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:

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

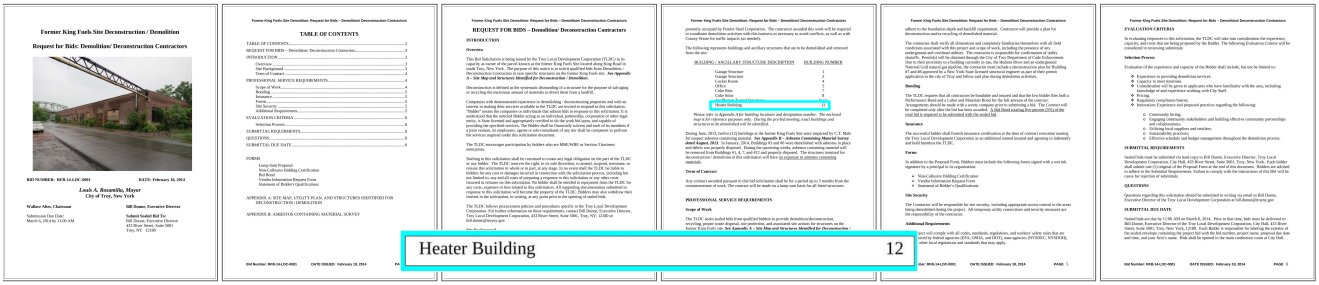
OCR-Mask Matching: In this objective, we compute $\mathbf{R}_q^{(\text{out})}$, $\mathbf{R}_m^{(\text{out})}$, and $\mathbf{S}_{qt}$ similarly to OCR-Mask Matching. We employ a hard negative mining strategy [32] to select challenging negative examples based on the similarity values in the $\mathbf{S}_{qt}$ 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.

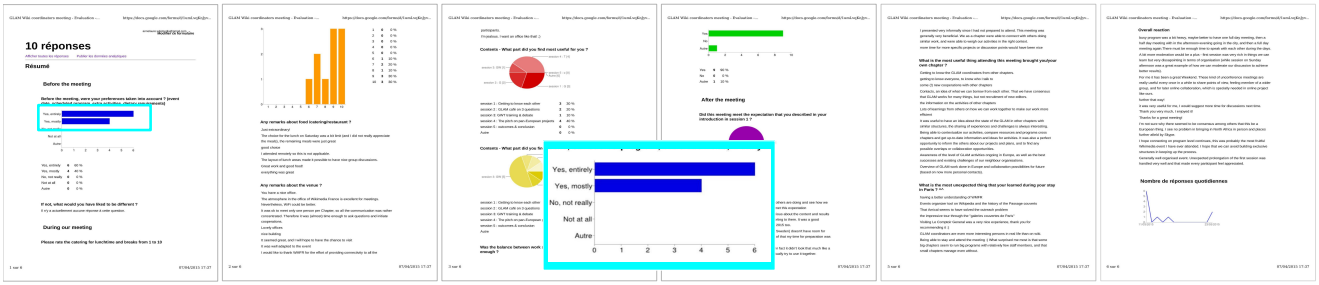
Q: What is the building number of the heater building?



LLaVA: 'NA'

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

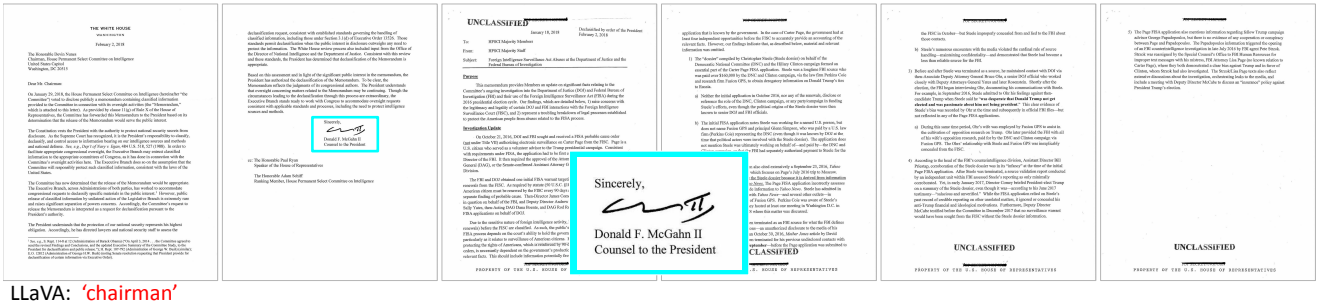
Q: What is the difference between the number of people who answered 'Yes, entirely' and 'Yes, mostly' in the chart on the first page of the document?



LLaVA: '1000000000'

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

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

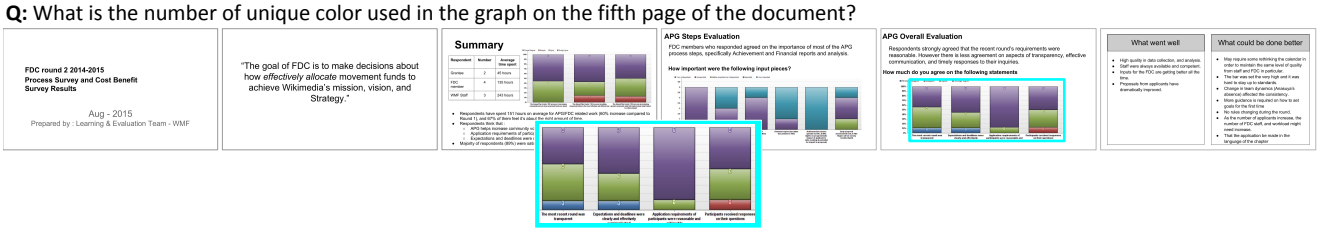


LLaVA: 'chairman'

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.

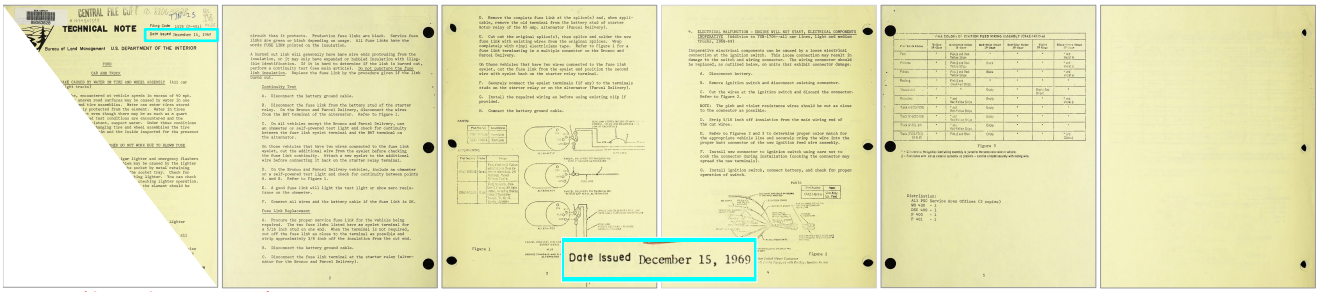
Q: What is the number of unique color used in the graph on the fifth page of the document?



LLaVA: '5'

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

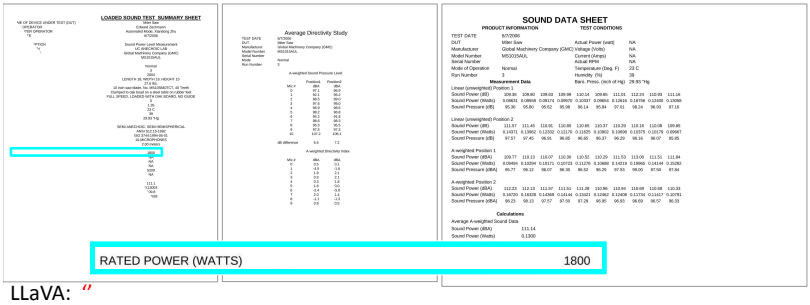
Q: When did this document published?



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.