DocVLM: Make Your VLM an Efficient Reader

Mor Shpigel Nacson*† Aviad Aberdam* Roy Ganz Elad Ben Avraham
Alona Golts Yair Kittenplon Shai Mazor Ron Litman
AWS AI Labs

Abstract

Vision-Language Models (VLMs) excel in diverse visual tasks but face challenges in document understanding, which requires fine-grained text processing. While typical visual tasks perform well with low-resolution inputs, reading-intensive applications demand high-resolution, resulting in significant computational overhead. Using OCR-extracted text in VLM prompts partially addresses this issue but underperforms compared to full-resolution counterpart, as it lacks the complete visual context needed for optimal performance. We introduce DocVLM, *a method that integrates an OCR-based modality into VLMs to enhance document processing while preserving original weights. Our approach employs an OCR encoder to capture textual content and layout, compressing these into a compact set of learned queries incorporated into the VLM. Comprehensive evaluations across leading VLMs show that DocVLM significantly reduces reliance on high-resolution images for document understanding. In limited-token regimes (448×448), DocVLM with 64 learned queries improves DocVQA results from 56.0% to 86.6% when integrated with InternVL2 and from 84.4% to 91.2% with Qwen2-VL. In LLaVA-OneVision, DocVLM achieves improved results while using 80% less image tokens. The reduced token usage allows processing multiple pages effectively, showing impressive zero-shot results on DUDE and state-of-the-art performance on MP-DocVQA, highlighting DocVLM's potential for applications requiring high-performance and efficiency.*

1. Introduction

The ability to read and interpret text within images is crucial for numerous real-world applications, particularly in document understanding. This field encompasses diverse document types, from dense-text to infographics and multipage documents [25, 39–41, 50], involving tasks that require capabilities in text comprehension, layout understanding, and

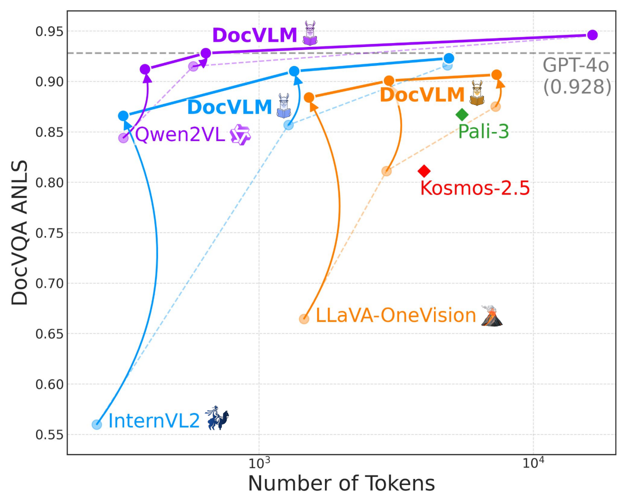

Figure 1. **DocVLM enhances VLMs' reading capabilities.** Integrating DocVLM (solid lines) in top-performing VLMs (dashed lines) consistently improves the performance across all token budgets, frequently surpassing the baseline at higher token counts.

visual interpretation. Despite progress in VLMs, processing such documents remains challenging [37], primarily due to the tension between resolution requirements and computational efficiency. While typical computer vision tasks achieve good performance with low-resolution inputs (typically 224×224 or 336×336 pixels), document analysis demands significantly higher resolutions, resulting in substantial computational overhead [10, 60].

To address these challenges, some methods incorporate OCR-extracted text directly into language model prompts [8, 17]. However, this approach typically lags behind OCR-free methods with full-resolution, as it fails to capture crucial visual context and layout information [54]. Moreover, it often produces long sequence inputs, increasing latency and computational costs, especially for dense documents.

Alternatively, recent VLMs [20, 31, 53] have introduced specialized mechanisms to reduce visual token count, such as image resizing, tiling limits, and feature downsampling. However, when applied to document understanding, these methods result in significant performance degradation, creating an undesirable trade-off between computational efficiency and accuracy, as demonstrated in Fig. 1. These lim-

*Equal contribution. Corresponding author: mnacson@amazon.com.
†Work done during an Amazon internship.

itations underscore the need for a more efficient approach that maintains high-performance document understanding while reducing computational demands.

To overcome these limitations, we introduce DocVLM, a model-agnostic method that enhances VLMs' reading ability by utilizing OCR information effectively. Our approach employs an OCR encoder to capture both contextual and layout details from OCR-extracted text, compressing these encodings into 64 learned queries. These queries are projected and fed into the LLM part of the VLM alongside visual features. Unlike some previous methods [5, 19, 30, 32], our compression mechanism avoids the need for a separate compression module or alterations to the LLM architecture.

We demonstrate DocVLM's effectiveness across multiple state-of-the-art VLMs: LlaVA-OneVision [31], InternVL2 [18], and Qwen2-VL [53], each employing a unique image token reduction technique. As illustrated in Figure 1, our method significantly improves performance, particularly in low input token regimes. Our experiments demonstrate consistently, across all studied VLMs and visual token budgets, that DocVLM's OCR encoder not only outperforms the baseline of inserting OCR words into VLMs but also achieves this superior performance when compressed to just 64 tokens. This dual advantage of improved performance and reduced token usage allows for better utilization of fixed token budgets, enabling the allocation of more tokens for visual processing and further enhancing overall performance.

Importantly, this reduction in sequence length improves the scalability of our approach, enabling its application to multipage document understanding tasks without additional training. We demonstrate that DocVLM can be seamlessly extended to long-context scenarios, such as multipage DocVQA [50, 51]. While current OCR-free approaches struggle with the overwhelming amount of data in multipage documents [27], our method achieves strong zero-shot performance on DUDE and surpasses the current state-of-the-art results on MP-DocVQA (86.3% vs. 80.3%), despite not being trained on multipage data.

Main contributions:

- DocVLM, a model-agnostic method that efficiently integrates OCR information into VLMs, capturing both text and layout without complex integration techniques.
- A compression mechanism that reduces OCR data into a compact set of typically 64 learned queries, significantly reducing computational overhead.
- Demonstration of DocVLM's effectiveness across different VLM architectures, LlaVA-OneVision, InternVL2, and Qwen2-VL, showing significant performance improvements in low input token regimes (448×448).
- Extension of DocVLM to long-context tasks, achieving strong zero-shot performance on DUDE and SOTA results on MP-DocVQA without multipage training data.

2. Related Work

OCR-free Document VLMs. Early VLMs[5, 9, 22, 23, 32, 36, 44, 59, 61] used relatively small image sizes (e.g., 224×224 and 336×336), performing well on natural-image tasks but falling short in document understanding. To address this, recent approaches enhance document understanding capabilities by operating on high-resolution images, developing various strategies to manage the resulting computational burden. Direct processing methods like Donut [29], PaLI-X [16], and Qwen2-VL [53] attempt full-resolution processing but often resort to image resizing for computational feasibility. Tile-based approaches such as UReader [56] and InternVL2 [18] improve efficiency by processing image tiles independently. Other methods, exemplified by LLaVA-1.5 [31] and LLaVA-OneVision [35], process full-scale images as tiles but downsample the resulting visual features. While these approaches offer different trade-offs, our experiments show their performance deteriorates significantly when constrained to fewer visual tokens.

OCR-Enhanced Document Understanding. The widespread availability of efficient, open-source OCR models and cost-effective commercial solutions has driven broad adoption of OCR-based approaches in document understanding [1–4, 21, 28, 34, 43, 46, 57, 58]. Several recent works [9, 16, 17, 19] have explored integrating OCR systems with VLMs by feeding extracted text directly into the language model component. Some approaches further enhance this integration by incorporating spatial layout information [12, 14, 22, 52, 54]. While these methods reduce the computational burden of processing high-resolution images, they currently lag behind OCR-free approaches in performance. Additionally, they face challenges with lengthy input sequences, particularly in multipage settings, which can increase latency and computational costs.

Document Representation Compression. To address efficiency challenges in processing documents, various compression techniques have been developed. For OCR-enhanced approaches, [14] proposed compressing the OCR signal in multi-page documents using a Compression Transformer, which, despite improving performance in multi-page benchmarks, introduces significant complexity to the system. In the OCR-free setting, generic VLM approaches like Q-former [32] and Resampler [5] compress visual features but struggle with text-dense images. Document-specific methods such as TokenPacker [33] and DocCompressor [27] achieve effective visual compression but show reduced performance on document understanding tasks. In contrast, rather than compressing high-resolution visual inputs, our DocVLM method operates on lower-resolution images and compresses the extensive OCR signal, including textual and layout information, into a compact set of features (typically 64).

3. Our Method

We present *DocVLM*, a model-agnostic approach that enhances VLMs' reading capabilities, enabling operation with lower-resolution inputs while maintaining or improving document understanding accuracy. Our design preserves the base VLM weights, facilitating easy integration across different model architectures and providing flexibility to balance OCR and visual tokens during inference.

3.1. Architecture

Our method introduces two main components that complement existing VLM architectures: an OCR encoder that processes OCR extracted text and layout information, and a query compression mechanism that distills this information into a compact representation. We integrate these components with pre-trained VLMs, which employ various strategies to control the number of visual tokens for efficient processing. Figure 2 illustrates the overall architecture.

OCR Encoder Architecture We utilize DocFormerV2 [7], a T5-based encoder-decoder [45] designed for document understanding, which incorporates vision, language, and spatial features. Specifically, we leverage only the encoder component, which comprises 344 million parameters, and omit its visual branch to eliminate redundancy with the VLM's vision capabilities and reduce computational complexity. The encoder processes two types of inputs: user instructions and OCR data from an OCR system, which consists of textual tokens and their corresponding 2D positional information [6, 7, 12, 14, 22, 26].

Query Compression Mechanism To efficiently integrate OCR information into VLMs, we introduce an instruction-aware compression mechanism that distills the OCR encoder's output into a compact set of learned queries. This mechanism significantly reduces the input sequence length for the language model while preserving essential document information. The compression process utilizes M learnable queries $\mathbf{Q}$ (typically $M = 64$), initialized randomly following the OCR encoder embeddings' distribution. These queries are processed by the OCR encoder alongside two types of embeddings: OCR embeddings ($\mathbf{E}_{\text{OCR}}$), which encode both OCR tokens and their bounding boxes, and instruction embeddings ($\mathbf{E}_{\text{Instructions}}$). The encoding process can be represented as:

$$\text{Encoder}([\mathbf{E}_{\text{OCR}}, \mathbf{E}_{\text{Instructions}}, \mathbf{Q}]).$$

From the encoder output, we retain only the M features corresponding to the learned queries. These compressed features are then projected to match the VLM's hidden dimension and concatenated with the visual tokens before entering the language model. This compression significantly reduces the LLM's input sequence length, enabling either more efficient processing or, under a fixed token budget, allocation of additional tokens to visual features.

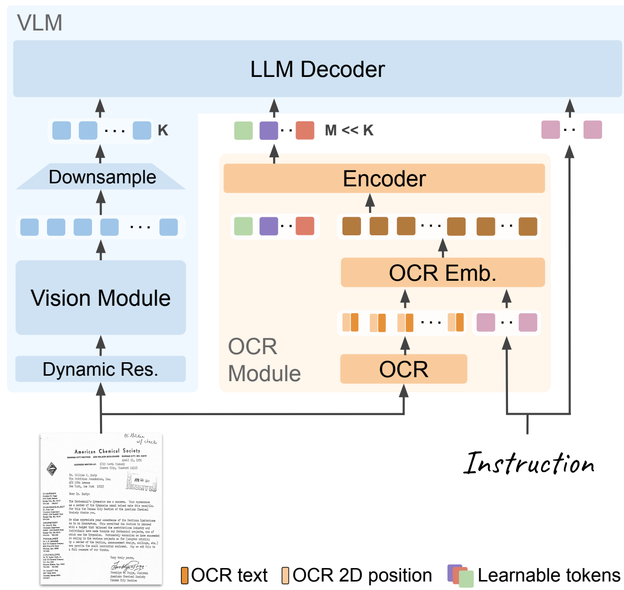

Figure 2. **DocVLM Architecture**. DocVLM enhances document understanding in frozen VLMs by integrating an OCR module with a query compression mechanism. By condensing OCR data into M = 64 learnable tokens, DocVLM effectively complements visual information, surpassing the VLM's inherent approaches of increasing image resolution or visual feature dimensions.

Vision Process OCR-free VLMs employ different visual processing methods and various strategies to control the number of visual tokens, aiming to reduce the computational cost of processing high-resolution images needed for document understanding. These approaches can be grouped into three main paradigms:

1. **Full Image Processing with Image Resizing (e.g., Qwen2-VL [53]):** In this approach, the model processes the entire image as a single input, controlling the number of visual tokens by resizing the image to a fixed or range-constrained image resolutions. While this image processing preserves global context, it incurs quadratic computational complexity with respect to the number of visual tokens.
2. **Patch-Based Processing with Controlled Tile Count (e.g., InternVL2 [18]):** This strategy segments the input image into spatial tiles, processing each independently, and controls the number of visual tokens by limiting the number of tiles. While most implementations incorporate a low-resolution global view, the primary focus on local processing may compromise global context understanding. The computational complexity for this approach scales linearly with respect to the number of tiles as the image increase. The result is improved memory efficiency compared to processing the full image at once, especially for large images.
3. **Full-Scale Processing with Feature Downsampling (e.g., LlaVA-OneVision [35]):** Some VLMs initially

process full-scale images, but then downsample the visual features into maximal token count before feeding them to the LLM. While this method captures both global and local context, it introduces significant computational overhead during initial full-scale processing.

Experimental results confirm that our OCR query compression mechanism significantly enhances document understanding capabilities across all three visual processing strategies, demonstrating its effectiveness as a universal enhancement to existing VLM architectures.

3.2. Training Strategy

Our training strategy aims to integrate the OCR modality into existing VLMs while preserving their core strengths. To achieve this, we keep the VLM completely frozen throughout the process and train only the newly introduced OCR components, i.e., the learnable queries, the OCR encoder, and the projection layer. We employ a two-stage training strategy to gradually incorporate the OCR modality into the pre-trained VLM:

Stage I: OCR-LLM Alignment. During this stage, we withhold image input from the VLM, forcing the model to rely solely on the newly introduced OCR modality. This approach ensures full utilization of OCR data, aligns OCR components with the LLM input space, and reduces sequence length, improving training efficiency. Given the focus on text input, our dataset selection concentrates on text-related tasks. We begin by training only the randomly initialized components: the learnable queries and projection layer. This allows these components to adapt without disrupting the pretrained OCR encoder. Subsequently, we unfreeze the OCR encoder for fine-tuning, enabling a more comprehensive alignment of the entire encoder to the VLM.

Stage II: Vision Alignment. In this final stage, we incorporate visual information extracted from the visual encoder, encouraging the OCR components to complement the visual features. Our experiments reveal this stage has a particularly strong effect when using fewer learned queries, allowing the compressed OCR information to better supplement the information acquired from the visual modality (see Sec. 5). During this stage, we add more visually focused datasets to the training process. Note that despite our method preserving the original VLM weights, it might implicitly inject bias through prompt tuning. To avoid this, the training data should represent all tasks of interest.

3.3. Multipage Document Extension

We conduct our training procedure on single-page data only. However, our approach can be extended to operate on multipage documents. Given a multipage input and its OCR information, the VLM independently processes each page image and concatenates the resulting visual features. For the OCR information, we explore two strategies: *Global Encoding*, which compresses the entire document's OCR information into 64 learnable queries, and *Page-wise Encoding*, which compresses each page's OCR information separately into 64 learnable queries and then concatenates them, resulting in 64 $\times$ number of pages learned queries. After processing the OCR information using either strategy, we feed the resulting compressed OCR features along with the concatenated visual features into the LLM.

Our experiments demonstrate that both approaches are highly effective and efficient in processing multipage documents. Using either approach with a restricted number of visual tokens, we obtain strong zero-shot results on DUDE [51] and state-of-the-art results on MP-DocVQA [50]. The page-wise encoding strategy yields slightly better results for lower numbers of visual tokens.

4. Experiments

4.1. Experimental Setting

Model Integration: We evaluate DocVLM through integration with three leading open-source VLMs: LLaVA-OneVision [31], InternVL2 [18], and Qwen2-VL [53]. As discussed in Sec. 3.1, these models employ distinct token reduction strategies, enabling us to assess DocVLM's effectiveness across different visual processing approaches.

Training: Our training protocol, detailed in Sec. 3.2, employs a two-phase strategy. The initial phase focuses on text-centric tasks using datasets spanning document understanding (DocVQA [40], InfoVQA [41]), scene text analysis (ST-VQA [11], TextVQA [48], OCR-VQA [42]), and specialized tasks (ChartQA [39], TextCaps [47], TAT-DQA [62]). The subsequent vision alignment phase incorporates additional visual-centric datasets: COCO Caption [15] and VQA-V2 [24].

Evaluation: For evaluation, we focus on five key benchmarks: DocVQA, TextVQA, ST-VQA, InfoVQA, and TextCaps. Results are reported on test sets where available, with TextVQA and TextCaps evaluated on validation sets due to test server restrictions. We use ANLS as the evaluation metric for all datasets, except TextVQA, which uses VQAScore, and TextCaps, which uses CIDEr. To demonstrate DocVLM's generalization capabilities, we conduct zero-shot evaluation on multipage document understanding benchmarks: DUDE [51] and MP-DocVQA [50]. This zero-shot performance is particularly noteworthy as our model is trained exclusively on single-page documents. The supplementary presents additional implementation details, including hyperparameters and optimization strategies, and robustness analysis to different OCR systems.

4.2. State-of-the-art Comparisons

Table 1 presents comprehensive comparisons between DocVLM and other state-of-the-art methods across var-

Method	# Tok.	#P	DocVQA	TextVQA	ST-VQA	InfoVQA	TextCAPS	MP-DocVQA	DUDE*
No Token Limitations									
GPT-4o			92.8	77.4	-	79.2	-	-	-
Gemini 1.5 Pro			93.1	78.7	-	81.0	-	-	-
GPT-4V			87.2	78.0	-	75.1	-	-	-
KOSMOS-2.5-CHAT	4K	1.3B	81.1	40.7	-	41.3	-	-	-
TextSquare	2.5K	8.6B	84.3	66.8	-	51.5	-	-	-
ScreenAI	3.5K	5B	87.8	-	-	57.8	-	72.9	-
ScreenAI+OCR	4.3K	5B	89.9	-	-	65.9	-	77.1	-
Pali-3	5.5K	5B	86.7	79.5	84.1	57.8	158.8	-	-
Pali-3+OCR	6.3K	5B	88.6	80.8	85.7	62.4	164.3	-	-
# Tokens $\leq$ 1.5k									
UReader	841	7B	65.4	57.6	-	42.2	118.4	-	-
Monkey	1.3K	9B	66.5	64.3	-	36.1	93.2	-	-
TextMonkey	768	9B	73.0	65.9	-	28.6	-	-	-
Vary	256	7B	76.3	-	-	-	-	-	-
DocOwl2	324	8B	80.7	66.7	-	46.4	131.8	69.4	46.8
GRAM	900	1B	85.3	-	-	-	-	80.3	51.2
$\text{GRAM}_{\text{C-Former}}$	256	1B	87.6	-	-	-	-	77.6	45.5
DocFormer v2	1K	1B	87.8	64.0	71.8	48.8	-	76.4	48.4
LLaVA-OneVision	7K	7B	87.5	76.1	71.1	68.8	138.0	OOM	OOM
LLaVA-OneVision	1.5K	7B	66.5	72.1	70.6	45.6	112.9	41.8	28.7
$\text{DocVLM}_{\text{LLaVA-OneVision}}$ (Ours)	1.5K	7B	88.4	76.9	70.8	61.0	145.3	77.9	43.8
InternVL 2	3.1K	8B	91.6	77.4	-	74.8	-	OOM	OOM
InternVL 2	256	8B	56.0	65.7	65.7	38.4	51.1	51.0	30.5
$\text{DocVLM}_{\text{InternVL2}}$ (Ours)	320	8B	86.6	71.2	74.3	57.6	119.4	76.2	43.3
InternVL 2	1280	8B	85.7	75.5	68.3	61.5	43.7	78.1	42.2
$\text{DocVLM}_{\text{InternVL2}}$ (Ours)	1344	8B	91.0	76.7	76.7	65.4	123.4	81.8	45.6
Qwen2-VL	16k	7B	94.5	84.3	70.7	76.5	150.2	OOM	OOM
Qwen2-VL	320	7B	84.4	78.0	70.1	54.1	142.1	73.0	41.5
$\text{DocVLM}_{\text{Qwen2-VL}}$ (Ours)	320	7B	91.2	79.6	76.5	61.2	144.3	81.7	46.1
Qwen2-VL	576	7B	91.5	82.3	70.5	65.3	145.0	82.1	45.9
$\text{DocVLM}_{\text{Qwen2-VL}}$ (Ours)	576	7B	**92.8**	**82.8**	**79.8**	**66.8**	**150.4**	**84.5**	**47.4**

Table 1. **Comparison with State-of-the-Art Methods.** Performance evaluation of DocVLM against state-of-the-art approaches on document understanding benchmarks. Results are categorized into unconstrained models and those with a 1.5k token limit. In the constrained token regime, DocVLM consistently enhances the performance of baseline VLMs across various tasks and visual token budgets. Notably, DocVLM paired with Qwen2-VL (576 tokens) achieves superior performance across all evaluated datasets, including state-of-the-art zero-shot accuracy on DUDE. '$\star$' indicates zero-shot evaluation, with grey entries denoting non-zero-shot results.

ious document understanding benchmarks, highlighting DocVLM's ability to improve performance under token constraints. We categorize the results into two main groups: methods without token constraints (both closed and open-source models) and those operating under a 1.5k token limit. We mainly focused on models with around 7B parameters and include methods using an OCR system such as Pali-3 [10] ScreenAI [8], DocFormerV2 [7], and GRAM [14].

To evaluate DocVLM's effectiveness under token constraints, we integrated it with three baseline models: LlaVA-OneVision [31], InternVL2 [18], and Qwen2-VL [53], each configured to operate within the 1.5k token limit. For LlaVA-OneVision, we utilized the minimal visual token configuration (single visual features tile). InternVL2 was tested with both single-tile (256 tokens) and four-tile (1280 tokens) configurations, while Qwen2-VL was evaluated with 256 and 512 visual tokens, corresponding to image sizes of 448×448 and 616×616 respectively.

Under the practical 1.5k token constraint, incorporating DocVLM with each of these baseline models yields substantial and consistent improvements. Notably, these improvements persist even in looser token regimes, such as InternVL2 with 1280 visual tokens and Qwen2-VL with 576 visual tokens. Within this constraint, our Qwen2-VL variant with DocVLM, using just 576 tokens, achieves state-of-the-art performance across all benchmarks: 92.8% on DocVQA, 82.8% on TextVQA, 79.8% on ST-VQA, 66.8% on InfoVQA, and a CIDEr score of 150.4 on TextCAPS. As detailed in the Appendix, DocVLM achieves these results with significant computational advantages.

DocVLM also demonstrates exceptional capability in handling multipage documents. Using the same 576-token configuration described earlier, it achieves 84.5% accuracy on MP-DocVQA, surpassing previous state-of-the-art re-

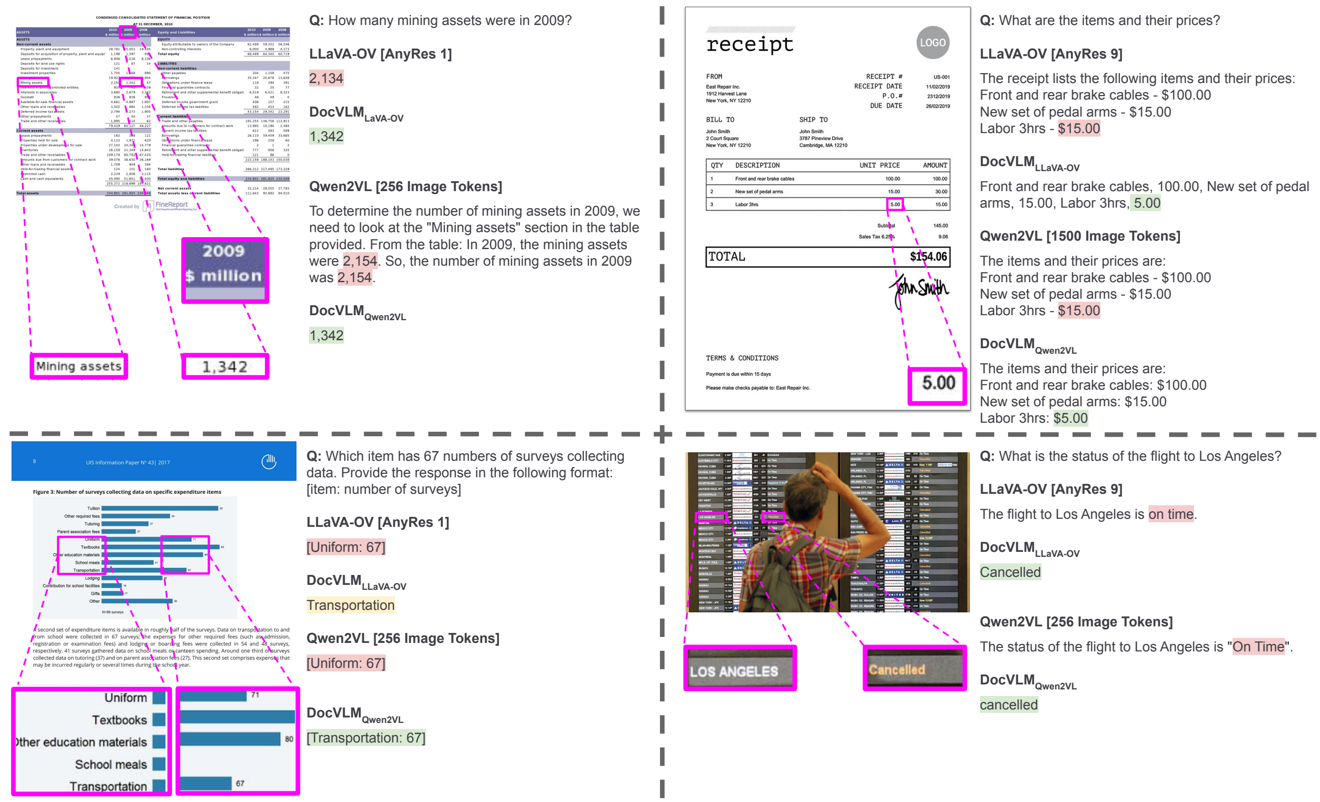

Figure 3. **Qualitative Results.** Representative examples of DocVLM's performance across diverse document formats, from dense text to infographics and scene text. Our model successfully handles complex layouts, dense content, and presents instruction-following capabilities without explicit training on such datasets. Each example includes an image-instruction pair with baseline and DocVLM predictions.

sults. Furthermore, this setup shows robust generalization capabilities with a zero-shot performance of 47.4% accuracy on the DUDE dataset, despite not being specifically trained for multipage document processing.

4.3. Qualitative Results

Figure 3 illustrates DocVLM's enhanced capabilities through representative examples that demonstrate three key strengths of our method: (1) improved reading comprehension in complex document layouts, (2) effective handling of dense textual content despite using compressed representations, and (3) preserved and enhanced instruction-following capabilities.

The examples span diverse document types, from dense text documents to infographics and scene text, showcasing DocVLM's versatility. In the infographic example, DocVLM not only preserves but enhances the base model's instruction-following capabilities, despite our OCR component not being explicitly trained on instruction-following datasets like [49]. This demonstrates that our compression mechanism successfully retains crucial textual and layout information while significantly reducing token usage.

4.4. Scaling to Multipage Documents

Building on the promising multipage results in Tab. 1, we conduct an in-depth analysis of different DocVLM configurations for multipage document understanding. This analysis focuses on the Qwen2-VL base model, tested on the MP-DocVQA dataset with documents up to 20 pages long – a scale that most other state-of-the-art methods struggle to handle due to token limitations.

Table 2 compares four OCR integration strategies used during inference in multipage scenarios:

- Baseline: vision-only input (no additional tokens)
- Direct OCR word insertion: up to 800 tokens per page
- Global OCR encoding: 64 tokens total
- Page-wise OCR encoding: 64 tokens per page

Results indicate consistent improvements over the baseline across all image resolutions (256, 512, and 1024 tokens), with only minimal additional token usage – just 64 tokens for the entire document in the global encoding case. Notably, at 256 visual tokens per page, both page-wise encoding (82.4%) and global encoding (81.7%) outperform direct OCR word insertion (79.1%) while using significantly fewer tokens.

Our best configuration achieves state-of-the-art perfor-

Method	LLM OCR Input	Image Tok.	OCR Tok.	ANLS
DocOwl2 [27]	–	324 × pg	–	69.4
GRAM$_{\text{C-Former}}$ [14]	–	100 × pg	256	77.6
GRAM [14]	–	100 × pg	800 × pg	80.3
Qwen2-VL	–	256 × pg	–	73.0
	OCR Words	256 × pg	800 × pg	79.1
DocVLM$_{\text{Qwen2-VL}}$	Global Encoding	256 × pg	64	81.7
DocVLM$_{\text{Qwen2-VL}}$	Page-wise Encoding	256 × pg	64 × pg	82.4
Qwen2-VL	–	512 × pg	–	82.1
DocVLM$_{\text{Qwen2-VL}}$	Global Encoding	512 × pg	64	84.5
DocVLM$_{\text{Qwen2-VL}}$	Page-wise Encoding	512 × pg	64 × pg	<u>85.2</u>
Qwen2-VL	–	1024 × pg	–	<u>85.2</u>
DocVLM$_{\text{Qwen2-VL}}$	Global Encoding	1024 × pg	64	**86.3**
DocVLM$_{\text{Qwen2-VL}}$	Page-wise Encoding	1024 × pg	64 × pg	**86.3**

Table 2. **Extension to Multipage.** Comparison on MP-DocVQA of approaches for incorporating OCR information in multipage document understanding. Both DocVLM multipage extension strategies: global encoding (64 tokens per document) and page-wise encoding (64 tokens per page), outperform previous state-of-the-art methods, notably, without any explicit multipage training.

mance of 86.3% ANLS, significantly outperforming specialized multipage models like GRAM (80.3%). This is particularly impressive considering that DocVLM was trained exclusively on single-page inputs, demonstrating strong zero-shot generalization to multipage scenarios.

The comparison between encoding strategies reveals that page-wise encoding consistently outperforms global encoding at lower visual token counts, providing a +0.7% improvement for both 256 and 512 image tokens per page. This advantage diminishes at 1024 tokens where both achieve identical performance (86.3% ANLS). Remarkably, DocVLM with page-wise encoding matches or even outperforms the baseline Qwen2-VL using twice as many visual tokens, highlighting the efficiency of our approach.

5. Ablation Study

Impact of OCR Encoding Strategies To evaluate the impact of OCR encoding compression, we compare three strategies for integrating OCR information: (1) inserting raw OCR words in the original VLM, (2) using DocVLM uncompressed OCR encodings, and (3) DocVLM compressed OCR encodings with 64 learned queries. We evaluate these approaches on the DocVQA test set using three representative model configurations: LLaVA-OneVision with 1.5K visual tokens, and both InternVL2 and Qwen2-VL with 256 visual tokens each.

Results in Tab. 3 demonstrate that DocVLM's OCR encodings significantly outperform raw OCR words across all three models while maintaining the same token count. Notably, our compressed encoding approach, using just 64 tokens instead of 800 OCR tokens, preserves most of these improvements while drastically reducing sequence length. This efficient compression enables a more favorable allocation of the token budget, allowing models to dedicate more tokens to visual processing without compromising OCR effectiveness. The results validate that DocVLM's compression strategy successfully balances performance with computational efficiency, a key factor for practical applications.

LLM OCR Input	OCR Tok.	LLaVA-OV 1.5K	InternVL2 256	Qwen2-VL 256
OCR Words	800	85.8	84.4	89.1
OCR Encoding	800	89.4	89.2	91.9
64 Compressed Encoding	64	88.4	86.6	91.2

Table 3. **OCR Encoding Strategies.** DocVQA results for inserting OCR information using (1) OCR words (baseline), (2) uncompressed OCR encodings, and (3) 64 compressed OCR encodings.

Balancing Vision and OCR Token Allocation Modern VLMs employ various mechanisms to reduce visual token count, creating an inherent trade-off between computational efficiency and model performance, as discussed in Sec. 3.1. We investigate how DocVLM can improve this trade-off by comparing four configurations: (1) baseline VLM without OCR, (2) direct OCR word insertion, (3) DocVLM with uncompressed OCR encodings, and (4) DocVLM with 64 compressed learned queries.

Figure 4 presents the performance scores on DocVQA validation (left y-axis) and total token counts (right y-axis) for three VLM architectures, showing how these metrics vary across different visual token allocations. Each model employs a distinct token reduction approach: LLaVA-OneVision controls token count through feature downsampling (AnyRes Max), InternVL2 limits the number of processed image tiles (Dynamic Max Batch), and Qwen2-VL adjusts image resolution to constrain token count.

Our analysis reveals that integrating OCR information, regardless of the method used, consistently improves performance across all models, with particularly pronounced gains in low visual token regimes. However, uncompressed OCR integration methods, whether through direct word insertion or uncompressed DocVLM encodings, require 800 tokens – a significant overhead that could otherwise be allocated to visual processing. For instance, allocating 128 tokens for visual and 800 for OCR in Qwen2-VL achieves 84.3% using OCR words and 90.1% using uncompressed encodings. In contrast, using 896 pure visual tokens reaches 92.4%, demonstrating the potential benefit of allocating more tokens to visual processing. DocVLM's compression mechanism provides a superior option by requiring only 64 tokens for OCR information while maintaining strong performance. In the above example, our approach using fewer tokens, allocating 768 visual tokens and 64 OCR tokens, reaches 93.0%, outperforming the 90.1% obtained with the uncompressed encodings, highlighting DocVLM's effective balance between visual and OCR tokens.

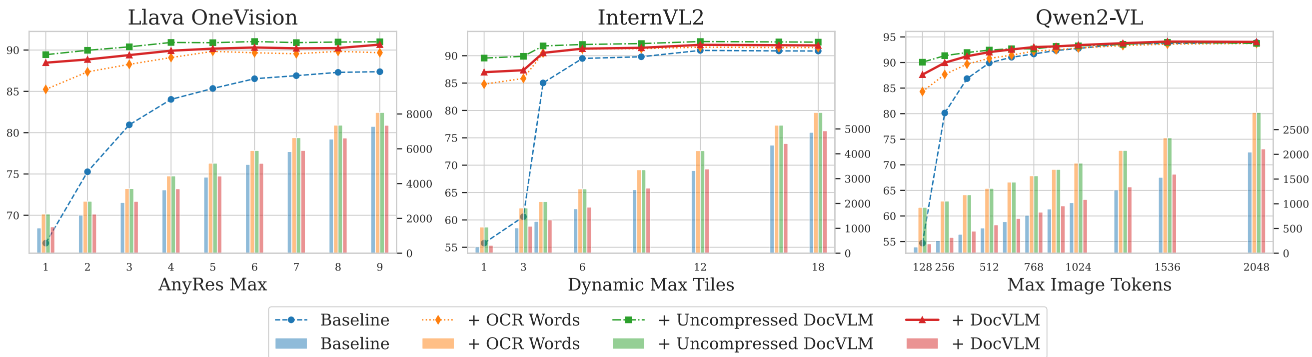

Figure 4. **Balancing Performance and Compute.** Analysis of model performance (lines, left y-axis) and token usage (bars, right y-axis) as a function of visual token allocation. Each model employs its inherent token control strategy: AnyRes max for feature downsampling (LLaVA One-Vision), dynamic max tiles (InternVL2), and max image tokens for resolution control (Qwen2-VL). The results highlight that DocVLM consistently improves performance with minimal overhead (64 tokens), offering an efficient OCR-visual token allocation.

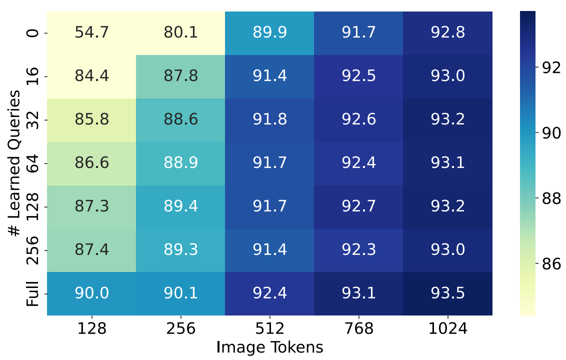

Figure 5. **Compression Levels.** DocVQA validation results for DocVLM integrated with Qwen2-VL across varying OCR and image token budgets. "0" represents the baseline, while "Full" indicates uncompressed encodings.

Compression Levels We deepen our analysis of the OCR-visual token trade-off by examining different compression levels in DocVLM integrated with Qwen2-VL. Fig. 5 presents ANLS scores across various combinations of visual tokens (128-1024) and learned queries (16-256), including baselines without OCR and with uncompressed encoding. In low visual token regimes, increasing the number of learned queries yields substantial improvements, validating our compression mechanism's effectiveness in capturing relevant OCR information. Notably, even with just 16 learned queries, DocVLM outperforms the baseline across all visual token configurations, offering strong performance with minimal computational overhead.

Training Stages Tab. 4 illustrates the impact of the vision alignment stage on ANLS performance for DocVLM with the LLaVA-OneVision base model on the DocVQA test set, showing results for varying numbers of learned queries (16 to 128) and the uncompressed case. Our two-stage training process initially trains OCR modality components without image input, forcing reliance on OCR data alone, before reintroducing the image in the vision alignment stage to adapt learned queries alongside visual information. The results reveal that vision alignment significantly boosts performance, especially with fewer learned queries: for instance, with 16 learned queries, there's an improvement of +6.2, compared to +0.7 in the uncompressed case. Notably, after vision alignment, DocVLM with only 16 learned queries outperforms the baseline of OCR words (from Table 3). These findings underscore the effectiveness of our two-stage training method.

Training Phases	Compressed Enc.			OCR Enc.
	16	64	128	800
Stage I: OCR-LLM Alignment	81.7	85.8	86.3	89.4
+ Stage II: Vision Alignment	**87.9**	**88.4**	**88.4**	**90.1**
Δ	**+6.2**	**+2.6**	**+2.1**	**+0.7**

Table 4. **Training Phases.** Second-stage training consistently improves DocVQA performance, both with compressed DocVLM tokens and full OCR encoding.

6. Conclusions

Our results demonstrate that DocVLM can be effectively integrated into various VLMs to enhance their document reading capabilities while significantly reducing their dependency on extensive vision tokens. The key takeaway is that in token-constrained scenarios, allocating a small portion of tokens to OCR information consistently yields better results than than using those tokens solely for visual processing. The effectiveness of our compression mechanism extends beyond single-page documents, as evidenced by achieving state-of-the-art results on MP-DocVQA using the same 64 tokens to represent multiple pages. These results establish DocVLM as a practical solution for enhancing document understanding in real-world applications where computational efficiency is crucial.

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. 2

[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.

[4] Ofir Abramovich, Niv Nayman, Sharon Fogel, Inbal Lavi, Ron Litman, Shahar Tsiper, Royee Tichauer, Srikar Appalaraju, Shai Mazor, and R Manmatha. Visfocus: Prompt-guided vision encoders for ocr-free dense document understanding. In *European Conference on Computer Vision*, pages 241–259. Springer, 2025. 2

[5] 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. 2

[6] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 993–1003, 2021. 3

[7] Srikar Appalaraju, Peng Tang, Qi Dong, Nishant Sankaran, Yichu Zhou, and R Manmatha. Docformerv2: Local features for document understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 709–718, 2024. 3, 5, 12

[8] Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Cărbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. Screenai: A vision-language model for ui and infographics understanding. *arXiv preprint arXiv:2402.04615*, 2024. 1, 5

[9] 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. *arXiv preprint arXiv:2308.12966*, 1(2):3, 2023. 2

[10] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 1, 5

[11] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301, 2019. 4, 12

[12] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R Manmatha. Latr: Layout-aware transformer for scene-text vqa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16548–16558, 2022. 2, 3

[13] Ali Furkan Biten, Ruben Tito, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. Ocr-idl: Ocr annotations for industry document library dataset. *arXiv preprint arXiv:2202.12985*, 2022. 12

[14] 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. 2, 3, 5, 7

[15] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 4, 12

[16] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023. 2

[17] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023. 1, 2

[18] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. Internvl2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy, 2024. 2, 3, 4, 5

[19] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Albert Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *ArXiv*, abs/2305.06500, 2023. 2

[20] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, et al. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *arXiv preprint arXiv:2404.06512*, 2024. 1

[21] Masato Fujitake. Dtrocr: Decoder-only transformer for optical character recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8025–8035, 2024. 2

[22] 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

[23] Roy Ganz, Yair Kittenplon, Aviad Aberdam, Elad Ben Avraham, Oren Nuriel, Shai Mazor, and Ron Litman. Question aware vision transformer for multimodal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision*

and Pattern Recognition (CVPR), pages 13861–13871, 2024. 2

[24] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4, 12

[25] Jean-Philippe Thiran Guillaume Jaume, Hazim Kemal Ekenel. Funsd: A dataset for form understanding in noisy scanned documents. In *Accepted to ICDAR-OST*, 2019. 1

[26] 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. 3

[27] Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. *arXiv preprint arXiv:2409.03420*, 2024. 2, 7

[28] Taeho Kil, Seonghyeon Kim, Sukmin Seo, Yoonsik Kim, and Daehee Kim. Towards unified scene text spotting based on sequence generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15223–15232, 2023. 2

[29] Geewook Kim, Teakgyu Hong, Moonbin Yim, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Donut: Document understanding transformer without ocr. *arXiv preprint arXiv:2111.15664*, 7(15):2, 2021. 2

[30] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024. 2

[31] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1, 2, 4, 5

[32] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 2

[33] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *arXiv preprint arXiv:2407.02392*, 2024. 2

[34] 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. 2

[35] 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. 2, 3

[36] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 2

[37] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: On the hidden mystery of ocr in large multimodal models, 2024. 1

[38] Jiahao Lyu, Wei Wang, Dongbao Yang, Jinwen Zhong, and Yu Zhou. Arbitrary reading order scene text spotter with local semantics guidance. *arXiv preprint arXiv:2412.10159*, 2024. 14

[39] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 1, 4, 12

[40] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 4, 12, 13

[41] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022. 1, 4, 12, 13

[42] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE, 2019. 4, 12

[43] 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. 2

[44] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 2

[45] 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. *Journal of machine learning research*, 21(140):1–67, 2020. 3

[46] Roi Ronen, Shahar Tsiper, Oron Anschel, Inbal Lavi, Amir Markovitz, and R Manmatha. Glass: Global to local attention for scene-text spotting. *arXiv preprint arXiv:2208.03364*, 2022. 2

[47] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 742–758. Springer, 2020. 4, 12

[48] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 4, 12

[49] Ryota Tanaka, Taichi Iki, Kyosuke Nishida, Kuniko Saito, and Jun Suzuki. Instructdoc: A dataset for zero-shot generalization of visual document understanding with instructions.

In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 19071–19079, 2024. 6

[50] Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multipage docvqa. *Pattern Recognition*, 144:109834, 2023. 1, 2, 4, 12

[51] Jordy Van Landeghem, Rubèn Tito, Łukasz Borchmann, Michał Pietruszka, Pawel Joziak, Rafal Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Anckaert, Ernest Valveny, et al. Document understanding dataset and evaluation (dude). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19528–19540, 2023. 2, 4, 12

[52] Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. Docllm: A layout-aware generative language model for multimodal document understanding. *arXiv preprint arXiv:2401.00908*, 2023. 2, 13

[53] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 1, 2, 3, 4, 5

[54] Wenjin Wang, Yunhao Li, Yixin Ou, and Yin Zhang. Layout and task aware instruction prompt for zero-shot document image question answering. *arXiv preprint arXiv:2306.00526*, 2023. 1, 2, 13

[55] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1192–1200, 2020. 13

[56] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *arXiv preprint arXiv:2310.05126*, 2023. 2

[57] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Bo Du, and Dacheng Tao. Dptext-detr: Towards better scene text detection with dynamic points in transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3241–3249, 2023. 2

[58] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Tongliang Liu, Bo Du, and Dacheng Tao. Deepsolo: Let transformer decoder with explicit points solo for text spotting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19348–19357, 2023. 2

[59] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 2

[60] Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, Haodong Duan, Bin Wang, Linke Ouyang, et al. Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv preprint arXiv:2407.03320*, 2024. 1

[61] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 2

[62] Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat-Seng Chua. Towards complex document understanding by discrete reasoning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4857–4866, 2022. 4, 12