of representations. We define K to be the number of learnable queries used as input to the OCR-Q. Upon integration into a VL model, these compressed representations are concatenated with user instructions and fed into the VL model, as illustrated on the right side of Fig. 2.

3.2. Layout-Aware pretraining

The layout-aware pretraining phase aims to produce an OCR-Q capable of compressing OCR content while extracting meaningful information based on textual input. Inspired by the pretraining of BLIP2 [34], we adopt a three-objective scheme: OCR-Grounded Mask Denoising, OCR-Mask Contrastive Learning, and OCR-Mask Matching. To this end, we utilize the myriad of publicly available documents and their OCR information. The overall scheme is illustrated on the left side of Fig. 2.

This approach leverages publicly available unlabeled document corpora by randomly masking spans of OCR, which include both text and layout information. Thus creating pairs consisting of masked versions of the OCR and their corresponding masked words for use during pretraining. For each of the pretraining objectives we follow the same general outline (Figs. 3 to 5): the OCR encoder produces rich embeddings given the masked OCR. These embeddings are then fed to the OCR transformer module via the cross-attention layers, resulting in a fixed number of representations per document (determined by K). The text transformer module receives the masked words as input and a special token, dependent on the pretraining task. The spatial information of these tokens are omitted as the text transformer module operates on free-form unstructured text. Its output is then fed into a task-specific projection layer, preceding the loss. The interaction between the text and OCR transformer modules is governed by the masking mechanism employed, contingent upon the characteristics of the chosen pretraining objective. Next we elaborate on

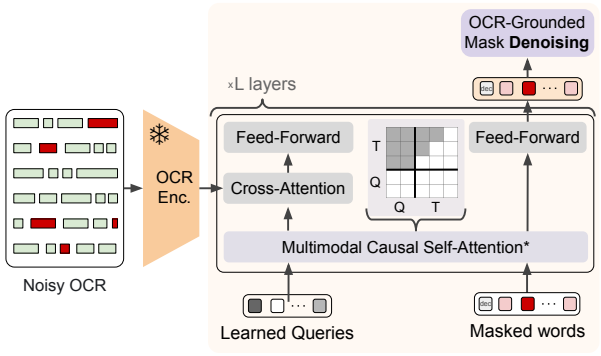


Figure 3. **OCR-Grounded Mask Denoising**. Denoising pretraining mechanism where the OCR-Q predicts the masked words, enhancing comprehension of unmasked text semantics and layout information.

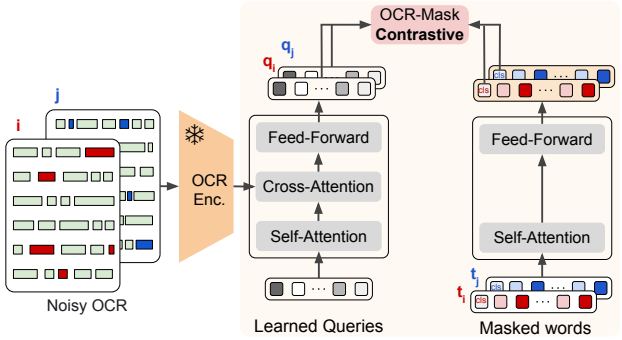


Figure 4. **OCR-Mask Contrastive Learning**. Contrastive learning task aligning the learnable queries representation (interacting with the masked OCR) with the ones of the masked words.

each pretraining task, for more details please refer to Appendix A.5.

OCR-Grounded Mask Denoising tasks the OCR-Q with restoring masked words from noisy OCR input, as illustrated in Fig. 3. This encourages meaningful compressed representations, leveraging both textual and layout information. Throughout this task, the text transformer module indirectly queries the noisy OCR inputs, via the intermediate learned query representations. Since the text transformer module lacks direct access to OCR content, minimizing this loss is feasible only if the representations corresponding to learnable queries are enriched with the relevant OCR information. We use a multimodal mask in our self-attention layers [22, 34] to access compressed OCR information (Fig. 3). This mask restricts learnable queries from attending to text tokens but allows interaction among themselves, while text tokens can attend to learnable queries but only self-interact through a causal mask. A special `<dec>` token is prepended to the masked words as the denoising task prefix.

OCR-Mask Contrastive Learning aims to align the outputs of the OCR transformer module with the text transformer module of the OCR-Q (Fig. 4). The OCR transformer module has access to the noised OCR content, while

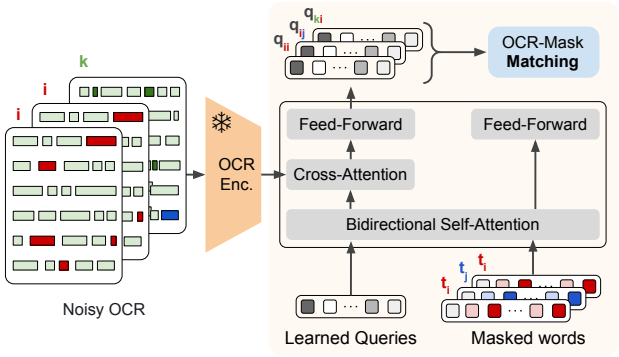


Figure 5. **OCR-Mask Matching**. Visualization of the matching phase, aimed at enabling the OCR-Q to determine the correspondence between masked OCR content and the masked words.

the text transformer module receives a `<cls>` special token followed by the masked words. The primary purpose of this alignment is to enhance the mutual understanding between OCR-encoded information and textual representations. In this task, we use a unimodal mask where the learnable queries and the masked words can only attend to themselves.

OCR-Mask Matching task involves a binary classification objective of matching the representations of the compressed noisy OCR information and the ones of the masked words (Fig. 5). Specifically, for a given noisy document, we couple it with its corresponding masked words with probability p and with a non-matching hard negative with probability $1 - p$. Notably, for this objective, we apply a bidirectional self-attention mask allowing all queries and masked words to attend to each other.

3.3. Incorporating OCR QFormer in VL Models

Following our model agnostic pretraining, we align our OCR module with any VL model in a two-phase fine-tuning procedure. First, we employ an *OCR-to-language alignment*, in which we integrate the OCR encoder and OCR-Q to a frozen LLM. In this stage, we train the OCR module to fit the LLM via fine-tuning using OCR-centric VQA datasets. We use these datasets as most answers can be inferred solely from the OCR information and do not require direct access to the visual inputs [10, 24]. Next, we conduct an *OCR-vision-to-language alignment* in which we consider the entire VL model, as depicted in Fig. 2. In this setting, the LLM is fed with textual instructions along with both visual and OCR features, from the OCR and vision modules. In addition to the textual instructions, appending the raw OCR word list, as commonly done in VL works, is an optional design choice. To present the trade-off between the two options we introduce TAP-VL$_{\text{Light}}$, which omits the raw OCR word list as input to the LLM, making it more computationally efficient than TAP-VL (see Sec. 4.2). In this stage, we conduct a multi-task fine-tuning using a mixture of OCR-oriented and non-OCR-oriented VQA and captioning datasets, including both documents and natural images, to align the system's building block. Specifically, we train the OCR components and employ low-rank adaptation to the LLM [28] while keeping the visual module frozen. The results in a VL system capable of effectively reasoning over both visual and OCR information.

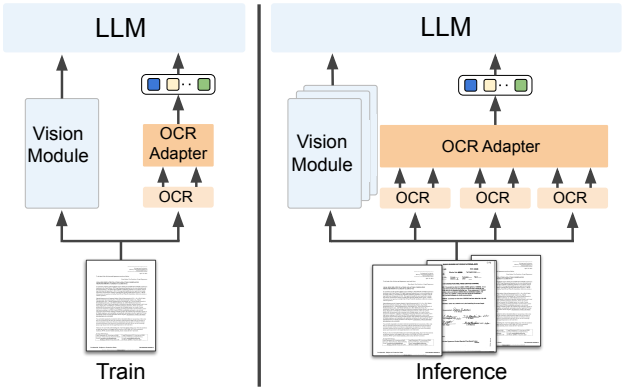


Figure 6. **Extending TAP-VL to multi-page documents**. During inference, the OCR adapter processes all the pages together, while the vision module encodes each page independently. Subsequently, we concatenate the results into a single output, which is then fed into the LLM.

4. Experiments

In this section, we demonstrate the advantages of employing TAP-VL to seamlessly integrate OCR information into state-of-the-art VL models in both scene-text and document understanding benchmarks. Initially, we assess various state-of-the-art VL methods, comparing their performance with and without TAP-VL across diverse tasks and domains, including a zero-shot scenario. Subsequently, we highlight the efficacy of incorporating TAP-VL$_{\text{Light}}$, which creates a rich condensed representation of the OCR, addressing the challenge of long OCR sequences, as in multi-page document question answering.

4.1. Experimental Setting

For all experiments, we use the open-source versions of the VL architectures and initialize from their weights respectively. Unless mentioned otherwise, the OCR encoder is a based on a T5 large encoder [48] with 2D layout embedding initially pretrained in a similar fashion to DocFormerv2 [6]. We initialize the OCR-Q from the pretrained weights of BERT [21], the cross attention weights are trained from scratch. To align the OCR-Q with the OCR encoder, we conduct our pretraining protocol on the expansive unlabeled document dataset IDL [11]. The models are fine-tuned on a variety of domains and tasks simultaneously: document question answering, scene-text visual question answering, general visual question answering, scene-text captioning

	Method	LLM	Scene-Text			Document		0-shot	Average	
			TextVQA [50]	STVQA [9]	TextCaps [49]	DocVQA [43]	InfoVQA [44]	DUDE [31]	Scene-Text	Document
			VQAScore	ANLS	CIDEr	ANLS	ANLS	ANLS		
Specialist	UniTNT [24]	-	55.4	66.0	109.0	-	-	-	-	-
	DocFormer v2 [6]	T5 large [48]	64.0	71.8	-	87.8	48.8	-	-	-
	GIT2 [52]	-	-	75.8	145.0	-	-	-	-	-
	PALI 17B [18]	mT5-XXL [54]	71.8	79.9	160.4	-	-	-	-	-
	PALI-X 55B [16]	-	80.8	84.5	163.7	86.8	54.8	-	-	-
	PALI 3 [17]	UL2 [51]	78.3	85.7	164.3	88.6	62.4	-	-	-
Generalist	InstructBlip [20]	Flan-T5-XL [19]	64.0	63.9	139.9	77.2	43.6	36.3	89.3	60.4
	+ TAP-VL		67.3	66.3	145.5	85.5	51.6	40.9	93.0	68.6
	Δ		+3.3	+2.4	+5.6	+8.3	+8.0	+4.6	+3.7	+8.2
	InstructBlip [20]	Flan-T5-XXL [19]	66.8	65.0	143.1	81.1	49.4	40.0	91.6	65.3
	+ TAP-VL		69.5	67.1	146.9	85.9	54.2	42.9	94.5	70.1
	Δ		+2.7	+2.1	+3.8	+4.8	+4.8	+2.9	+2.9	+4.8
	LLaVA-1.6[39]	Mistral-7B [30]	70.6	71.4	130.0	81.2	46.7	37.9	90.6	64.0
	+ TAP-VL		72.8	72.8	142.2	86.7	54.3	41.4	95.9	70.5
	Δ		+2.2	+1.4	+12.2	+5.5	+7.6	+3.5	+5.3	+6.5
	Qwen-VL [8]	Qwen-7B [7]	75.6	77.6	141.1	80.5	40.4	34.0	98.1	60.5
	+ TAP-VL		76.4	78.8	141.3	85.1	47.8	36.7	98.8	66.5
	Δ		+0.8	+1.2	+0.2	+4.6	+7.4	+2.7	+0.7	+6.0

Table 1. **TAP-VL Results**. Quantitative comparison of TAP-VL integrated into InstructBLIP, LLaVA and Qwen-VL of different sizes, and the corresponding baseline. Evaluation covers scene-text, document understanding benchmarks and 0-shot capabilities, with average performance per domain. TAP-VL consistently outperforms the baselines, demonstrating its effectiveness and versatility

and image captioning .

4.2. Integrating TAP-VL into leading VL methods

In Tab. 1 we report the results of various VL models on several scene-text and document understanding tasks. In the scene-text domain results are provided for TextVQA and STVQA for question answering and TextCaps for captioning. In the document understanding domain, we utilize DocVQA and InfoVQA. Additionally, we provide zero-shot results on the multipage document understanding dataset DUDE [31]. First, we list the specialist models, which were fine-tuned on each dataset independently. These models range from a few hundred million parameters [24] to 55B billion parameters [16] and are pretrained on different datasets. As such, we report these number only for reference as they are not necessarily comparable. In the lower section of the table, we present a comparative analysis between our method and baseline approaches, which were trained under identical settings as our method. This comparison includes performance gaps to underscore the benefits achieved. Specifically, we evaluate our approach within the context of VL models such as InstructBLIP [20], LLaVA [39], and Qwen-VL [8], integrating TAP-VL into each of them. For the InstructBLIP baseline, we consider both the Flan-T5-XL and Flan-T5-XXL architectures [19].

The outcomes of our analyses demonstrate the superior performance of our method compared to the baselines across various architectures and benchmarks. For instance, when integrated with LLaVA, TAP-VL showcases enhancements in TextVQA, DocVQA, and InfoVQA, with notable improvements of **+2.2%**, **+5.5%**, and **+7.6%**, respectively. Furthermore, we calculate the average scores over scene-text and document-based datasets, revealing consistent benefits even when compared to the top-performing method Qwen-VL. Furthermore, we direct the reader to results on non-OCR related benchmarks in Tab. 5, demonstrating comparable or even slightly improved performance compared to the baseline.

Method	TFLOPs (↓)	Datasets			Average
		DocVQA (↑)	InfoVQA (↑)	DUDE (↑)	Document
InstructBlip XL	3.5	77.2	43.6	36.3	60.4
+TAP-VL	4.4	85.5 (+8.3)	51.6 (+8.0)	40.9 (+4.6)	68.6
+TAP-VL$_{\text{Light}}$	**1.3**	85.4 (+8.2)	50.4 (+6.8)	40.1 (+3.8)	67.9
InstructBlip XXL	12.3	81.1	49.4	40.0	65.3
+TAP-VL	13.4	85.9 (+4.8)	54.2 (+4.8)	42.9 (+2.9)	70.1
+TAP-VL$_{\text{Light}}$	**1.8**	84.3 (+3.2)	51.7 (+2.3)	41.0 (+1.0)	68.0
LLaVA-1.6	20.3	81.2	46.7	37.9	64.0
+TAP-VL	21.7	86.7 (+5.5)	54.3 (+7.6)	41.4 (+3.5)	70.5
+TAP-VL$_{\text{Light}}$	**5.5**	84.4 (+3.2)	51.5 (+4.8)	40.8 (+2.9)	68.0
Qwen-VL	19.8	80.5	40.4	34.0	60.5
+TAP-VL	21.0	85.1 (+4.6)	47.8 (+7.4)	36.7 (+2.7)	66.5
+TAP-VL$_{\text{Light}}$	**5.4**	85.1 (+4.6)	48.2 (+7.8)	38.8 (+4.8)	66.6

Table 2. **TAP-VL$_{\text{Light}}$ Results**. Outcomes of integrating TAP-VL$_{\text{Light}}$ into the base model, with TAP-VL results for comparison. Alongside performance metrics, we provide the TFLOPs. Our compression technique reduces computational costs while maintaining improved performance.

In the realm of document analysis, significant improvements are observed, with average performance gaps reaching up to **8.2%**. While TAP-VL demonstrates improvements in both scene-text and document datasets, it can be seen that in the document datasets the improvements are larger. This suggests that in the documents domain, a dedicated OCR encoder has a pivotal role. This aligns with the typically dense textual content and intricate structures found in documents. Consequently, a dedicated OCR encoder, equipped with spatial information as input, possesses the capability to provide more comprehensive answers compared to the VL model alone, which is confronted with raw, unstructured text. Therefore, the effective integration of

this essential component through our method equips the VL model with the supplementary information required to address the tasks proficiently.

To evaluate the performance in a zero-shot setting, we opt for the multi-page DocVQA scenario. Particularly, we combine the OCR from each page to create a single long OCR sequence representing the entire document. The visual encoding is applied individually to each page and subsequently concatenated before being fed into the VL model's LLM (see Figure 6). As demonstrated in the Tab. 1, our method yields significant improvements in the multi-page zero-shot scenario (up to **4.6%** improvement).

After presenting the quantitative impact of using TAP-VL, we now introduce qualitative findings. In Figure 7, we display how our method integrates into LLaVA-1.6. The top row shows examples from scene-text benchmarks, while the bottom row features document-related benchmarks. Our method notably enhances the VL model's OCR and layout comprehension, leading to improved performance. For instance, in the second top-row example, the base model struggles to identify the "*second from the bottom*" book, whereas TAP-VL effectively uses layout information to understand it. Similarly, in the bottom row, our model shows significant improvements in comprehending complex structures like tables.

Leveraging TAP-VL for efficiency In this section we present results on TAP-VL$_{\text{Light}}$, a light-weight version of TAP-VL that improves OCR-oriented tasks while reducing computational load. Processing extended sequences demands substantial computational resources, this is especially significant in transformers which are composed of attention blocks that require quadratic complexity computation. TAP-VL$_{\text{Light}}$ utilizes the condensed representation alone, omitting raw text OCR input. This is particularly crucial for documents, which consist of dense-text images. In TAP-VL$_{\text{Light}}$, the sequence length of the input to the LLM remains unaffected by the length of the OCR, and is bounded to K tokens. Tab. 2 shows TAP-VL$_{\text{Light}}$ results on document understanding tasks.

Our method improves performance across all benchmarks while offering computational advantages. The DUDE benchmark, known for its multi-page document VQA datasets containing sequences that extend up to 10K tokens, poses a significant challenge to VL systems, pushing them to their operational limits. For example, when examining the integration of TAP-VL into LLaVA, our compressed OCR version yields improvements of **+3.2%**, **+4.8%**, and **+2.9%** on DocVQA, InfoVQA, and DUDE, respectively, while reducing TFLOPs from 20.3 to 5.5. Moreover, in the case of Qwen-VL, the benefits of inputting the raw OCR sequence into the LLM are negligible.

Method	TAP-VL's Components				Average	
	Layout information	OCR-Module	Pretraining	OCR-to-language alignment	Scene-Text	Documents
InstructBlip	✗	✗	✗	✗	89.3	60.4
	2D-embeddings	✗	✗	✗	90.0	60.0
		✓	✗	✗	87.6	61.2
		✓	✗	✓	89.1	60.5
	OCR-encoder	✓	✓	✗	89.0	60.3
TAP-VL		✓	✓	✓	92.0	65.7

Table 3. **Analysis of TAP-VL's Components through Ablation Studies**. We examine the impact of each component of TAP-VL. Interestingly, only when OCR-Q is combined with our layout-aware pretraining and OCR-to-language alignment, our method yields consistent performance improvements.

5. Ablation studies

This section analyzes performance improvements from our proposed architecture and pretraining framework. We examine the effects of TAP-VL's individual components, pretraining objectives, and data quantity on the InstructBlip (Flan-T5-XL) architecture with T5 base OCR encoder.

TAP-VL components: In Tab. 3, we incrementally incorporate TAP-VL's components to assess their individual effects, starting from the InstructBlip baseline. Initially, we explore a naive method to integrate layout information into OCR by adding a 2D embedding for each OCR token derived from its spatial location via a designated spatial embedding layer in a residual manner. This methods perform similarly to the baseline, suggesting incomplete utilization of residual or encoded information. Subsequently, integrating our OCR module without additional pretraining improves document results by 0.8% but decreases scene-text results by 1.7%. Further, we analyze the impact of our pretraining and OCR-to-language alignment step. While applying each separately leads to improvements in scene-text and degradation in documents, applying both steps results in consistent enhancement in both domains.

Pretraining Objective: We conduct an ablation study focusing on different configurations of our pretraining tasks. We incrementally incorporate the three pretraining objectives: OCR-Grounded Mask Denoising, OCR Mask Contrastive and OCR-Grounded Mask Matching. As indicated in Tab. 4, each pretraining task contributes to enhancing the overall effectiveness of the final model.

Scale of pretraining Data: We examine the impact of varying data volumes during pretraining, as shown in Tab. 4. The results reveal a correlation between data volume and model performance on both scene-text and document benchmarks. This relationship is particularly notable in the case of document benchmarks, where results exhibit an upward trend with increased pretraining length, rising from 62.8 with 2M pretraining samples to 65.7 with 31M training samples. While the correlation persists in scene-text benchmarks, we note a decline in performance when pretraining samples increase from 13M to 22M. However, the best result is achieved with the maximum pretraining sam-

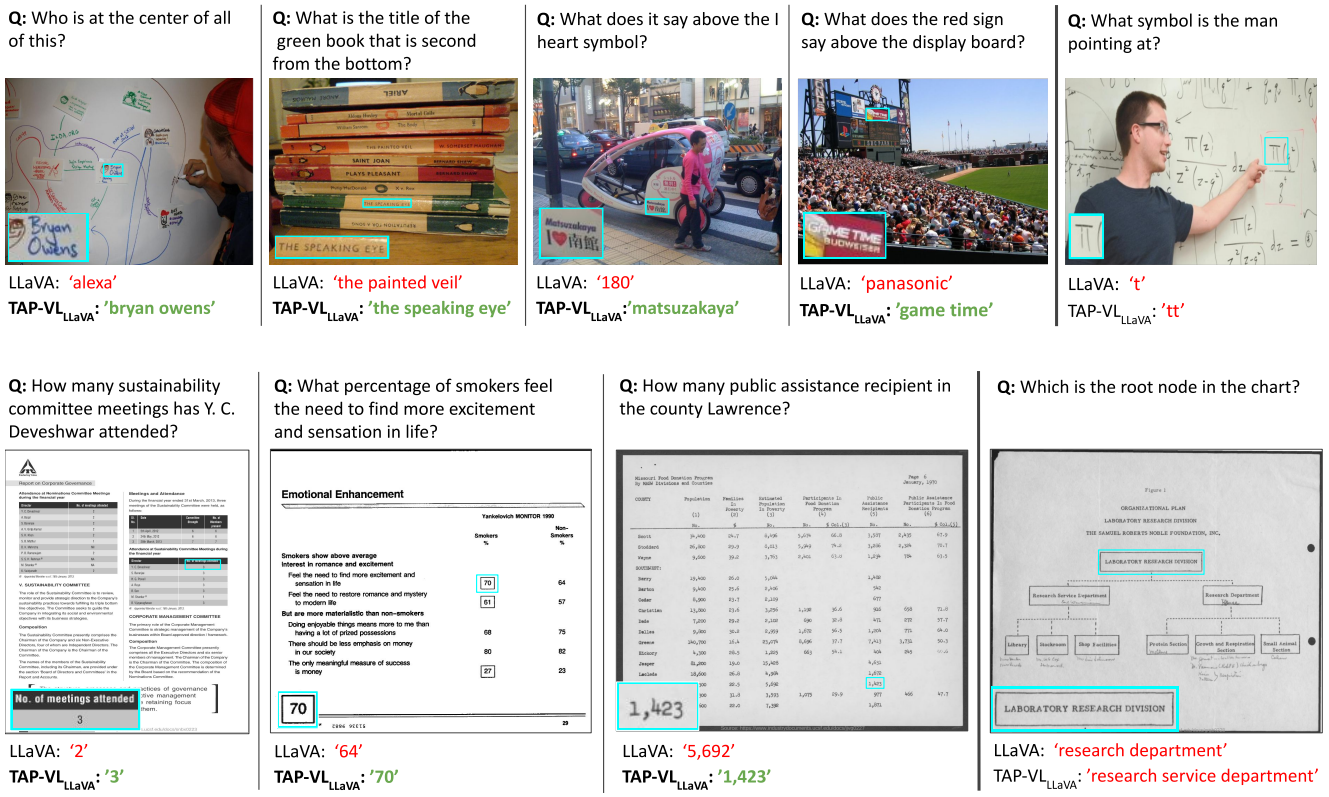


Figure 7. **Qualitative Improvements Demonstrated by TAP-VL**. Illustrative examples showcasing the improvements achieved by our method on scene-text (top) and document understanding (bottom) benchmarks using LLaVA. TAP-VL enhances the base model's ability to grasp OCR and layout information, yielding significant improvements across both benchmark types.

Pretraining Tasks			Samples	Average	
Denoising	Contrastive	Matching		Scene-Text	Documents
✗	✗	✗	-	86.8†	58.3†
✓	✗	✗	13M	85.9†	60.7†
✓	✓	✗	13M	87.5†	62.0†
✓	✓	✓	13M	89.5†	62.2†
✓	✓	✓	2M	91.2	62.8
✓	✓	✓	13M	91.7	63.4
✓	✓	✓	22M	91.0	65.2
✓	✓	✓	31M	92.0	65.7

Table 4. **Pretraining Objectives and Data**. The top section of the table depicts the influence of each pretraining objective. Below, we investigate the impact of increasing the data size and pretraining iterations. Notably, performance in both document and scene-text tasks shows ongoing improvement without reaching a plateau, even after processing 31 million documents. † Experiments which underwent shorter training.

Method	LLM	VQAv2	COCO
		VQAScore	CIDEr
InstructBlip [20]	Flan-T5-XL	78.2	141.1
+ TAP-VL$_{\text{Light}}$		78.2	143.1
+ TAP-VL		78.1	143.8
InstructBlip [20]	Flan-T5-XXL	78.9	138.2
+ TAP-VL$_{\text{Light}}$		78.5	141.3
+ TAP-VL		78.8	141.3
LLaVA-1.6 [39]	Mistral-7B	80.7	133.3
+ TAP-VL$_{\text{Light}}$		80.7	135.3
+ TAP-VL		80.8	135.5
Qwen-VL [37]	Qwen-7B	80.4	126.7
+ TAP-VL$_{\text{Light}}$		80.4	127.2
+ TAP-VL		80.3	127.2

Table 5. **TAP-VL Results**. TAP-VL results on VQAv2 and COCO, integrated into InstructBLIP, LLaVA-1.6, and Qwen-VL.

ples, 31M, indicating that pretraining on documents has the potential to enhance OCR-oriented scene-text performance.

Performance on General Benchmarks: We evaluated our system on a general VQA dataset (VQAv2) and a CAPS dataset (COCO), which do not specifically require OCR. Our analysis indicates that TAP-VL either preserves or enhances the non-OCR capabilities of the baseline vision-language model. Integration with InstructBlip (XXL) resulted in a performance boost on the COCO dataset by 3.1%, with only a marginal decline on VQAv2 of 0.1%.

6. Discussion and Conclusion

In this study, we proposed TAP-VL, a novel method for integrating OCR information into VL models. Our approach, which treats OCR as a distinct modality, utilizes a lightweight transformer-based OCR adapter to compress OCR and layout information into fixed-length sequences

for input into VL models. Through extensive experiments, we demonstrated consistent performance improvements across various benchmarks, including both natural images and document-based VL tasks. Additionally, we proposed TAP-VL$_{\text{Light}}$ which significantly reduces computational costs by using a concise representation of the OCR. Overall, our findings suggest that integrating OCR information into VL models can lead to substantial performance gains and computational savings, making it a promising avenue for future research in the field.

References

[1] Aviad Aberdam, Ron Litman, Shahar Tsiper, Oron Anschel, Ron Slossberg, Shai Mazor, R Manmatha, and Pietro Perona. Sequence-to-sequence contrastive learning for text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15302–15312, 2021. 12

[2] Aviad Aberdam, Roy Ganz, Shai Mazor, and Ron Litman. Multimodal semi-supervised learning for text recognition. *arXiv preprint arXiv:2205.03873*, 2022.

[3] Aviad Aberdam, David Bensaïd, Alona Golts, Roy Ganz, Oren Nuriel, Royee Tichauer, Shai Mazor, and Ron Litman. Clipter: Looking at the bigger picture in scene text recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21706–21717, 2023. 12

[4] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 1, 2

[5] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 1

[6] Srikar Appalaraju, Peng Tang, Qi Dong, Nishant Sankaran, Yichu Zhou, and R. Manmatha. Docformerv2: Local features for document understanding, 2023. 2, 3, 5, 6

[7] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. 3, 6

[8] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. 1, 2, 3, 6

[9] Ali Furkan Biten, Rubèn Tito, Andrés Mafla, Lluis Gomez, Marçal Rusiñol, C.V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. Scene text visual question answering. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4290–4300, 2019. 6, 13

[10] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R. Manmatha. Latr: Layout-aware transformer for scene-text vqa. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16527–16537, 2022. 2, 3, 5, 12

[11] Ali Furkan Biten, Rubèn Tito, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. Ocr-idl: Ocr annotations for industry document library dataset. In *Computer Vision – ECCV 2022 Workshops*, pages 241–252, Cham, 2023. Springer Nature Switzerland. 2, 5

[12] Tsachi Blau, Sharon Fogel, Roi Ronen, Alona Golts, Roy Ganz, Elad Ben Avraham, Aviad Aberdam, Shahar Tsiper, and Ron Litman. Gram: Global reasoning for multi-page vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15598–15607, 2024. 1

[13] Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu. X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages, 2023. 3

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. 14

[15] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server, 2015. 1, 13

[16] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, Siamak Shakeri, Mostafa Dehghani, Daniel Salz, Mario Lucic, Michael Tschannen, Arsha Nagrani, Hexiang Hu, Mandar Joshi, Bo Pang, Ceslee Montgomery, Paulina Pietrzyk, Marvin Ritter, AJ Piergiovanni, Matthias Minderer, Filip Pavetic, Austin Waters, Gang Li, Ibrahim Alabdulmohsin, Lucas Beyer, Julien Amelot, Kenton Lee, Andreas Peter Steiner, Yang Li, Daniel Keysers, Anurag Arnab, Yuanzhong Xu, Keran Rong, Alexander Kolesnikov, Mojtaba Seyedhosseini, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali-x: On scaling up a multilingual vision and language model, 2023. 2, 6

[17] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, Daniel Salz, Xi Xiong, Daniel Vlasic, Filip Pavetic, Keran Rong, Tianli Yu, Daniel Keysers, Xiaohua Zhai, and Radu Soricut. Pali-3 vision language models: Smaller, faster, stronger, 2023. 6

[18] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam

Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model, 2023. 1, 2, 3, 6

[19] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. 6

[20] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 1, 2, 3, 6, 8

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. 5

[22] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32, 2019. 4

[23] Roy Ganz and Michael Elad. Clipag: Towards generator-free text-to-image generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3843–3853, 2024. 1

[24] Roy Ganz, Oren Nuriel, Aviad Aberdam, Yair Kittenplon, Shai Mazor, and Ron Litman. Towards models that can see and read. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 21718–21728, 2023. 2, 3, 5, 6

[25] Roy Ganz, Yair Kittenplon, Aviad Aberdam, Elad Ben Avraham, Oren Nuriel, Shai Mazor, and Ron Litman. Question aware vision transformer for multimodal reasoning. *arXiv preprint arXiv:2402.05472*, 2024. 1, 2

[26] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 13

[27] Benjamin Hsu, Xiaoyu Liu, Huayang Li, Yoshinari Fujinuma, Maria Nadejde, Xing Niu, Yair Kittenplon, Ron Litman, and Raghavendra Pappagari. M3t: A new benchmark dataset for multi-modal document-level machine translation. *arXiv preprint arXiv:2406.08255*, 2024. 2

[28] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 5, 12

[29] Wenbo Hu, Yifan Xu, Yi Li, Weiyue Li, Zeyuan Chen, and Zhuowen Tu. Bliva: A simple multimodal llm for better handling of text-rich visual questions. *arXiv preprint arXiv:2308.09936*, 2023. 1

[30] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. 6

[31] Jordy Van Landeghem, Rubén Tito, Łukasz Borchmann, Michał Pietruszka, Paweł Józiak, Rafał Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Ackaert, Ernest Valveny, Matthew Blaschko, Sien Moens, and Tomasz Stanisławek. Document understanding dataset and evaluation (dude), 2023. 6, 13

[32] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 14

[33] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 1

[34] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, 2023. 2, 3, 4

[35] Ron Litman, Shahar Tsiper, Royee Tichauer, Srikar Appalaraju, Shai Mazor, and R Manmatha. Visfocus: Prompt-guided vision encoders for ocr-free dense document understanding. 1

[36] Ron Litman, Oron Anschel, Shahar Tsiper, Roee Litman, Shai Mazor, and R Manmatha. Scatter: selective context attentional scene text recognizer. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11962–11972, 2020. 12

[37] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 8

[38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1, 2, 3

[39] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 3, 6, 8

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 12

[41] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 12

[42] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland, 2022. Association for Computational Linguistics. 13

[43] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2199–2208, 2021. 1, 6, 13

[44] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. Infographicvqa. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2582–2591, 2022. 6, 13

[45] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019. 13

[46] Oren Nuriel, Sharon Fogel, and Ron Litman. Textadain: Paying attention to shortcut learning in text recognizers. In *European Conference on Computer Vision*, pages 427–445. Springer, 2022. 12

[47] Artemis Panagopoulou, Le Xue, Ning Yu, Junnan Li, Dongxu Li, Shafiq Joty, Ran Xu, Silvio Savarese, Caiming Xiong, and Juan Carlos Niebles. X-instructblip: A framework for aligning x-modal instruction-aware representations to llms and emergent cross-modal reasoning, 2023. 2, 3

[48] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 5, 6

[49] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: A dataset for image captioning with reading comprehension. In *Computer Vision – ECCV 2020*, pages 742–758, Cham, 2020. Springer International Publishing. 1, 6, 13

[50] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8309–8318, 2019. 1, 6, 13

[51] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*, 2022. 6

[52] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language, 2022. 6

[53] Peng Wang, Shijie Wang, Junyang Lin, Shuai Bai, Xiaohuan Zhou, Jingren Zhou, Xinggang Wang, and Chang Zhou. One-peace: Exploring one general representation model toward unlimited modalities, 2023. 3

[54] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer, 2021. 6

[55] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. Tap: Text-aware pre-training for text-vqa and text-caption. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8751–8761, 2021. 3

[56] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding, 2023. 3

[57] Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. Llavar: Enhanced visual instruction tuning for text-rich image understanding, 2023. 3

Supplementary Materials for TAP-VL: Text Layout Aware Pretraining for Enriched Vision-Language Models

A. Implementation Details

For all considered architectures, we used a uniform training procedure that involves applying LoRa [28] to the LLM and fine-tuning the vision projection module while keeping the ViT frozen. When implementing TAP-VL, we additionally pretrained and fine-tuned the OCR module during the Text Layout-Aware Pretraining and OCR-to-Language Alignment stages. Both the baseline models and TAP-VL were fine-tuned using the same mixture of datasets described in the next section (OCR-Vision-to-Language Alignment). All experiments were conducted on 8 Nvidia A100 (40G) GPUs using bfloat16 precision. The OCR system used in this paper is Textract OCR* [1–3, 36, 46].

A.1. Training Datasets

For our experiments, we employed two distinct dataset combinations for the OCR-to-Language Alignment and OCR-Vision-to-Language Alignment phases. The datasets used in these phases, along with their evaluation metrics and splits, are detailed in Tab. 7.

OCR-to-Language Alignment: This phase was dedicated to OCR-centric VQA datasets, where answers can be directly inferred from the OCR text including: DocVQA, InfoVQA, ChartQA, OCRVQA, TextVQA, and STVQA.

OCR-Vision-to-Language Alignment was trained on a combination of OCR-centric and non-OCR-centric datasets, specifically DocVQA, InfoVQA, ChartQA, OCRVQA, TextVQA, STVQA, TextCaps, COCO, and VQAv2.

A.2. Training hyperparameters

In each stage of TAP-VL's training, the OCR module consistently generated 32 query tokens. Additionally, the AdamW optimizer [40] and the Cosine Annealing scheduler [41] were uniformly applied. Beyond these constants, each stage was characterized by its own distinct set of hyperparameters

Text Layout-Aware Pretraining: The pretraining stage comprised 140,000 training steps, with a learning rate $1e-4$ and and a batch size of 224. The OCR module was trained with a maximum of 512 token in the OCR module and a masking probability of 0.15. More information about the pretraining optimization can be found in Tab. 8.

OCR-to-Language Alignment: Detailed in Tab. 8, this stage maintained a uniform structure across models, with $300K$ training steps, 1000 warmup steps, a learning rate of $2e-5$, and a batch size of 24 for InstructBlip and 32 for the other models. The OCR branch's maximum token length was set to 1,024 for this training phase.

OCR-Vision-to-Language Alignment: In Tab. 9, we present the hyperparameters for the OCR-vision-to-language alignment phase. This phase mirrored the prior alignment stage's training steps, batch size, warmup procedure, and learning rates. During the training phase, the OCR module was trainable for all models except Qwen-VL. The image resolution used to feed the frozen vision encoder was the original one used by the baseline models, with InstructBlip XL and XXL set at 224, LLaVA-1.6 at 336, and Qwen-VL at 448. The maximum OCR module length was set to 2,000 tokens for TAP-VL$_{\text{InstructBlip XL}}$ and 1,400 for the other configurations. The LLM prompt length was constrained by RAM limitations, set to different maximums tailored to each model configuration.

Template
Could you write a short image caption?
Could you write a short image description?
What does this image show?
Could you write a short description for the image?
Could you write a description for the photo?
Could you provide a description of what is presented in the photo?
Could you briefly describe the content of the image?
Can you briefly explain what you see in the image?
Could you use a few words to describe what you perceive in the photo?
Could you provide a short depiction of the picture?
Could you use a few words to illustrate what is happening in the picture?

Table 6. **Instruction Templates for Captioning.** Overview of templates employed across captioning datasets to guide caption generation.

A.3. Instruction templates

For the VQA-based datasets, we use the given question as the instruction. For the captioning datasets, we randomly select an instruction that asks the model to describe the image among the one in Tab. 6.

A.4. Pretraining data preparation:

Each document contains OCR tokens, denoted as $t_1, t_2, \ldots, t_n$ with corresponding bounding boxes $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$. To create training pairs, we randomly mask spans of OCR tokens along with their positional information. Specifically, we:

- Sample M spans, each defined by a start index s_i and an end index e_i.
- Replace tokens and bounding boxes in each span $(s_i, s_i + 1, \ldots, e_i)$ with a special token `<extra_id_i>` and the minimal bounding box covering the span, respectively (following the method in [10]).
- Generate pairs consisting of the noisy OCR input and the masked words, i.e., the original tokens in the masked spans. The masked words are represented as a sequence

*https://aws.amazon.com/textract/

Task	Dataset	Description	Train split	Eval split	Metric
Scene-Text VQA	TextVQA [50]	Text-oriented VQA on natural images	train	val	vqa-score(↑)
	STVQA [9]	Text-oriented VQA on natural images	train	test	ANLS(↑)
	OCRVQA [45]	Text-oriented VQA on natural images	train	-	Acc@1(↑)
Document Understanding	DocVQA [43]	VQA on single page scanned documents	train	test	ANLS(↑)
	InfoVQA [44]	VQA on infographic images	train	test	ANLS(↑)
	ChartQA [42]	VQA on chart images	train (human)	-	RA
	Dude [31]	VQA on multipage scanned documents	-	test	ANLS(↑)
Image Caption	COCO [15]	Captioning of natural images	train	test	CIDEr(↑)
Scene-Text Caption	TextCaps [49]	Text-oriented Captioning of natural images	train	test	CIDEr(↑)
General VQA	VQAv2 [26]	VQA on natural images	train	val	vqa-score(↑)

Table 7. **Datasets** used during the finetuning stages.

Stage	# Steps	Batch Size	Base LR	# Warmup steps	Weight Decay	OCR-Q Prompt	# OCR module token	Mask density
Pretraining	$140K$	224	$1e-4$	1000	0.05	<extra_id_i> {masked_words_i}	512	0.15
Alignment	$300K$	$24^{\dagger}$	$2e-5$	1000	0.05	Question: {Instruction}	1024	0

Table 8. **Model hyperparameters used during the Text Layout-Aware Pretraining and OCR-to-language Alignment.** † Qwen-VL and LLaVA was trained using a batch size of 32 during the OCR-to-language Alignment.

VL Model	# Steps	# LLM token	# OCR module token	LoRA $(\alpha, r, dropout, modules)$	OCR module	Image resolution
InstructBlip	$300K$	1024	2000	$16, 32, 0.05, [W_q, W_v]$	Trained	224
InstructBlip XXL	$300K$	400	1024	$16, 32, 0.05, [W_q, W_v]$	Trained	224
LLaVA-1.6	$300K$	400	1024	$16, 32, 0.05, [W_q, W_v]$	Trained	336
Qwen-VL	$100K$	600	1400	$16, 32, 0.05, [W_q, W_v]$	Frozen	448

Table 9. **Hyperparameters for OCR-Vision-to-Language Alignment.** Parameters such as batch size, learning rate, warm-up period, and optimizer are not specified here, as they remain consistent with those used in the OCR-to-Language Alignment stage.

(<extra_id_i> value_i>)$_{i=1}^{i=M}$ where value_i is the original text in the i-th masked span.

A.5. Layout-Aware pretraining

For all the pretraining objectives, the OCR encoder produces rich embeddings from the noisy OCR. We denote these embeddings as $O \in \mathbb{R}^{B \times l \times d_{\text{ocr}}}$, where B is the batch size, l is the number of noisy OCR tokens, and d_{ocr} is the dimensionality of the embeddings. Additionally, we define $\mathbf{R}_q \in \mathbb{R}^{B \times K \times d}$ to represent the learnable queries, where B is the batch size, K is the number of learnable queries, and d is their dimensionality. The representation of the M mask words, preceded by the task specific special token, is denoted as $\mathbf{R}_m \in \mathbb{R}^{B \times (2M+1) \times d}$.

OCR-Grounded Mask Denoising: In this pretraining objective, we use a <dec> special token. The OCR-Q processes $\mathbf{R}_q$ through its self-attention (SA) layers, while processing $\mathbf{R}_m$ using causal self-attention (CSA) layers conditioned on $\mathbf{R}_q$. This results in mask-words representations that integrate contextual information from the queries tokens. Subsequently, the queries representations are updated using the noisy OCR content using a cross-attention (CA) layer. Two finals feed-forward (FF) layers are applied on top of $\mathbf{R}_q$ and $\mathbf{R}_m$. Formally, we compute:

$$\mathbf{R}_q^{(out)} = \text{FF}\left(\text{CA}\left(\text{SA}\left(\mathbf{R}_q^{(in)}\right), O\right)\right) \quad (1)$$

$$\mathbf{R}_m^{(out)} = \text{FF}\left(\text{CSA}\left(\mathbf{R}_m^{(in)} \mid \mathbf{R}_q^{(in)}\right)\right) \quad (2)$$

where (in) and (out) represent the input and output representations.

We then apply a language modeling loss $\mathcal{L}_{\text{LM}}$ over the output representations $\mathbf{R}_m^{(\text{out})}$ to recover the original masked content. Specifically, we pass $\mathbf{R}_m^{(\text{out})}$ through a softmax function to obtain the predicted token probabilities:

$$\hat{y}_i = \text{Softmax}\left(\mathbf{R}_m^{i(\text{out})}\right) \quad (3)$$

The language modeling loss $\mathcal{L}_{\text{LM}}$ is then computed using the cross-entropy between the predicted probabilities and the ground truth tokens $(y)_{i=1}^{M}$:

$$\mathcal{L}_{\text{LM}} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{M}\log\left(\hat{y}_j | y_{t<j}\right). \quad (4)$$

OCR-Mask Contrastive Learning: In this objective, we use a the <cls> token and its specific representation is denoted as $\mathbf{R}_t \in \mathbb{R}^{B\times 1\times d}$.

The OCR-Q processes $\mathbf{R}_q$ and $\mathbf{R}_m$ using two independent self-attention layers. This results in mask-words representations that are independent from the from the queries tokens representation. Subsequently, the queries representations are updated using the noisy OCR content using a cross-attention layer. Two finals feed-forward layers are applied on top of $\mathbf{R}_q$ and $\mathbf{R}_m$. Formally, we compute:

$$\mathbf{R}_q^{(out)} = \text{FF}\left(\text{CA}\left(\text{SA}\left(\mathbf{R}_q^{(in)}\right), O\right)\right) \tag{5}$$

$$\mathbf{R}_m^{(out)} = \text{FF}\left(\text{SA}\left(\mathbf{R}_m^{(in)}\right)\right) \tag{6}$$

The $\mathbf{R}_t^{(out)}$ representation is then extracted from $\mathbf{R}_m^{(out)}$ in order to compute the pairwise similarity between $\mathbf{R}_q^{(out)}$ and $\mathbf{R}_t^{(out)}$, yielding $\mathbf{P}_{qt} \in \mathbb{R}^{B\times B\times K}$, and subsequently, the maximum similarity is selected across the last dimension, resulting in $\mathbf{S}_{qt} \in \mathbb{R}^{B\times B}$. Finally, we apply the contrastive learning loss [14], with a temperature scalar τ on the $\mathbf{S}_{qt}$ matrix:

$$\mathcal{L}_{\text{Contrastive}}(\mathbf{S}) = -\frac{1}{B}\sum_{i=1}^{B}\log\left(\frac{\exp(\mathbf{S}_{ii}/\tau)}{\sum_{j\neq i}\exp(\mathbf{S}_{ij}/\tau)}\right) \tag{7}$$

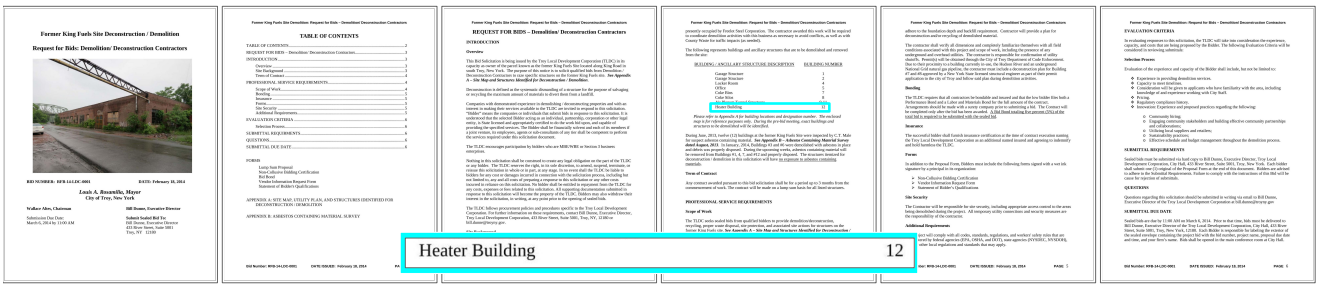
OCR-Mask Matching: In this objective, we compute $\mathbf{R}_q^{(\text{out})}$, $\mathbf{R}_m^{(\text{out})}$, and $\mathbf{S}_{qt}$ similarly to OCR-Mask Matching. We employ a hard negative mining strategy [32] to select challenging negative examples based on the similarity values in the $\mathbf{S}_{qt}$ matrix. This approach yields pairs of representations $\mathbf{R}_q^{i(\text{out})}$ and $\mathbf{R}_m^{j(\text{out})}$ where $i \neq j$, indicating they come from different documents. Additionally, we consider pairs where $i = j$, which represent positive examples originating from the same document. The query representation $\mathbf{R}_q^{(\text{out})}$ is projected to $\mathbf{R}_q \in \mathbf{R}^{B\times 1\times 1}$ using a feed-forward layer followed by average pooling across the query dimension. This provides a single similarity value for each pair of noised OCR-masked words. Finally, we apply a binary cross-entropy loss to encourage the model to correctly detect matching pairs, where $y_i = 1$ if the pair matches and 0 otherwise.

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{B}\sum_{i=1}^{B} y_i \log\left(\sigma\left(\mathbf{P}_q^i\right)\right) + (1-y_i)\log\left(1-\sigma\left(\mathbf{P}_q^i\right)\right) \tag{8}$$

B. Additional results

Multipage qualitative results In Figs. 8 and 9, we display how our method integrates into LLaVA-1.6 to leverage the information available inside a multiple page document in order to answer a given question. For instance, the base model struggles to identify the "*number of the heater building*" located in the fourth page (over six), whereas TAP-VL effectively uses layout information to understand it.

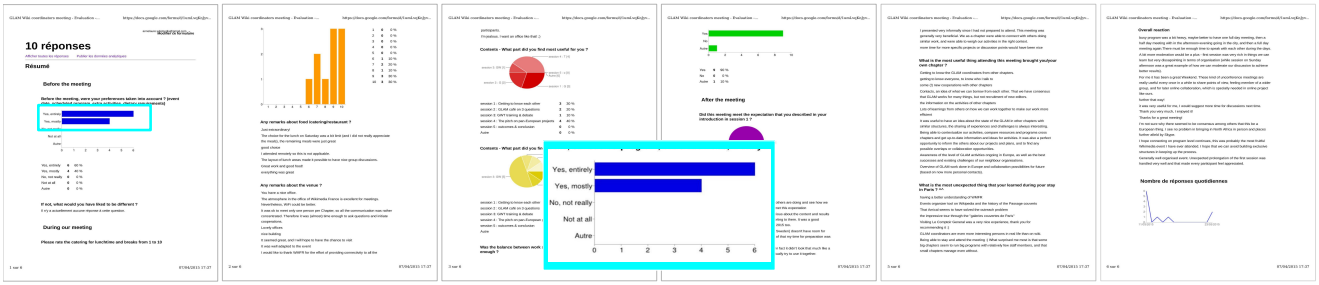
Q: What is the building number of the heater building?



LLaVA: 'NA'

TAP-VL$_{\text{LLaVA}}$: '12'

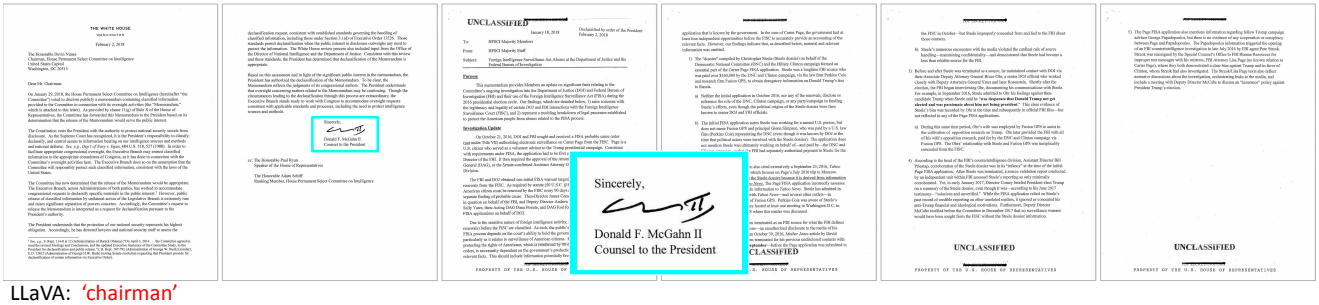
Q: What is the difference between the number of people who answered 'Yes, entirely' and 'Yes, mostly' in the chart on the first page of the document?



LLaVA: '1000000000'

TAP-VL$_{\text{LLaVA}}$: '2'

Q: What title holds the person who signed the letter on the page 2 of the document?

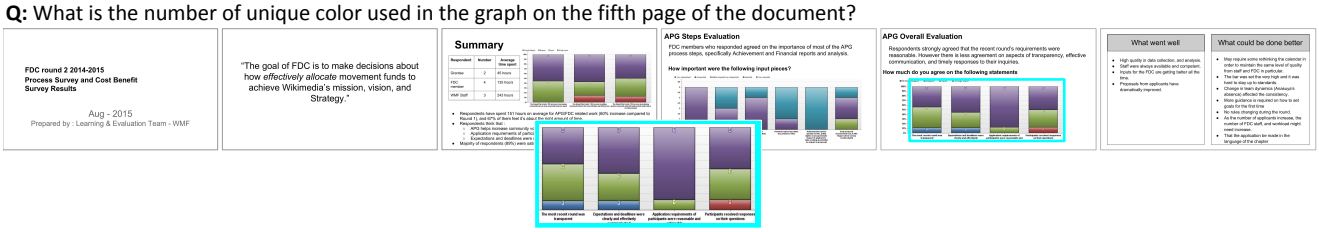


LLaVA: 'chairman'

TAP-VL$_{\text{LLaVA}}$: 'counsel to the president'

Figure 8. **Qualitative Improvements Demonstrated by TAP-VL**. Illustrative examples showcasing the improvements achieved by our method on multipage document VQA benchmarks using LLaVA. TAP-VL enhances the base model's ability to grasp OCR and layout information, yielding significant improvements across both benchmark types.

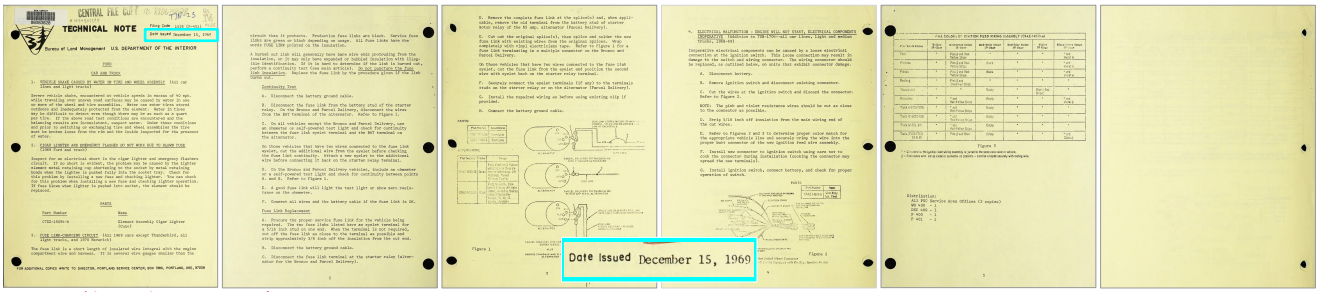
Q: What is the number of unique color used in the graph on the fifth page of the document?



LLaVA: '5'

TAP-VL$_{\text{LLaVA}}$: '4'

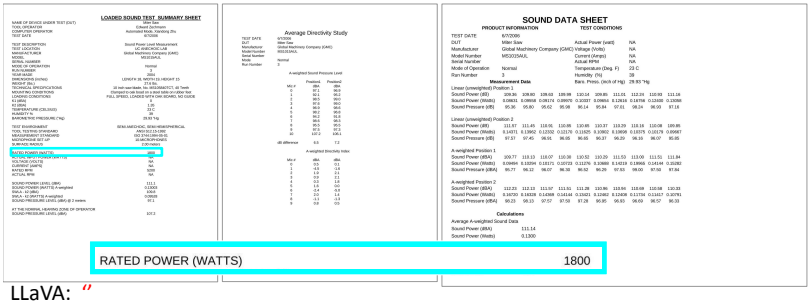
Q: When did this document published?



LLaVA: 'december 16, 1973'

TAP-VL$_{\text{LLaVA}}$: 'december 15, 1969'

Q: What is the rated power of the product in Watts?



LLaVA: ''

TAP-VL$_{\text{LLaVA}}$: '1800'

Figure 9. **Qualitative Improvements Demonstrated by TAP-VL**. Illustrative examples showcasing the improvements achieved by our method on multipage document VQA benchmarks using LLaVA. TAP-VL enhances the base model's ability to grasp OCR and layout information, yielding significant improvements across both benchmark types.