Question Aware Vision Transformer for Multimodal Reasoning

Roy Ganz*
Technion, Israel
ganz@cs.technion.ac.il

Yair Kittenplon†
AWS AI Labs
yairk@amazon.com

Aviad Aberdam
AWS AI Labs
aaberdam@amazon.com

Elad Ben Avraham
AWS AI Labs
eladba@amazon.com

Oren Nuriel
AWS AI Labs
onuriel@amazon.com

Shai Mazor
AWS AI Labs
smazor@amazon.com

Ron Litman†
AWS AI Labs
litmanr@amazon.com

Abstract

Vision-Language (VL) models have gained significant research focus, enabling remarkable advances in multimodal reasoning. These architectures typically comprise a vision encoder, a Large Language Model (LLM), and a projection module that aligns visual features with the LLM's representation space. Despite their success, a critical limitation persists: the vision encoding process remains decoupled from user queries, often in the form of image-related questions. Consequently, the resulting visual features may not be optimally attuned to the query-specific elements of the image. To address this, we introduce QA-ViT, a Question Aware Vision Transformer approach for multimodal reasoning, which embeds question awareness directly within the vision encoder. This integration results in dynamic visual features focusing on relevant image aspects to the posed question. QA-ViT is model-agnostic and can be incorporated efficiently into any VL architecture. Extensive experiments demonstrate the effectiveness of applying our method to various multimodal architectures, leading to consistent improvement across diverse tasks and showcasing its potential for enhancing visual and scene-text understanding.

1. Introduction

In recent years, VL architectures have emerged as a pivotal research area, leading to significant progress in the domain of multimodal reasoning [3, 15, 19, 20, 24, 30, 31, 34, 43, 54]. Such architectures fundamentally seek to bridge the gap between visual and textual data, enabling models to interpret, comprehend, and generate content based on both visual and textual information. This fusion of modalities has diverse applications and tasks, from image captioning (CAP) [10, 45] and visual question answering

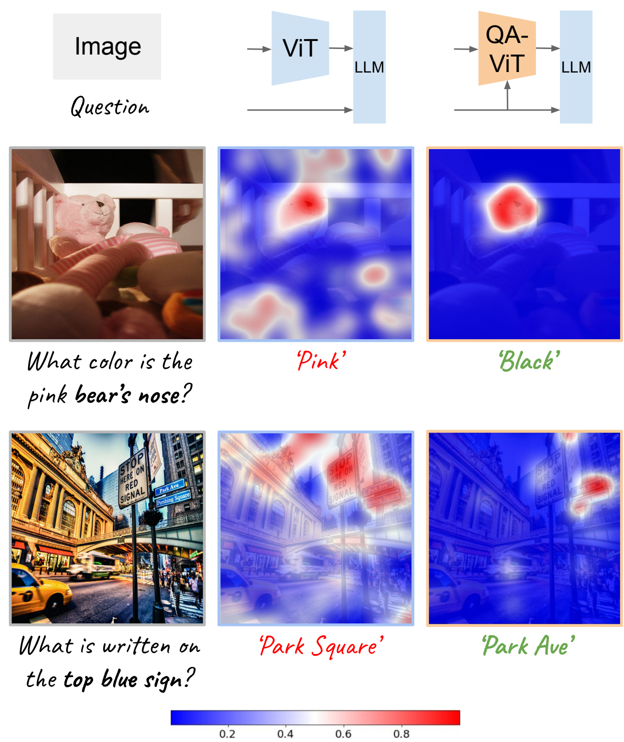

Figure 1. **Question-Aware Vision Encoding.** Comparative illustrations for VQAv2 (upper) and TextVQA (lower) predictions of ViT+T5 and QA-ViT+T5 VL models. Employing GradCAM highlights the focus areas with respect to key terms in the posed questions. This vividly demonstrates the motivation behind QA-ViT: enhancing ViT with the question enables it to focus on the relevant image aspects, resulting in more accurate predictions.

(VQA) [4, 46] to tasks in autonomous robotics and human-computer interactions. As the list of applications continues to grow, the role of VL architectures becomes increasingly crucial within the broader field of deep learning.

At the heart of multimodal VL architectures lies the concept of vision-language Modeling. These models typically

*Work done during an Amazon internship.
†Corresponding author.

consist of three essential steps. First, a unimodal vision architecture extracts meaningful information from images. Typically, the vision encoder is a frozen Vision-Transformer (ViT), often based on CLIP [17, 41]. Second, a projection module bridges the gap between vision and language, transforming visual features into ones that can be comprehended and processed by a language model. This module is usually either a simple linear layer or MLP [33, 34, 54], or a cross-attention-based transformer architecture [6, 15, 31]. Lastly, the projected visual information and the textual instruction, commonly in the form of questions or prompts, are inserted into a Large Language Model (LLM) to complete the task.

Despite the remarkable progress achieved in VL research, we have identified an intriguing yet often overlooked limitation within such architectures. The success of such a model hinges on its ability to not only comprehend the visual content but also to do so through the lens of the accompanying textual instruction, *e.g.*, the provided question, often requiring focus on fine-grained details inside the entire image. Existing architectures, however, are suboptimal in this aspect, as they perform the vision encoding unaware of the posed question, resulting in visual features not optimally aligned with the user query. As the vision encoder outputs a fixed size features sequence F_V, it is limited in the level of information encoded in them. Due to the relatively high abstraction level, it is likely to disregard or overlook low-level details in the image. This oversight becomes particularly problematic in scenarios where nuanced image understanding is essential to accurately respond to queries. Thus, we claim that the vision encoder $\mathcal{V}$ should be cast from a single input function into a conditional function. Namely, $\mathcal{V}(I|Q)$ instead of $\mathcal{V}(I)$, where I, Q are the image and question, respectively.

To mitigate this limitation and yield a textual conditioned vision encoding, we present **QA-ViT**, Question Aware Vision Transformer for multimodal reasoning. The intuition of our method is clear: if the model understands the posed question and the inherent context, it can extract visual features that directly correspond to the relevant image aspects essential for answering it correctly. We illustrate this behavior in Fig. 1; By applying GradCAM [44] to both vanilla CLIP-based ViT and QA-ViT, w.r.t. textual prompts correspond with a distinct spatial location. While the baseline tends to favor high abstraction level features, even when prompted with region-specific descriptions, QA-ViT focuses significantly more on the relevant image parts. For instance, considering the bottom image and the question like "What is written on the top blue sign?", we can see that while the baseline vision encoder generates features that contain a wealth of information about the scene (*e.g.*, the buildings, cars, and people), QA-ViT is able to pinpoint the specific region of interest, namely, the blue sign. Our approach achieves the above goal by directly integrating textual representations into any vision encoder while keeping most of it frozen, preserving its visual understanding capabilities (Fig. 2). In practice, we utilize the preexisting self-attention mechanism in the ViT to also attend to textual encodings, representing the user query.

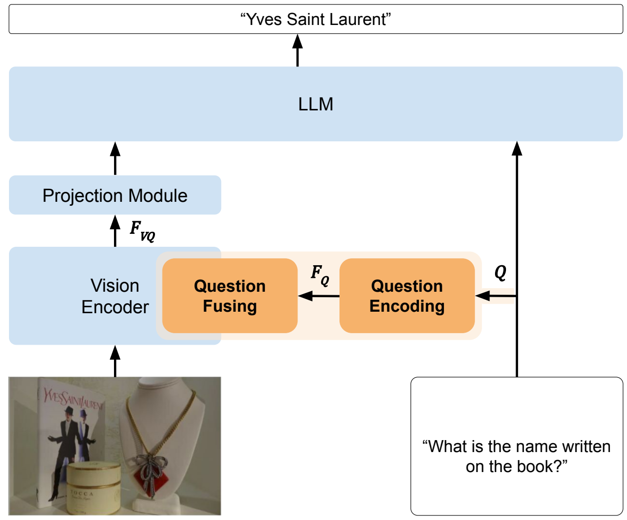

Figure 2. **Method overview.** A high-level illustration of the QA-ViT (highlighted in orange) incorporated into a general VL architecture (depicted in blue). This is achieved by encoding the question Q into features F_Q, which are fused into the vision encoder, resulting in question-aware visual features F_{VQ}.

To demonstrate QA-ViT effectiveness, we leverage the model-agnostic nature of our method and integrate it into top-performing systems, including BLIP2 [31], InstructBLIP [15], and LLaVA-1.5 [33]. In addition, we also integrate QA-ViT into a simple ViT+T5 architecture, without pretraining, to demonstrate its benefit when training an unaligned VL system from scratch. We train all these architectures on a combined dataset of visual question answering and image captioning, requiring visual and Optical Character Recognition (OCR) understanding, and evaluate them accordingly. Despite the architectural differences between the considered VL models in the vision-encoder, projection module (QFormer vs. MLP), and LLM structure (encoder-decoder vs. decoder only), extensive experiments show that QA-ViT consistently improves the performance over all the tested models and benchmarks, attesting to its versatility.

To summarize:

- We identify an overlooked suboptimality in the paradigm of vision-language modeling stemming from the lack of instruction-aware image encoding.
- We introduce QA-ViT, a model-agnostic method that enables existing vision encoders to be conditioned on textual prompts or questions.
- Thorough experiments on multiple architectures demonstrate our method's ability to enhance multimodal reasoning, improving the performance on various benchmarks.

2. Related Work

Vision-Language Models. Earlier-generation VL models pursue the paradigm of rigorous and extensive pretraining, using contrastive losses, followed by designated fine-tuning for specific tasks [28–30, 50–52]. While this approach constituted a critical milestone, it led to specialist models that only perform well on a specific downstream task [8, 20, 46]. By leveraging the capabilities of recent Large Language Models (LLMs) [14, 47–49], current top-performing VL models are generalist models, showcasing remarkable performance across various VL tasks. Interestingly, such models demonstrate strong zero-shot performance and generalization to unseen data and tasks [3, 6, 12, 15, 31, 33], and sometimes even surpassing specialist models.

Architecturally, there are two main types of VL models, which mainly differ in the integration mechanism of the visual features into the LLM. The first type projects the visual features using a cross-attention-based transformer model (*e.g.*, QFormer), which also reduces the visual sequence length [6, 15, 31]. The introduction of such a mechanism enables keeping both the LLM and the vision encoder frozen. The second line of research demonstrates that the projection module can be simplified to a linear projection (or an MLP) while also training the LLM [12, 33, 34, 54]. Despite such differences, all current top-performing VL models perform image encoding in an unaware manner to the given textual prompt.

Question-Aware Vision Encoding. A possible solution for the limitation above was proposed in the OCR-free text-oriented multimodal understanding by pix2struct [27], which suggests directly rendering the question as a header at the top of the original image instead of passing it to the LLM. However, this approach relies highly on their OCR-oriented pretraining and is suboptimal in the general VL case. Another step towards instruction-aware visual features is InstructBlip [15], which introduces the visual features into the QFormer alongside the instruction. Nevertheless, it operates solely on top of the outputs of the vision encoder and, thus, is incapable of compensating for overlooked image aspects. In this paper, we propose to integrate question information into any ViT-based image encoder in a flexible and modular manner.

3. Method

Our method proposes a versatile and lightweight model-agnostic approach, which can be integrated into any vision transformer model in any VL architecture, designed to transform trained image encoders into question-aware ones effectively. Formally, given the image and question I, Q, we argue that the vision encoding module $\mathcal{V}$ should be casted into a conditioned one:

$$F_V = \mathcal{V}(I) \rightarrow F_{VQ} = \mathcal{V}(I|Q). \tag{1}$$

In this section, we first describe our high-level design and then delve into the details of each building block.

3.1. Overall Architecture

As illustrated in Fig. 2, our method comprises two fundamental components. First, the question, denoted as Q, is fed into a "**Question Encoding**" module, which processes and projects the textual prompt, bridging the gap between the linguistic and visual features domains. Subsequently, the textual encoded features, denoted as F_Q, are integrated inside a frozen vision model via "**Question Fusing**" module, producing text-aware visual features F_{VQ}. Lastly, the F_{VQ} is projected by the projection module, concatenated with the instruction embeddings, and fed into the LLM, which processes and produces the overall system's output. In general, QA-ViT modifies solely the vision encoder, maintaining the rest of the architecture intact.

3.2. Question Encoding

In order to introduce text prompts Q into an unimodal vision transformer, we propose a streamlined two-stage process.

Question Representation. First, we encode the natural language prompt (*e.g.*, the question) into meaningful representations, denoted as F'_Q. Formally, we define this operation as $\mathcal{E}(Q) = F'_Q$, where $\mathcal{E}$ represents the encoding function. This step introduces flexibility in choosing $\mathcal{E}$, the source of these textual representations – the preexisting LLM's encoder or embeddings or a designated language model. We mainly focus on the former as it offers more parameter efficiency and can lead to more seamless integration, as the same LLM subsequently processes the visual features. We compare these approaches in Sec. 5.1.

Representation Projection. Second, we utilize MLPs to project the textual representations into the vision model features space. Due to the vision model's hierarchical structure, different layers have different abstraction levels [17, 42]. Hence, we adopt a per-layer MLP to obtain better alignment. We denote the projected textual representation for layer i as F^i_Q. Overall, the question encoding phase operates as follows:

$$F^i_Q = \text{MLP}^i(\mathcal{E}(Q)). \tag{2}$$

For simplicity, we omit the layer index from now on.

3.3. Question Fusing

Given the projected textual representations F_Q, we propose a parameter-efficient fusing mechanism to integrate them into frozen ViT architectures in a model-agnostic way. Keeping the vision encoder frozen enables text-conditioned encoding of the image while preserving the model's original capabilities intact. While such integration can be done in various ways, we propose a straightforward approach that harnesses the ViT preexisting self-attention mechanism, illustrated in Fig. 3.

Fusing Mechanism. We extend the input sequence of the self-attention layer to contain the projected representations $F_Q \in \mathbb{R}^{K \times C}$ by concatenating it with the visual representations $F_V \in \mathbb{R}^{M \times C}$, where C is the channel dimension. This yields a sequence of length $K + M$, containing vision and question information. Next, the frozen self-attention mechanism is applied to produce the attention scores and outputs while also attending to the textual information F_Q, enabling cross-modal attention. We select the attention output that corresponds with the input visual representations, resulting in $F'_{VQ} \in \mathbb{R}^{M \times C}$. More formally,

$$F'_{VQ} = \text{Attention}(\text{concat}(F_V, F_Q))_{[0:\text{M}]}. \tag{3}$$

An additional projection followed by a learnable gating mechanism [2, 3, 20, 22] is introduced in parallel to the existing frozen projection head. This module compensates for the distribution shift from incorporating question information in the frozen self-attention layer. The goal of such a gating is to enable the gradual blending of the residual projected information with the existing one, avoiding a significant feature modification and a degradation of the overall performance. Such gating is done by multiplying the additional projection layer's outputs with $\tanh(\beta)$, where β is a learnable parameter initialized to zero. This technique is designed to maintain the layer's outputs with minimal deviation at initialization, improving stability while enabling a residual learnable stream of information. Mathematically, our fusing mechanism functions as follows:

$$F_{VQ} = \mathcal{P}(F'_{VQ}) + \mathcal{P}_g(F'_{VQ}) \cdot \tanh(\beta). \tag{4}$$

Integration Point. An important design choice in our fusing mechanism is the choice of the integration point of the textual representations into the vision transformer layers. Specifically, we perform `late fusion`, namely, applying the fusing in the top L self-attention layers of the N-layered ViT, where $L < N$. This choice is motivated by the nature of ViT layers hierarchy – lower layers primarily capture low-level visual details, while the higher layers mainly focus on high-level concepts [17, 42]. Therefore, the likelihood of disregarding fine-grained details is expected to emerge in the higher layers, making them an optimal target for our method. We validate this choice in Sec. 5.

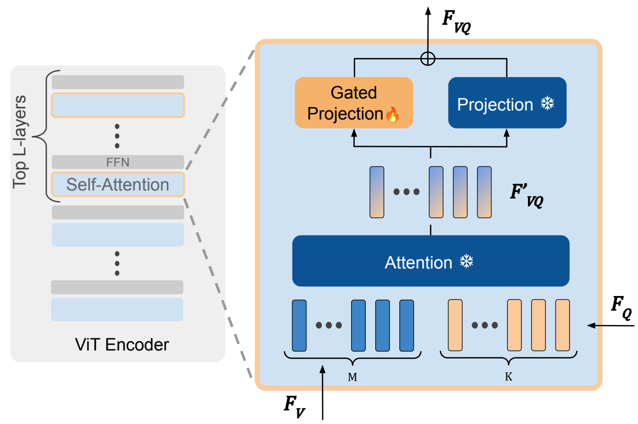

Figure 3. **Textual representations fusing.** Left: General scheme of the ViT encoder. Right: Zoom in to our fusing mechanism in one of the top-L self-attention layers. The M visual features from the previous layer F_V, are concatenated with K textual features F_Q and fed into the frozen self-attention mechanism to obtain M text-attended visual representations F'_{VQ}. Next, a parallel gated projection obtains the question-aware visual features of F_{VQ}.

4. Experiments

We conduct a comprehensive set of experiments to assess the capabilities of QA-ViT. Given the model-agnostic nature of our method, which enables seamless integration into any existing VL architecture, our experiments are designed to showcase its versatility in two distinct architectural settings. In the first setting, we experiment with a straightforward VL approach consisting of a vision encoder and encoder-decoder-based LLM, denoted as ViT+T5. The second setting involves integrating our method into already trained top-performing vision-language models, specifically LLAVA-1.5 [33], BLIP2 [31], and instructBLIP [15]. This allows us to assess the benefits of QA-ViT for already finetuned models. In both settings, we train and evaluate the models using a combined dataset of visual question answering and image captioning, requiring both visual and OCR understanding [1, 2, 32]. In the OCR case, we are interested in the **OCR-free** setting; we do not equip the models with OCR tokens.

4.1. Training Data

For training across all considered architectures, we adopt a multi-task approach using concatenated VL datasets that involve reasoning over both visual and OCR information. In particular, we consider general visual question-answering datasets [21, 25] alongside scene-text [8, 40, 46] and document-oriented ones [37–39]. For these datasets, We insert the question representations into the vision encoder when applying QA-ViT. In addition, we include captioning datasets (COCO Captions [11] and TextCaps [45]),

Figure 4. **Paying attention to details in visual question answering.** Representative examples require answering questions regarding subtle or less conspicuous image details (zoomed-in) from VQAv2 and TextVQA datasets. Each sample includes an image-question pair alongside predictions from ViT+T5 and QA-ViT+T5, where green indicates correct predictions and red indicates incorrect ones.

which leads to additional improvements, as can be seen in Sec. 5.2). In the captioning data, we utilize a random template instruction, as in [15], *e.g., "Please provide a short depiction of the picture"* and insert them into the ViT. We provide the complete list of such templates in the supplementary materials, alongside further details on the training dataset composition. Overall, our dataset comprises approximately 3 million assets from multiple training datasets of different sizes. We adopt a sampling strategy proportional to each dataset's size during training to address the size disparity. This approach is designed to prevent overfitting smaller datasets and underfitting larger ones.

4.2. QA-ViT Performance Gains

We evaluate QA-ViT on general (VQA$^{\text{v2}}$ and COCO) and scene-text (VQA$^{\text{T}}$, VQA$^{\text{ST}}$ and TextCaps) benchmarks, in addition to zero-shot setting (VizWiz [7]). Additionally, we calculate average scores by assigning equal weight to both visual question answering and image captioning tasks.

ViT+T5 First, we examine a simple yet effective approach – a frozen CLIP[1] [41] and Flan-T5 [14] of different sizes (`base`, `large`, and `xl`), with an MLP projection module. We train the system on the data described in Sec. 4.1, using both the standard CLIP-ViT and QA-ViT, with the same training hyperparameters. In particular, we adapt the LLM weights using LoRa [23], train the projection MLP, and, in the QA-ViT case, also the instruction fusing counterparts. Both the baseline and the QA-ViT settings exhibit high parameter efficiency, keeping the vast majority of the weights frozen. We report the quantitative results of the ViT+T5 and compare them with QA-ViT in Table 1. As can be seen, QA-ViT leads to a substantial and consistent improvement compared to the baseline in all the benchmarks and across all model sizes. Moreover, our method not only improves performance on the seen benchmarks, but it also benefits it in a zero-shot setting on VizWiz [7].

To better understand the gains achieved by QA-ViT, we provide qualitative results in the ViT+T5-large model in Fig. 4. As seen, QA-ViT leads to better performance, specifically on image-question pairs that require reasoning over nuanced low-level details inside the image. For example, the image-question pair on the right requires focusing on the board, which is relatively small and marginal in importance compared to the entire image. Similar behavior is observed throughout all such examples.

State-of-the-art Models After validating the efficacy of QA-ViT in a pretraining-free setting, we turn to experiment with already-trained leading VL models. In this setting, we finetune the base model with and without QA-ViT using our training data introduced in Sec. 4.1. As in the ViT+T5 case, we employ a similar training setting by applying LoRa to the LLM and tuning the projection model and the QA-ViT components, if applicable. Specifically, we consider BLIP2 [31], InstructBLIP [15], using different sizes, and LLaVA-1.5 [33], top-performing multimodal architectures, and report the results in Tab. 1. As can be seen, QA-ViT consistently improves the baselines in all the tested architectures and across all the seen benchmarks while showing benefit also in the unseen one (except in InstructBLIP).

4.3. QA-ViT Results Analysis

We turn to conduct a more in-depth analysis of the results provided in Tab. 1 to better understand the contributions of QA-ViT. Our method improves the performance of different architectures, highlighting the three-way model agnosticism of QA-ViT in terms of the vision encoder, projection module, and LLM.

- **Vision Encoder** – Despite BLIP2 and InstructBLIP utilizes a different vision encoder than LLaVA-1.5 (39-layered EVA-CLIP [18] with a resolution of 224×224 vs. a 24-layered CLIP ViT-L of 336×336 resolution), integrating QA-ViT leads to improved performance.
- **Projection Module** – On the one hand, BLIP2 and InstructBLIP use a QFormer, a transformer-based architecture with learnable tokens, that also reduces the sequence

[1] https://huggingface.co/openai/clip-vit-large-patch14-336

Method	LLM	General		Scene-Text			0-shot	Average	
		VQA^{v2} vqa-score	COCO CIDEr	VQA^{T} vqa-score	VQA^{ST} ANLS	TextCaps CIDEr	VizWiz vqa-score	General	Scene-Text
ViT+T5-base	Flan-T5-base	66.5	110.0	40.2	47.6	86.3	23.7	88.3	65.1
+ QA-ViT		71.7	114.9	45.0	51.1	96.1	23.9	93.3	72.1
Δ		**+5.2**	**+4.9**	**+4.8**	**+3.5**	**+9.8**	**+0.2**	**+5.0**	**+7.0**
ViT+T5-large	Flan-T5-large	70.0	114.3	44.7	50.6	96.0	24.6	92.2	71.8
+ QA-ViT		72.0	118.7	48.7	54.4	106.2	26.0	95.4	78.9
Δ		**+2.0**	**+4.4**	**+4.0**	**+3.8**	**+10.2**	**+1.4**	**+3.2**	**+7.1**
ViT+T5-xl	Flan-T5-xl	72.7	115.5	48.0	52.7	103.5	27.0	94.1	77.0
+ QA-ViT		73.5	116.5	50.3	54.9	108.2	28.3	95.0	80.4
Δ		**+0.8**	**+1.0**	**+2.3**	**+2.2**	**+4.7**	**+1.3**	**+0.9**	**+3.4**
BLIP2 [31]	Flan-T5-xl	72.5	134.8	34.5	36.4	93.6	28.2	103.7	64.5
+ QA-ViT		74.6	136.6	36.6	38.1	97.4	28.4	105.6	67.4
Δ		**+2.1**	**+1.8**	**+2.1**	**+1.7**	**+3.8**	**+0.2**	**+1.9**	**+2.9**
BLIP2 [31]	Flan-T5-xxl	74.8	134.8	36.5	37.9	97.4	29.8	104.8	67.3
+ QA-ViT		75.6	135.9	37.5	39.9	98.7	30.4	105.8	68.7
Δ		**+0.8**	**+1.1**	**+1.0**	**+2.0**	**+1.3**	**+0.6**	**+1.0**	**+1.4**
InstructBLIP [15]	Flan-T5-xl	75.7	135.9	36.2	38.1	98.2	28.9	105.8	67.7
+ QA-ViT		76.0	136.9	37.4	39.4	99.9	28.8	106.5	69.2
Δ		**+0.3**	**+1.0**	**+1.2**	**+1.3**	**+1.7**	**-0.1**	**+0.7**	**+1.5**
InstructBLIP [15]	Flan-T5-xxl	76.1	136.1	37.4	38.7	99.0	31.1	106.1	68.5
+ QA-ViT		76.5	**138.2**	38.4	40.0	101.7	30.7	107.4	70.5
Δ		**+0.4**	**+2.1**	**+1.0**	**+1.3**	**+2.7**	**-0.4**	**+1.3**	**+2.0**
LLaVA-1.5 [33]	Vicuna-7B	79.7	133.5	57.4	61.6	126.4	33.9	106.6	93.0
+ QA-ViT		**80.5**	134.7	**59.1**	**62.4**	**128.7**	**36.5**	**107.6**	**94.7**
Δ		**+0.8**	**+1.2**	**+1.7**	**+0.8**	**+2.3**	**+2.6**	**+1.0**	**+1.7**

Table 1. **QA-ViT results**. Quantitative comparison of QA-ViT integrated into ViT+T5, BLIP2, InstructBLIP, and LLaVA-1.5, using different model sizes, with these baselines trained on the data described in Sec. 4.1. The evaluation covers general and scene-text VL benchmarks and 0-shot capabilities. QA-ViT consistently outperforms the different baselines, demonstrating its effectiveness and versatility.

length of the visual features by processing the different visual features. On the other hand, LLaVA-1.5 and ViT+T5 utilize a simple MLP that operates separately on the visual features. Despite this crucial difference, our method is compatible with both, leading to consistent gains.

- **LLM Architecture** – We experiment with both encoder-decoder (FLAN-T5 [14]) and decoder-only (Vicuna [13]). In the encoder-decoder case, we encode the textual guidance using the preexisting encoder, and in the decoder-only, we utilize the model's embedding module. We provide a comparison between these two alternatives in Sec. 5.1. Our experiments show that despite the significant LLM architecture differences, QA-ViT is compatible with both, showcasing its versatility.

Next, we examine the effects of scale-up on our approach by comparing the results of different model sizes. In particular, we consider `base`, `large`, and `xl` and `xl` and `xxl` for ViT+T5 and BLIP2 and InstrucrtBLIP, respectively. Our quantitative analysis demonstrates that our approach leads to consistent improvement across all model scales, making it compatible with different LLM sizes. Remarkably, for a given LLM size, applying QA-ViT is more beneficial than scale-up in terms of average general and scene-text performance. For example, InstructBLIP-xl + QA-ViT leads to 106.5 and 69.2 (general and scene-text averages), compared to InstructBLIP-xxl with 106.1 and 68.5 – an improvement of $+0.4$ and $+0.7$, compared to the scale-up. Based on these results, we conduct a more thorough analysis of our method's contribution in Sec. 4.5.

Lastly, we focus on InstructBLIP, as it utilizes an instruction-aware QFormer. In particular, this component processes the visual features with respect to the provided text, which conceptually resembles QA-ViT. Thus, one might presume that utilizing such a model might make QA-ViT contribution redundant. However, it is fundamentally different as our method is integrated inside the ViT and not on top of it. Hence, the QFormer cannot compensate for information disregarded in the output features of the ViT. On the contrary, QA-ViT, by being integrated into the ViT layers, can emphasize the relevant features and prevent their

Method	VQA^{v2}	VQA^{T}	TextCaps	VizWiz
mPLUG-DocOwl [53]	-	52.6*	111.9*	-
BLIP2 [31]	65.0	23.4	70.4	29.4
InstructBLIP [15]	-	30.9	75.6*	30.9
InstructBLIP^{+OCR} [15]	-	46.6	126.0*	30.9
OpenFlamingo-9B [5]	50.3	24.2	-	17.7
IDEFICS-9B [26]	50.9	25.9	25.4	35.5
IDEFICS-80B [26]	60.0	30.9	56.8	36.0
Shikra [9]	77.4*	-	-	-
Qwen-VL [6]	79.5*	**63.8***	-	35.2
LLaVA-1.5 [33]	79.7*	57.4*	126.4*	33.9
+ QA-ViT	**80.5***	59.1*	**128.7***	**36.5**
Δ	**+0.8**	**+1.7**	**+2.3**	**+2.6**

Table 2. **Comparison to generalist models**. Results comparison of QA-ViT integrated into LLaVA-1.5 with top-performing generalist models on VQA and captioning. QA-ViT outperforms existing methods in the VQA^{v2}, TextCaps and VizWiz. Models marked with $^{+OCR}$ receive a list of OCR tokens, and scores noted with * signify that the dataset's training images are observed in training.

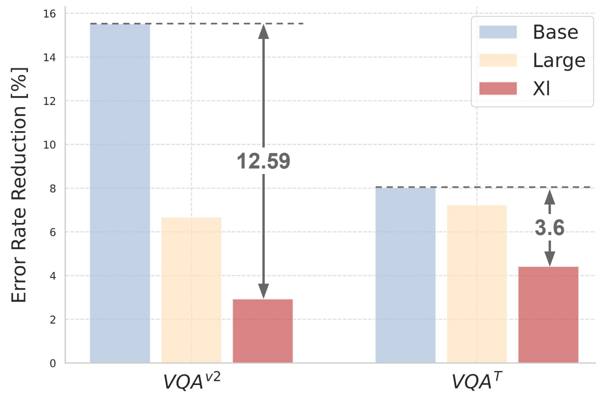

Figure 5. **QA-ViT effectiveness analysis.** Comparison of the trends in error rate reduction of QA-ViT in VQA^{T} and VQA^{v2} as the language model is scaled up. The relative performance improvements of our approach are more consistent across model scales in the former. These trends are attributed to each dataset's different question types' composition, where VQA^{T} exhibits more questions focusing on non-salient and overlooked elements.

potential disregardance, leading to performance gains.

4.4. Comparison to State-of-the-art

Despite QA-ViT being a model-agnostic approach that can be integrated into any VL model, we compare LLaVA-1.5 + QA-ViT to other state-of-the-art generalist methods. In particular, we consider mPLUG-DocOWL [53], OpenFlamingo-9B [5], IDEFICS-9B and 80B [26], Shikra [9] and Qwen-VL [6], and report the results in Tab. 2. As can be seen, QA-ViT pushes the performance of the LLaVA-1.5 model on the unseen VizWiZ beyond Qwen-VL and IDEFICS-80B, leading to the best performance across the considered models. In addition, QA-ViT leads to the top-performing generalist model in VQA^{v2}.

4.5. Why and When QA-ViT is Effective?

In this section, we better study the impact of QA-ViT. We argue that our method plays a crucial role in addressing two common image-question fail-cases within VL architectures: first, questions regarding image aspects disregarded by the vision model, and second, questions related to elements encoded by the vision model but misinterpreted by the LLM. While scaling up the LLM might mitigate some of the latter type of fail-case, the former remains challenging to address, hence, we consider the first as a more interesting setting for our method. To examine our claim, we propose to compare the gains of QA-ViT across different LLM scales in two datasets, VQA^{T} and VQA^{v2}, that differ in the composition of the fail-cases mentioned above. We categorize VQA^{T} as having more instances of the first fail-case and VQA^{v2} as having more of the second one since OCR information is more likely to be disregarded due to its relative scarcity in the ViT's pretraining captions compared to non-OCR visual data. Indeed, as anticipated, the trends in Fig. 5 align with our expectation that the gains of QA-ViT in VQA^{T} would be more significant when scaling up compared to VQA^{v2}. Although more substantial gains are generally observed in smaller models, our method leads to consistent improvements even on the largest models (*i.e.*, BLIP2-xxl InstructBLIP-xxl and LLaVA-1.5), as evidenced in Tab. 1.

5. Ablation Studies

In this section, we conduct extensive experiments to understand the performance improvements better and analyze the impact of our method. We first study the effect of different design choices (Sec. 5.1) and then analyze the contributions of different training data compositions (Sec. 5.2). Throughout this section, we focus on ViT-T5-large architecture.

5.1. Design Choices

We analyze different design choices and explore different settings for the textual guidance encoding and representations fusing while applying QA-ViT.

Finetuning Strategy Despite being parameter efficient, QA-ViT introduces more trainable parameters than the baseline. To validate that the improvements are credited to the method and not the additional capacity, we conduct experiments with two other finetuning techniques. First, analogous to deep prompt tuning, we train our model while inserting into QA-ViT a fixed textual prompt instead of the relevant question. By employing the same blocks as our method, this interpretation of prompt tuning (denoted as P.T.) isolates the contribution of question-conditioned image encoding. In addition, we also experiment with finetuning the entire baseline's vision encoder, which introduces

Inst.	Fuse	Freeze	VQA^{v2}	VQA^{T}
✗	✗	✓	70.0	44.7
P.T.	`late`	✓	70.1 (+0.1%)	45.8 (+1.1%)
✗	✗	✗	69.5 (-0.5%)	44.9 (+0.2%)
Enc.	`early`	✓	67.9 (-2.1%)	41.7 (-3.0%)
Enc.	`sparse`	✓	70.7 (+0.7%)	46.6 (+1.9%)
Enc.	`all`	✓	69.5 (-0.5%)	45.9 (+1.2%)
Emb.	`late`	✓	71.0 (+1.0%)	47.5 (+2.8%)
BERT	`late`	✓	71.8 (+1.8%)	48.3 (+3.6%)
CLIP	`late`	✓	71.8 (+1.8%)	48.0 (+3.3%)
Enc.	`late`	✓	**72.0** (+2.0%)	**48.7** (+4.0%)

Table 3. **Design choices ablation**. We mark the baseline and our top-performing configuration of QA-ViT in grey and yellow, respectively. Top: Results of different finetuning strategies. Middle: The effect of different integration points of QA-ViT. Bottom: Comparison of different instruction (Inst.) encodings.

a significant amount of trainable parameters. The results in the top part of Tab. 3 show that while QA-ViT leads to $+2.0\%$ and $+4.0\%$ on VQA^{v2} and VQA^{T}, P.T improves solely in $+0.1\%$ and $+1.1\%$, respectively. Comparing QA-ViT results with P.T. enables decomposing our method's improvement into gains attributed to additional capacity and to question-aware visual features, implying that the latter is the most significant. In addition, full finetuning CLIP, which introduces training instability, improves the baseline in VQA^{T} but reduces it on VQA^{v2}. This supports the choice of current VL works to freeze the ViT during pretraining.

Integration Point We explore different fusing locations – `early` (bottom layers), `late` (top layers), `sparse` (every 2 layers), and `all` (every layer). While `early`, `sparse`, and `late` add the same amount of trainable parameters, `all` doubles it. The results presented in the middle part of Tab. 3 demonstrate the significant advantage of `late` fusion. We attribute this to the hierarchical structure of the ViT's layers, in which early layers specialize in capturing low-level and localized visual details, while higher ones focus on extracting more abstract and high-level visual features. Thus, disregarding question-related image aspects is more likely to occur on the higher layers, QA-ViT is most effective in `late` fusion. Moreover, as the early layers extract low-level details, they should not be modified, and applying QA-ViT to them impairs the results.

Question Representation As specified in Sec. 3, we use the preexisting LLM's encoder (Enc.) to obtain the question representation. Here, we study the effect of different such choices and present their results at the bottom of Tab. 3. First, utilizing solely the embeddings (Emb.) is less effective than the encoder. We attribute this to the improved contextual understanding of the latter, enabling better guidance to the visual features in QA-ViT . Next, we experiment with using a designated language model, considering both a BERT [16] and the corresponding CLIP text encoder. While utilizing the system's language model is more parameter efficient and can lead to more seamless integration, a dedicated language model can better align with the vision model and offer a more modular and generic design. As can be seen, while both perform satisfactorily, the designated LLM is superior, while BERT outperforms CLIP.

Datasets	Size	VQA^{v2}	VQA^{T}	COCO	TextCaps
VQA	2.3M	71.2	45.8	29.9	34.3
+ CAP	3.0M	71.5	47.4	117.5	106.1
+ DOC	3.1M	**72.0**	**48.7**	**118.7**	**106.2**

Table 4. **Training data ablation**. Contribution analysis of different training dataset compositions on visual question answering and captioning, demonstrating the importance of multi-task data.

5.2. The Impact of Training Data

Our training data, described in Sec. 4.1, consists of three main data types: i) natural images visual question answering (VQA); ii) natural image captioning (CAP); and iii) documents understanding (DOC). We turn to evaluate the contribution of each of them and report the results in Tab. 4. As can be seen, adding CAP datasets into the VQA ones (second row) not only improves the captioning performance but also boosts the performance on the VQA ones. We attribute this to the enlargement and diversification of the training data. Moreover, incorporating DOC data, despite the significant change of domain (natural images vs. documents), increases the performance. We hypothesize that this is because QA-ViT maintains the original visual capabilities; it prevents the performance drop due to multi-domain data while leading to better OCR understanding. This, in return, improves the overall results, as observed in [20].

6. Discussion and Conclusions

In this work, we introduced an approach to condition the vision encoder in any multimodal vision-language architecture, named QA-ViT. Our method leads to question-aware visual features, improving their alignment with the provided query. Through extensive experimentation across a diverse set of vision-language models, we have demonstrated the effectiveness and versatility of our method. It consistently enhances the performance of these models across a range of benchmark tasks, encompassing both general and scene-text domains, as well as the challenging zero-shot setting. The introduction of QA-ViT represents a notable advancement in the pursuit of question-aware vision within VL modeling, making models more context-aware and enabling them to excel in various tasks. We hope our method will inspire further research striving towards improved text-aware mechanisms and designated pretraining techniques.

References

[1] Aviad Aberdam, Roy Ganz, Shai Mazor, and Ron Litman. Multimodal semi-supervised learning for text recognition. *arXiv preprint arXiv:2205.03873*, 2022. 4

[2] 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. *arXiv preprint arXiv:2301.07464*, 2023. 4

[3] 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, 3, 4

[4] 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

[5] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 7

[6] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 2, 3, 7

[7] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342, 2010. 5

[8] 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. 3, 4

[9] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 7

[10] 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. 1

[11] 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] 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. 3

[13] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 6

[14] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, 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. 3, 5, 6, 1

[15] 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, 4, 5, 6, 7

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 8

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3, 4

[18] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023. 5

[19] Roy Ganz and Michael Elad. Clipag: Towards generator-free text-to-image generation. *arXiv preprint arXiv:2306.16805*, 2023. 1

[20] Roy Ganz, Oren Nuriel, Aviad Aberdam, Yair Kittenplon, Shai Mazor, and Ron Litman. Towards models that can see and read. *arXiv preprint arXiv:2301.07389*, 2023. 1, 3, 4, 8

[21] 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

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[23] 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. *arXiv preprint arXiv:2106.09685*, 2021. 5, 1

[24] Wenbo Hu, Yifan Xu, Y Li, W Li, Z Chen, and Z Tu. Bliva: A simple multimodal llm for better handling of text-rich visual questions. *arXiv preprint arXiv:2308.09936*, 2023. 1

[25] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 4

[26] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. Obelics: An open web-scale filtered dataset of interleaved image-text documents, 2023. 7

[27] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR, 2023. 3

[28] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, et al. mplug: Effective and efficient vision-language learning by cross-modal skip-connections. *arXiv preprint arXiv:2205.12005*, 2022. 3

[29] 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.

[30] 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, 3

[31] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1, 2, 3, 4, 5, 6, 7

[32] 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. 4

[33] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 2, 3, 4, 5, 6, 7

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1, 2, 3

[35] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1

[36] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 1

[37] 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. 4

[38] 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.

[39] 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. 4

[40] 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

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 5, 1

[42] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021. 3, 4

[43] Noam Rotstein, David Bensaid, Shaked Brody, Roy Ganz, and Ron Kimmel. Fusecap: Leveraging large language models to fuse visual data into enriched image captions. *arXiv preprint arXiv:2305.17718*, 2023. 1

[44] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2

[45] 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. 1, 4

[46] 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. 1, 3, 4

[47] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023. 3

[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[49] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al.

Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 3

[50] 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. *arXiv preprint arXiv:2205.14100*, 2022. 3

[51] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022.

[52] 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

[53] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, et al. mplug-docowl: Modularized multimodal large language model for document understanding. *arXiv preprint arXiv:2307.02499*, 2023. 7

[54] 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. *arXiv preprint arXiv:2306.17107*, 2023. 1, 2, 3