

---

# PaLI-X: On Scaling up a Multilingual Vision and Language Model

---

**Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu,  
 Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, Siamak Shakeri,  
 Mostafa Dehghani, Daniel Salz, Mario Lucic, Michael Tschannen, Arsha Nagrani,  
 Hexiang Hu, Mandar Joshi, Bo Pang, Ceslee Montgomery, Paulina Pietrzky,  
 Marvin Ritter, AJ Piergiovanni, Matthias Minderer, Filip Pavetic, Austin Waters,  
 Gang Li, Ibrahim Alabdulmohsin, Lucas Beyer, Julien Amelot, Kenton Lee,  
 Andreas Peter Steiner, Yang Li, Daniel Keysers, Anurag Arnab, Yuanzhong Xu,  
 Keran Rong, Alexander Kolesnikov, Mojtaba Seyedhosseini,  
 Anelia Angelova, Xiaohua Zhai, Neil Houlsby, Radu Soricut**

Google Research  
 pali-communications@google.com

## Abstract

We present the training recipe and results of scaling up PaLI-X, a multilingual vision and language model, both in terms of size of the components and the breadth of its training task mixture. Our model achieves new levels of performance on a wide-range of varied and complex tasks, including multiple image-based captioning and question-answering tasks, image-based document understanding and few-shot (in-context) learning, as well as object detection, video question answering, and video captioning. PaLI-X advances the state-of-the-art on most vision-and-language benchmarks considered (25+ of them). Finally, we observe emerging capabilities, such as complex counting and multilingual object detection, tasks that are not explicitly in the training mix.

## 1 Introduction

The success of scaling language models [1, 2, 3, 4] makes it appealing to similarly scale Vision-Language (V&L) models, and investigate the improvements, capabilities, and emergent properties of such models. Inspired by the work in [5], we present PaLI-X, a multilingual vision and language model with reusable scaled-up components, consisting of a pretrained large-capacity visual encoder (using [6] as the starting point) and a pretrained language-only encoder-decoder (using [7] as the starting point), further trained at-scale on a vision-and-language data mixture using a combination of self-supervision and full-supervision signals.

One clear pattern that emerges from the combination of results from PaLI [5] and the work we present in this paper is that scaling *both* V&L components together brings increases in performance across a wide range of tasks. We show this by comparing against the same benchmarks used for PaLI (Fig. 1, Left), and also against new benchmarks for which the new capabilities of PaLI-X are evaluated (e.g., ChartQA, AI2D, DocVQA, InfographicVQA, as well as video understanding tasks). We observe that scaling leads to large improvements over the results of the PaLI model, and also over specialized large-scale models that are trained specifically to solve certain tasks, often with the help of (often much larger) text-only LLMs [8]. In particular, we find that increasing both the effective capacity of the vision component (which [9] does more unilaterally) and of the language component

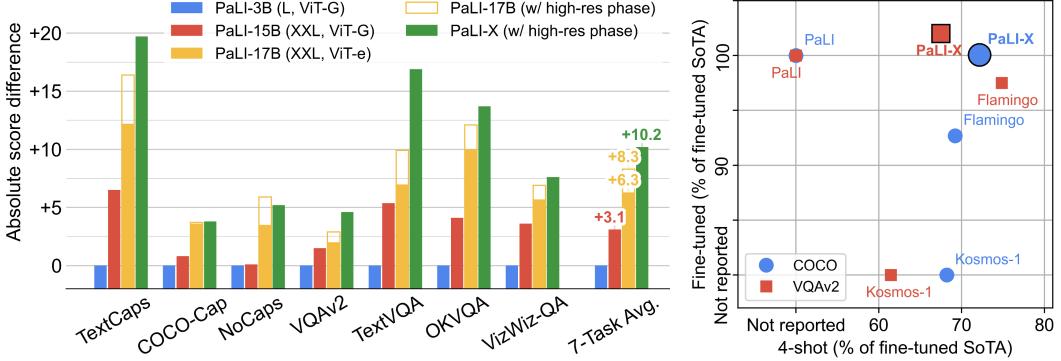


Figure 1: [Left] Comparing PaLI-X against PaLI on image-captioning and VQA benchmarks. [Right] The Pareto frontier between few-shot and fine-tuned performance, comparing PaLI-X with PaLI [5], Flamingo [10], and Kosmos-1 [11].

(which [10] also does unilaterally) is beneficial; the new PaLI-X model provides more balanced parameter allocation than any other prior work (roughly 40%-60% split of the total capacity).

Aside from confirming the impact of scale, the original contribution of PaLI-X consists in leveraging the mixture-of-objectives proposed in [7] for vision-and-language modeling, and showing that it results in a model that improves both state-of-the-art results and the Pareto frontier for fine-tuning and few-shot configurations (Fig. 1, Right).

We also observe emergent properties based on PaLI-X’s results compared to previous models with similar architecture but smaller sizes. For instance, we report drastically improved performance on the counting ability (See Table 1 and Appendix B), both for the plain variety (count all instances of a class) and the complex variety (count instances based on a natural language description), that are not attributable to training design<sup>1</sup>. Additionally, we present qualitative insights into the model’s performance (Appendix A), with an emphasis on multilingual transfer learning such as the ability to detect objects using non-English labels (Fig. 2), and the ability to switch between the language of text present in the image (e.g., English) and the language of the generated image caption (e.g., Romanian).

Our technical contributions include the following:

1. We scale a Vision-Language model to achieve outstanding performance on a wide variety of benchmarks. We observe that scaling *both* the Vision & Language components is advantageous and report that performance remains unsaturated at this scale.
2. We show that training such a model with a mixture of objectives that combines prefix-completion and masked-token completion improves the Pareto frontier for fine-tuning vs few-shot performance at this scale.
3. We show that a high-capacity vision encoder (ViT-22B) can be effectively co-trained for image classification and OCR label classification<sup>2</sup> to achieve significant improvements on V&L tasks for which the understanding of text-within-image is crucial.
4. Overall, PaLI-X improves SoTA results via fine-tuning on 15+ benchmarks, and we show that it is the first of its kind to simultaneously adapt via multitask fine-tuning to a diverse set of benchmarks without significant performance degradation.

## 2 Related Work

Similar to large language models such as GPT4 [12] and PaLM [1], the benefit of scale has also been observed in recent vision and vision-language models. Flamingo [10] used a frozen language

<sup>1</sup>Plain counting is usually achievable via good object detection, while complex counting requires a fine-grained understanding of the alignment between language-based specifications and visually-based occurrences.

<sup>2</sup>We use OCR tokens produced by the GCP Vision API over the training images as targets.

component and demonstrated the benefit of scaling up this part up to 70B parameters on the few-shot multimodal capabilities, while the vision encoder is fixed with 435M parameters. GIT [9], on the other hand, explored scaling of the vision component up to 4.8B parameter, with a 300M parameter language decoder. PaLI [5] explored jointly scaling the vision and language component, to 4B and 17B, respectively, and showed that scaling both components benefits a wide range of vision-language tasks. All these models took advantage of vision and language unimodal pretrained models as backbones to start multimodal training. Recently, on the vision model side, a vision transformer with 22B parameter has been introduced [6]. In this work, we make use of a ViT-22B model specifically tuned for OCR capability to explore scaling Vision-Language models to even larger parameter regime.

As first shown in [13], *large* language models are sometimes able to solve new unseen tasks at inference as long as a few examples –or *shots*– are provided as inputs. This is usually referred to as in-context learning [14]. Follow-up work proposed improved ways to split and prompt the shots, such as Chain of Thought [15] or Least-to-Most prompting [16]. So far, the vast majority of this work has been done in the context of language inputs [17]. In this work, we explore multimodal in-context learning with pairs of images and captions. Our work is aligned in spirit to Flamingo [10] that uses interleaved image text pairs in the same web page and in-context tuning [18] during pre-training. We first group the image-text pairs by url and split each group to a “shots” set and a “target” set. Then we use the few examples in the “shots” set as input features to predict the examples in the target set.

Besides solving vision-language tasks in multiple domains, recent VLMs also attempted solving these tasks at once instead of fine-tuning on each individual benchmark. Unified-IO [19] performed multitask fine-tuning and reported solid results across 16 benchmarks. Spotlight [20] reported that inside the UI domain, multitask fine-tuning can achieve a performance close to task-specific fine-tuning. In this work, we show that PaLI-X can be simultaneously fine-tuned with a diverse set of benchmarks in multiple domains without performance degradation.

### 3 Model

#### 3.1 Architecture

The PaLI-X model architecture follows the encoder-decoder architecture: image(s) are processed by a ViT encoder, with the resulting visual embeddings fed to an encoder-decoder backbone, along with embeddings from additional text input (e.g., question / prefix / prompt). More details are provided in Appendix A.

**Visual component** Our visual backbone is scaled to 22B parameters, as introduced by [6], the largest dense ViT model to date. To equip the model with a variety of complex vision-language tasks, we specifically focus on its OCR capabilities. To that end, we incorporate an OCR-based pretraining as follows: images from the WebLI dataset [5] are annotated with OCR-text detected by GCP Vision API; the encoder is then further pre-trained with a mixture of the original JFT-based classification task and a new OCR-based classification task (whether or not a given token occurred in the image according to OCR results). See Appendix A for additional details on the visual component. PaLI-X is designed to take  $n \geq 1$  images as inputs (for few-shot and video understanding), with tasks involving a single image as the  $n = 1$  case. For  $n > 1$ , each image is independently processed by the ViT module, and the patch-level embeddings coming out of ViT are flattened and concatenated to form the visual input (See Appendix A). Note that similar to the single-image case, there is no pooling over the spatial dimension before visual embeddings are aggregated over the temporal dimension. That is, for an  $n$ -frame input with  $k$ -patches per frame, the resulting visual input has  $n * k$  tokens.

**Overall model** The encoder-decoder backbone is initialized from a variant of the UL2 [7] encoder-decoder model that uses 32B parameters. The architecture of this variant has 50 layers in both encoder and decoder (up from 32 layers in [7]), and is pretrained on a mixture of text data similar to [7]. The visual embeddings, after going through a projection layer, are concatenated with the token embeddings of the text input, and fed to the encoder-decoder backbone. Most of the pretraining tasks (with the exception of the masked image token task) predict text-only output from this multimodal input. The text input to the model typically consists of a prompt that marks what type of task it is (e.g., "Generate caption in  $\langle \text{lang} \rangle$ " for captioning tasks) and encode necessary textual input for the task (e.g., "Answer in  $\langle \text{lang} \rangle$ : {question}" for VQA tasks). For tasks that need OCR capabilities, we experiment with either relying solely on the text-encoding capabilities of the vision encoder, or optionally including tokens extracted by an upstream OCR system fed as additional text inputs.

**Few-shot formulation** In the few-shot setting, for a given *target example* the model receives a number of “labeled” examples (in the form of additional  $\langle$ image, text $\rangle$  pairs) that we refer to as *shots/exemplars*. The hypothesis is that information contained in these exemplars provides the model with useful context to generate predictions for the target example. Formally, the input with  $N$  shots is a sequence  $(t_1, \dots, t_N, t_T, i_1, \dots, i_N, i_T)$ , where  $t_1 : t_N$  and  $i_1 : i_N$  are texts and images for the  $N$  shots, and  $t_T$  and  $i_T$  are the text (prompt) and image for the target example. PaLI-X processes this input as follows: all images, including the target one, are first independently processed by the visual encoder, and the resulting patch-level embeddings are flattened and concatenated to form the visual input sequence. After going through a projection layer, they are concatenated with the text embeddings to form the multimodal input sequence used by the encoder. We implement additional optimizations including distributing the exemplars between the encoder and the decoder, and an attention re-weighting mechanism (see Appendix B).

### 3.2 Pretraining Data and Mixture

The main pretraining data for our model is based on WebLI [5], consisting of roughly one billion images with alt-texts from the web and OCR annotations (using the GCP Vision API), covering over 100 languages. In addition to WebLI  $\langle$ image, text $\rangle$  pairs, we introduce here *Episodic WebLI* data, where each episode corresponds to a set of such pairs. We aim to have each episode contain loosely related images (i.e., they are clustered according to their URL field), so as to encourage attention among examples in an “episode”. We find this new dataset (with 75M episodes and around 400M images in total) important for developing the few-shot capabilities of the model.

The pretraining mixture consists of the following data and objectives: (i) span corruption on text-only data (15% of tokens); (ii) split-captioning on WebLI alt-text data [21, 5]; (iii) captioning on CC3M [22] on native and translated alt-text data (over the same 35 languages covered by XM3600 [23]); (iv) split-ocr [24] on WebLI OCR annotations; (v) visual-question-answering objective over  $\langle$ image, question, answer $\rangle$  pairs generated using the VQ<sup>2</sup>A method [25] over the CC3M training split, over native and translated text (same 35 language pairs); (vi) visual-question-generation objective, using the same pairs as above; (vii) visual-question-answering objective over  $\langle$ image, question, answer $\rangle$  pairs using the Object-Aware method [26] (English only); (viii) captioning on Episodic WebLI examples (target alt-text predicted from the remaining alt-text and images); (ix) visual-question-answering on 4-pair examples (resembling Episodic WebLI and using VQ<sup>2</sup>A-CC3M pairs), with the answer target conditioned on the other pairs of  $\langle$ image, question, answer $\rangle$  data. (x) pix2struct objective, introduced in [27], targeting page layout and structure using screenshot images paired with DOM-tree representations of html pages. (xi) Captioning on short video data, using the VTP data [10] (using four frames per video). (xii) object-detection objective on WebLI data, whereby an OWL-ViT model [28] (L/14) is used to annotate WebLI images, resulting in hundreds of pseudo object labels and bounding boxes per image. (xiii) image-token prediction objective, whereby we tokenize WebLI images (256×256 resolution) using a ViT-VQGAN [29] model with patch size 16×16 (256 tokens per image); this objective is framed as a 2D masked-token task (i.e., fill-in the missing grid pieces, with the corresponding image pixels also masked). Note that the image-token prediction objective is added mainly as a condition to check whether it adversarially impacts the performance on language-output tasks; our ablation experiments show that is does not.

### 3.3 Training Stages

Our model is trained in two stages. In stage 1, the visual encoder (after mixed-objective training) is kept frozen, while the rest of the parameters are trained on a total of 2.2B examples at the base resolution 224×224 (native to ViT-22B), using the entire mixture. In stage 2, it continues training using only the OCR-related objectives (pix2struct and split-ocr) plus the object detection objective; this is done in several substages, during which image resolution is gradually increased to 448×448, 672×672 and finally 756×756.

## 4 Experiments

### 4.1 Image Captioning and Visual Question Answering

Our results demonstrate that the larger capacity in PaLI-X scales well in both its vision and language components, and it is particularly beneficial for more challenging scene-text and document understanding tasks. Our model outperforms the SOTA on diverse vision-language tasks, with significant margins in some cases.

**Benchmark datasets** The Image Captioning and VQA benchmarks used for evaluation is summarized in Appendix B, including 6 Image Captioning benchmarks (COCO (Karpathy split [30]), NoCaps [31], TextCaps [32], VizWiz-Cap [33], Screen2Words [34], Widget-Cap [35]) and 13 VQA benchmarks (VQAv2 [36], OKVQA [37], TallyQA [38], TextVQA [39], VizWiz-VQA [40], STVQA [41], OCRVQA [42], InfographicVQA [43], DocVQA [44], AI2D [45] ChartQA [46], OVEN [47], InfoSeek [48]). These tasks span a wide range of visual domains, from natural images, illustrations to documents and user interfaces (UIs). We also include results of multilingual captioning on XM3600 in Appendix B.

#### 4.1.1 Per-task fine-tuning results

**Experimental setup** We fine-tune PaLI-X with frozen ViT-22B; the learning rate follows a linear decay from initial value 1e-4 for all fine-tuning experiments. See Appendix B for more details.

Model	COCO		NoCaps		VQAv2		OKVQA		TallyQA	
	Karp.-test		val	test	test-dev	test-std	val		simple	complex
GIT2 [9] (5.1B)	145.0		126.9	<b>124.8</b>	81.74	81.92	-	-	-	-
Flamingo [10] (80B)	138.1		-	-	82.0	82.1	57.8*	-	-	-
BEiT-3 [49] (1.9B)	147.6		-	-	84.2	84.0	-	-	-	-
PaLM-E [50] (562B)	138.7		-	-	80.0	-	<b>66.1</b>	-	-	-
MoVie [51]	-		-	-	69.26	-	-	-	74.9	56.8
PaLI [5](17B)	149.1		<b>127.0</b>	124.4	84.3	84.3	64.5	81.7	81.7	70.9
PaLI-X (55B)	<b>149.2</b>		126.3	124.3	<b>86.0</b>	<b>86.1</b>	<b>66.1</b>	<b>86.0</b>	<b>86.0</b>	<b>75.6</b>

Table 1: Results on COCO Captions (Karpathy split), NoCaps, VQAv2 [36], OKVQA [37], and TallyQA [38] with end-to-end modeling without OCR pipeline input (“simple” and “complex” are test subsplits).

Model	Text Caps	VizWiz Cap	Text VQA	VizWiz VQA	ST VQA	OCR VQA	Info VQA	Doc VQA	AI2D QA	Chart QA	Screen2 Words	Widget Cap	OVEN	Info Seek
<i>with OCR pipeline input</i>														
SoTA	160.4 [5]	124.7 [5]	73.67 [52]	73.3 [5]	79.9 [5]	67.5 [53]	47.4 [54]	84.7 [54]	38.5 [45]	45.5 [46]	-	-	-	-
PaLI-X	<b>163.7</b>	<b>125.7</b>	<b>80.78</b>	<b>74.6</b>	<b>84.5</b>	<b>77.3</b>	<b>54.8</b>	<b>86.8</b>	<b>81.4</b>	<b>72.3</b>	-	-	-	-
<i>without OCR pipeline input</i>														
SoTA	145.0 [9]	120.8 [9]	67.27 [9]	70.7 [5]	75.8 [9]	71.3 [27]	40.0 [27]	76.6 [27]	42.1 [27]	70.5 [8]	109.4 [27]	141.8 [20]	20.0 [47]	17.7 [48]
PaLI-X	<b>147.0</b>	<b>122.7</b>	<b>71.44</b>	<b>70.9</b>	<b>79.9</b>	<b>75.0</b>	<b>49.2</b>	<b>80.0</b>	<b>81.2</b>	<b>70.9</b>	<b>127.9</b>	<b>153.0</b>	<b>23.1</b>	<b>21.8</b>

Table 2: Results on benchmarks more focused on text understanding capabilities. For OVEN [47] & InfoSeek [48], we follow the proposed 224×224 resolution settings for fair comparison.

First, we present benchmarks results for the condition where external OCR systems are not used (Table 1, see Appendix B for an extended table.). The trend is that PaLI-X matches or improves SoTA results on these benchmarks, with a particularly significant improvement on the TallyQA benchmark over MoVie [51] (specialized counting model), at +11.1 for simple counting questions (e.g., “how many giraffes”) and +18.8 for complex counting questions (e.g., “how many giraffes are drinking water”); there are significant improvements over PaLI [5] as well, indicating that scale plays an important role in the ability of such models to perform counting tasks. We additionally note the state-of-the-art result on VQAv2 at 86.1 accuracy, achieved with an open-vocabulary generative

approach, and the performance on OKVQA at 66.1 accuracy, matching the much-larger PaLM-E [50] model performance.

Next, we examine text-heavy V&L benchmarks, for which upstream OCR systems can be used to improve performance. As shown in Table 2, PaLI-X improves SoTA for all Captioning and VQA benchmarks across the board, either without or with additional OCR input (using GCP Vision API). For instance, a significant jump of +42.9 points is observed on AI2D<sup>3</sup>, a multiple-choice benchmark where choices are provided along with each question. Being able to have the text choices as input benefits PaLI-X compared with the previous SoTA Pix2Struct [27] which has to render the text on the image, but this does not explain all the improvements. In a question-only configuration (no answer choice present), PaLI-X achieves 46.3 on AI2D, more than 4 points higher than Pix2Struct’s result.

In general, having access to OCR texts extracted by an external OCR pipeline boosts performance. Still, for several benchmarks (e.g., AI2D, ChartQA, OCRVQA and Widget-Cap), PaLI-X’s end-to-end performance when using its intrinsic OCR capability is close to that leveraging additional OCR input. A common feature for these benchmarks is that they have well-oriented text – diagrams, charts, book covers or user interfaces, with reasonably large font size at  $756 \times 756$  resolution. For tasks involving scene text in natural images (TextCaps, TextVQA, STVQA) or very high density of small texts (DocVQA, InfoVQA), results still highlight clear benefits when utilizing an external OCR model.

#### 4.1.2 Multitask Fine-tuning

We simultaneously fine-tune and evaluate the pretrained checkpoints on multiple benchmarks belonging to the same category. We deduplicated every training set over the test sets of every task in the mixture to prevent the leakage of any test-set examples into the mixed training set. This is useful as it leads to a single fine-tuned model that performs all the tasks, rather than having to fine-tune each task separately. We performed such multitask fine-tuning on all Image Captioning benchmarks and most VQA benchmarks, respectively.

Table 3 shows the multitask fine-tuning result for captioning tasks. The performance on COCO is slightly decreased in the multitask setting, which is likely a result of this task needing longer training to converge. For Screen2Words, having the smallest train and dev/test sets could be responsible for the performance fluctuation. Notably, VizWiz-Cap and Widget-Cap shows improved performance from multitask fine-tuning. Overall, the average performance decreases by 1.4 points (0.2 excluding Screen2Words) with multitask fine-tuning, while offering the clear advantage of having a single checkpoint to perform all these tasks. Appendix B shows similar results for VQA tasks. We consider this outcome a positive result that establishes the on-par performance between multitask fine-tuning and single-task fine-tuning for diverse benchmarks, in contrast with previous work which argued a gap between single-task and multitask fine-tuning [19], or demonstrated little gap over benchmarks from the same domain [20].

Method	COCO	NoCaps	Text Caps	VizWiz Cap	Screen2 Words	Widget Cap	Avg.
Split	Karp.-test	val	val	test-dev	test	test	-
SOTA (Single-task FT)	149.1	<b>127.0</b>	148.6	119.4	109.4	136.7	
PaLI-X Single-task FT	<b>149.2</b>	126.3	150.8	123.1	<b>127.9</b>	153.2	-
PaLI-X Multitask FT	147.3	125.6	<b>154.6</b>	<b>124.2</b>	120.6	<b>153.7</b>	-
Multitask (+/-)	<b>-1.9</b>	<b>-0.7</b>	<b>+3.8</b>	<b>+1.1</b>	<b>-7.3*</b>	<b>+0.5</b>	<b>-1.4 (-0.2 w/o “*”)</b>

Table 3: Scores from multitask fine-tuning compared with those from single-task fine-tuning for Image Captioning. Validation or test-dev set numbers are reported for some tasks.

#### 4.1.3 Few-shot Evaluation

We fine-tuned the PaLI-X model on a mixture of few-shot tasks. The few-shot mixture contains Episodic mixtures, (Non-Episodic) Webli and (Non-Episodic) CC3M data. Note that all of these datasets were already used in previous stages of training, but with lower mixture proportions. During

<sup>3</sup>As with all the other benchmarks, our training examples are carefully deduped to exclude images occurring in these benchmarks, including AI2D. Such results, therefore, are *not* attributable to train-test data leakage.

pre-training, we only use up to 4 shots, with both encoder and decoder shots (see Appendix B). For fine-tuning, we use up to 8 encoder shots and do not use decoder shots.

We evaluate the few-shot performance on COCO caption (Karpathy test split [30]), and XM3600 [23] datasets. For each task, we first create a “shots pool” with 256 examples that are randomly selected from the task’s training set. As the XM3600 benchmark does not come with a training set, we use Google Translate API to enhance the COCO Karpathy training set with captions in the 35 languages represented in XM3600. Then, for each test data point, we randomly pick  $N$  shots from the pool as the actual few-shot examples. Following [10], we also evaluate on 2 text-only shots settings where only the textual part of 2 randomly sampled few-shot examples are used.

Table 4 reports the few-shot captioning performance on English and multilingual captioning, as well as few-shot VQA performance on VQAv2. PaLI-X achieves SOTA few-shot results on COCO with both 4 shots and 32 shots; it outperforms previous SOTA by +4.4 CIDEr points for 4-shot, suggesting a strong ability to efficiently gather hints from few examples. We also report few-shot CIDEr scores averaged over 35 languages using XM3600, demonstrating PaLI-X’s multilingual capabilities. Meanwhile, although PaLI-X also performs decently on VQAv2, the gap behind the SoTA Flamingo model [10] (which freezes the language backbone) may be the result of losing some of the few-shot text-only QA capability by fine-tuning the language backbone, which supports the hypothesis regarding the tension between few-shot and fine-tuning abilities.

Method	COCO Captions		XM3600 Cap. (35-lang avg.)		VQAv2	
	4 shots	32 shots	4 shots	32 shots	4 shots	32 shots
Prev. SoTA [10]	103.2	113.8	N/A (53.6 w/ fine-tune [5])		<b>63.1</b>	<b>67.6</b>
PaLI-X	<b>107.6</b>	<b>114.5</b>	45.1	47.1	56.9	57.1

Table 4: Few-shot performance of the PaLI-X model (multilingual captioning for XM3600).

## 4.2 Video Captioning and Question Answering

We fine-tune and evaluate the PaLI-X model on 4 video captioning (MSR-VTT [55], VATEX [56], ActivityNet Captions [57], Spoken Moments in Time [58]) and 3 video question answering benchmarks (NExT-QA [59], MSR-VTT-QA [60], ActivityNet-QA [61]). A brief description of each benchmark and clarifications on their usage are provided in Appendix C.

**Experimental setup** We fine-tune our model (with base resolution  $224 \times 224$ ) for each task separately, use the validation split for early stopping, and report performance on the test split. We use a learning rate of  $10^{-4}$  for all tasks, and do not adapt any hyperparameters for specific tasks. Frames are sampled using a fixed temporal stride for each dataset (determined based on the video length distribution in that dataset such that the product of the number of frames and stride is larger than the total number of frames for half of the videos), and we experimented with including up to 8 or 16 frames per video. We did not include pooling over the spatial dimension; embeddings for  $16 \times 16$  patches per frame are provided as visual input to the multimodal encoder.

**Results** We report CIDEr score for the video captioning tasks. Video QA tasks are treated as open-ended generation tasks; we report full-string accuracy (for MSR-VTT-QA and ActivityNet-QA) and WUPS metrics (NExT-QA) in [65, 59]. As shown in Table 5, the 16-frames version has an edge over the 8-frame version, sometimes with a significant margin (e.g., close to a 6 point increase in CIDEr score for ActivityNet-Captions). More importantly, while PaLI-X pretraining was dominated by image-text tasks, we were able to achieve new SOTA performance for 5 out of 7 tasks<sup>4</sup>, and performed very close to prior SOTA on MSR-VTT-QA (47.1 vs 47.4).

## 4.3 Image classification

To test image classification capabilities we fine-tuned PaLI-X and models from [5] on ImageNet [66] and evaluated the resulting model on ImageNet-REAL [67] and out-of-distribution

<sup>4</sup>As noted in Table 5, current SOTA on NExT-QA for the open-ended QA task was achieved by Flamingo 32-shot, which had outperformed prior fine-tuning SOTA. To the best of our knowledge, PaLI-X performance on this task does outperform existing published fine-tuning performances, with the caveat that we do not have information on what Flamingo fine-tuning would have achieved on this task.

Method	MSR-VTT		Activity-Net		VATEX	SMIT	NExT-QA
	Cap. [55]	QA [60]	Cap. [57]	QA [61]	Cap. [56]	Cap. [58]	QA [59]
Prior SOTA	75.9	<b>47.4</b>	52.5	44.7	94.0 <sup>†</sup>	28.1 <sup>‡</sup>	33.5 <sup>§</sup>
	GIT2 [9]	Flamingo [10]	PDVC [62]	VINLU [63]	GIT2 [9]	MV-GPT [64]	Flamingo 32shot [10]
PaLI-X (8fr)	74.6	46.9	49.0	48.4	66.0	42.5	37.0
PaLI-X (16fr)	<b>76.8</b>	<b>47.1</b>	<b>54.9</b>	<b>49.4</b>	69.3	<b>43.5</b>	<b>38.3</b>

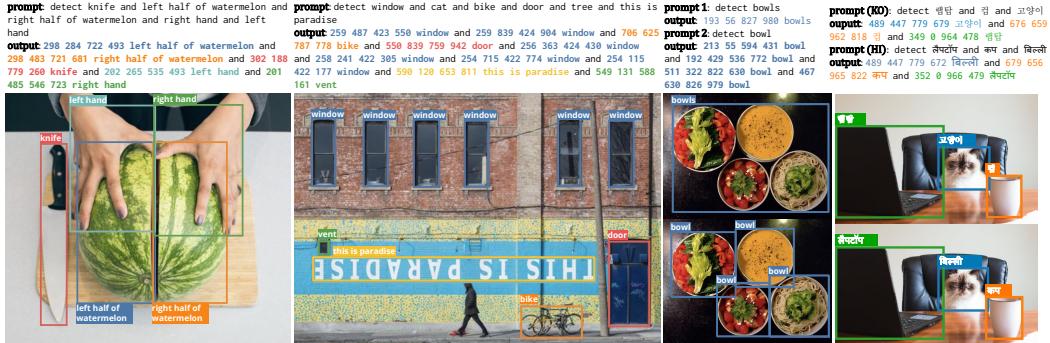
Table 5: Results for Video Captioning and Video-QA using 8 frames (8fr) or 16 frames (16fr). <sup>†</sup>GIT2 uses Self-Critical Sequence Training to directly optimize the CIDEr metric for VATEX. <sup>‡</sup>SMIT has not been used for video captioning before, we apply MV-GPT [64] and report results on the test set. <sup>§</sup>Numbers were obtained using 32-shot; since Flamingo 32-shot outperforms fine-tuning SOTA on this open-ended QA task, they did not conduct further fine-tuning experiments for this task.

datasets: ImageNet-R [68], ImageNet-A [69], ImageNet-Sketch [70], ImageNet-v2 [71]. We used the model from the first training stage (at resolution 224) and the one from the last training stage (at resolution 756). We used the same training hyperparameters for all of runs (selected without any hyperparameter tuning; mode details in Appendix D).

The results can be seen in Table 6. We compare the results to generative model with open vocab – GIT2 [9] (using 384 image resolution), which is the current SOTA for full fine-tuning on ImageNet. PaLI-X achieves SOTA results for generative models on Imagenet, and other datasets. We also performed zero-shot evaluation for PaLI-X and the results can be found in Appendix D.

Model (resolution)	INet [66]	REAL [67]	INet-R [68]	INet-A [69]	INet-Sketch [70]	INet-v2 [71]
GIT2 [9] (384)	<b>89.22</b>	-	-	-	-	-
PaLI-17B [5] (224)	86.13	88.84	78.21	50.00	71.21	78.91
PaLI-X (224)	88.22	90.36	77.66	55.97	72.56	81.42
PaLI-X (756)	<b>89.19</b>	<b>90.98</b>	<b>80.06</b>	<b>72.57</b>	<b>73.37</b>	<b>83.66</b>

Table 6: Classification accuracy (top-1) fine-tuned on Imagenet [66].



Credits: Watermelon/Cat; Sarah Pflug (burst), Bowls; ariesandrea (flickr), Wall; Matthew Henry (burst)

Figure 2: Examples demonstrating multilingual, OCR and other capabilities transferred to detection.

#### 4.4 Object Detection

Object detection can be easily formulated in our model as shown in pix2seq [72], The dataset mix used for pre-training is presented in Sec. 3; detection data was included up to and including the stage using resolution 672, after which a separate detection-specific model was fine-tuned on detection data. Before detection-specific tuning, LVIS [73] & COCO labels were removed from all detection training datasets, allowing zero-shot evaluation on LVIS.

Bounding box mean AP on LVIS is shown in Table 7, including zero-shot performance; the detection-tuned model reaches an AP of 31 in general, and 31.4 on rare classes, and about 12 for both in zero-shot. Performance on rare classes was on par with performance on common classes, a difficult

feat traditionally accomplished by complicated sampling schedules and augmentations. In our set up, it is directly enabled by PaLI-X’s diverse training mix. This could likely be further improved with investment in fine-tuning e.g. using noise-augmentation methods from pix2seq [72], or a further stage of high-resolution, LVIS only training. Qualitatively, we observe emergence of many interesting phenomena enabled by co-training with non-detection tasks; for example, multilingual detection, OCR bounding boxes and longer descriptions, none of which are included in detection training, are often handled well by PaLI-X. Additional results and information can be found in Appendix E.3.

	LVIS AP	LVIS AP <sub>Rare</sub>
ViLD [74] (tuned on non-rare LVIS)	29.3	26.3
Region-CLIP [75] (tuned on non-rare LVIS)	32.3	22.0
OwLViT-L/16 [28] (tuned on non-rare LVIS)	34.7	25.6
OwLViT-L/16 [28] (with Object365 and VG datasets)	34.6	31.2
PaLI-X (Zeroshot)	12.36	12.16
PaLI-X (Detection-tuned)	30.64	31.42

Table 7: PaLI-X object detection results on LVIS. The diverse pre-training mix enables parity performance between LVIS rare and common classes. Other related approaches are shown for context, but are not directly comparable.

## 5 Model Fairness, Biases, and Other Potential Issues

Large models, if left unchecked, have the potential to inflict harm on society – such as amplifying **biases** [76, 77, 78, 79], causing **disparities** [78, 80, 81], or **encoding narrow cultural perspectives** [82, 83]. Hence, evaluating PaLI-X for such potential issues is important. We focus our RAI evaluation on three parts: (1) **harmful associations**, such as toxicity and profanity, (2) **demographic parity in the model’s output**, such as **encoding societal stereotypes/biases**, and (3) **performance disparity across subgroups**. This breakdown follows earlier works in the literature, such as [84].

**Toxicity / profanity.** We estimate the level of toxicity and profanity in the generated captions, including when disaggregated across subgroups. We use the FairFace dataset [85] that comprises of images of people with ground-truth attributes: gender presentation, age and ethnicity. We generate captions and use the Perspective API [86] (threshold > 0.8) to measure toxicity and profanity. Table 8 summarizes the results; we observe a low level of toxicity/profanity across all slices. Tables 9 and 10 provide a detailed breakdown of toxicity/profanity results for all subgroups in FairFace dataset. In Tables 11 and 12, we report similar results in the MIAP [87] dataset, disaggregated by perceived gender and age.

	Gender		Ethnicity			Age			<b>Overall</b>
	Lowest	Highest	Lowest	Median	Highest	Lowest	Median	Highest	
<b>Toxicity</b>	0.14%	0.19%	0.00%	0.13%	0.39%	0.00%	0.17%	0.31%	<b>0.01%</b>
<b>Profanity</b>	0.00%	0.02%	0.00%	0.00%	0.05%	0.00%	0.00%	0.03%	<b>0.00%</b>

Table 8: Average toxicity/profanity in the captions generated by PaLI-X on FairFace dataset.

**Bias / Demographic Parity.** We estimate the level of demographic parity (DP) [88] in PaLI-X with respect to gender and occupation. To estimate the level of demographic parity (DP) in the model’s output, we feed an image into PaLI-X with the chosen occupation title as a prefix and record the average log-perplexity score of the captions generated by the model. To ensure that any observed parity would likely reflect unintended biases in the model itself as opposed to the evaluation dataset, we use CelebA [89] that contains celebrity images with gender presentation annotation. Our assumption is that many occupations reflecting societal stereotypes, such as secretaries and plumbers, are quite rare in the CelebA dataset so disparities in output may reflect what is encoded in the model itself. The list of occupations is compiled based on [90] and the US job statistics report in [91].

Figure 3 (TOP) summarizes the overall results. First, PaLI-X tends to assign a higher log-perplexity score to women than men across most occupations; i.e. **men are predicted to be more likely to hold such occupations**. Second, PaLI-X assigns a higher likelihood for a woman to be ‘secretary’ &

Ethnicity	Toxicity			Profanity		
	< 0.2	0.2 – 0.8	> 0.8	< 0.2	0.2 – 0.8	> 0.8
Middle Eastern	64.24%	35.76%	0.00%	94.87%	5.13%	0.00%
Black	59.47%	40.40%	0.13%	92.67%	7.33%	0.00%
Indian	63.86%	36.07%	0.07%	94.39%	5.61%	0.00%
Hispanic	61.09%	38.79%	0.12%	94.45%	5.55%	0.00%
White	62.45%	37.16%	0.39%	92.85%	7.10%	0.05%
Southeast Asian	63.18%	36.61%	0.21%	93.57%	6.43%	0.00%
East Asian	63.15%	36.72%	0.13%	91.55%	8.45%	0.00%

Table 9: Distribution of the predicted toxicity/profanity for the captions generated by PaLI-X on FairFace dataset disaggregated by ethnicity.

Age	Toxicity			Profanity		
	< 0.2	0.2 – 0.8	> 0.8	< 0.2	0.2 – 0.8	> 0.8
< 19	58.78%	40.00%	0.22%	89.71%	10.29%	0.00%
20 - 29	63.01%	36.86%	0.12%	93.24%	6.73%	0.03%
30 - 39	63.13%	36.70%	0.17%	95.41%	4.59%	0.00%
40 - 49	63.62%	36.31%	0.07%	95.27%	4.73%	0.00%
50 - 59	65.87%	33.88%	0.25%	96.48%	3.52%	0.00%
60 - 69	65.31%	34.38%	0.31%	95.95%	4.05%	0.00%
> 70	66.10%	33.90%	0.00%	92.37%	7.63%	0.00%

Table 10: Distribution of the predicted toxicity/profanity for the captions generated by PaLI-X on FairFace dataset disaggregated by age.

Perceived Gender	Toxicity			Profanity		
	< 0.2	0.2 – 0.8	> 0.8	< 0.2	0.2 – 0.8	> 0.8
Predominantly Feminine	53.98%	45.93%	0.09%	90.55%	9.39%	0.07%
Predominantly Masculine	70.76%	29.17%	0.06%	94.97%	5.01%	0.01%

Table 11: Distribution of the predicted toxicity/profanity for the captions generated by PaLI-X on MIAP dataset disaggregated by perceived gender.

Age Bucket	Toxicity			Profanity		
	< 0.2	0.2 – 0.8	> 0.8	< 0.2	0.2 – 0.8	> 0.8
0-2 yrs	28.00%	72.00%	0.00%	69.90%	30.10%	0.00%
3-19 yrs	49.96%	49.96%	0.07%	91.46%	8.54%	0.00%
20-59 yrs	66.27%	33.68%	0.05%	93.42%	6.55%	0.03%
> 60 yrs	65.46%	34.54%	0.00%	96.39%	3.61%	0.00%

Table 12: Distribution of the predicted toxicity/profanity for the captions generated by PaLI-X on MIAP dataset disaggregated by age bucket.

‘actor’) and a higher likelihood for a man to be (‘guard’ & ‘plumber’) at the 95% confidence level. Figure 3 (BOTTOM) displays the corresponding correlations between perceived gender presentation and occupations within the WebLI dataset, where we use the Pearson correlation coefficient by treating each label as a binary random variable and noting that for binary random variables, zero correlation implies full independence. All absolute correlation coefficients in the data are  $< 0.2$  with 99% of them being  $< 0.1$ .

**Performance Disparity.** We present here an evaluation of how well PaLI-X performs across different subgroups using the MIAP [87] dataset. For images containing exactly a single individual, we query PaLI-X with the question: “Is there a person in this image?” and evaluate the accuracy of its response. Note that there are no false positives in this evaluation. Table 13 summarizes the results. We observe that PaLI-X maintains a high accuracy across all subgroups.

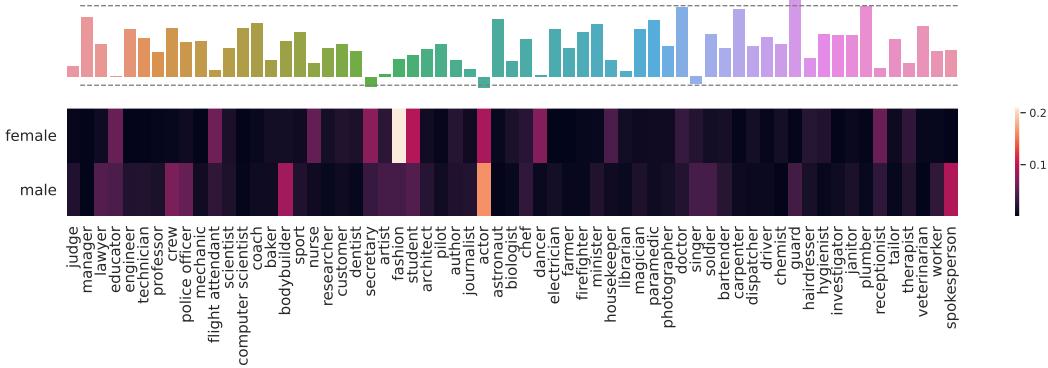


Figure 3: TOP: Level of demographic parity (DP) in PaLI-X’s output for CelebA images between women and men. Values close to zero indicate absence of bias. BOTTOM: *Absolute* Pearson correlation coefficients between gender presentation and occupations in WebLI.

Skin Tone	1 [2]	2 [871]	3 [3008]	4 [522]	5 [184]	6 [85]	7 [54]	8 [49]	9 [6]	10 [1]
	0.00%	0.11%	0.47%	1.53%	0.54%	1.18%	0.00%	0.00%	0.00%	0.00%
Gender	Predominantly Feminine [2437]					Predominantly Masculine [3544]				
	0.53%		0.85%							
Age Bucket	0-2 yrs [17]		3-19 yrs [568]		20-59 yrs [4925]		> 60 yrs [247]			
	0.00%		0.00%		0.77%		0.81%			

Table 13: Detection error rate for “person” in PaLI-X using the subset of the MIAP dataset [87] that contain exactly a single individual in the image. PaLI-X maintains a low error rate across all subgroups. Skin tone follows the Monk Skin Tone Scale [92]. Numbers inside square brackets correspond to the size of each bucket.

**Limitations.** The analysis carried out in this section is necessarily limited, since fairness is a societal concept that cannot be reduced to statistical metrics. We expect RAI evaluations to evolve over time as new issues are detected and reported in the literature and additional datasets become available. Statistical analysis is only a single step and does not substitute for studying the broad and delayed impact of deployed models.

In addition, we rely in some parts on automated tools for inferring attributes, which are not perfectly accurate and can lead to a broad categorization of people that misidentifies real identities. We do not support the creation or application of classifiers for sensitive attributes, such as gender or ethnicity, based on visual indicators and encourage readers to delve into the comprehensive work outlining their potential risks, such as [93, 94], for further insight. Also, while we use perceived gender presentation in our analysis that is provided by the data (i.e. in CelebA and FairFace), we acknowledge that people may express their gendered identities in numerous other ways.

In our evaluation, toxicity is predicted based on the generated captions only. However, without knowing the context of the image, this can introduce false positives.

## 6 Conclusions

In this work we draw more insights from further scaling vision and language models. We show that the scaling and the improved training recipe results in a model that substantially outperforms previous state-of-the-art models, leads to emergent behaviors and identifies further margins for improvements. In particular, we report that the model achieves significant improvements at document, chart, and infographic understanding, captioning, visual question answering, counting, and performs well on few-shot (in-context) captioning, video captioning and question-answering, and object detection.

## **Acknowledgements**

We would like to thank Sarah Laszlo, Kathy Meier-Hellstern, Caroline Pantofaru, Susanna Ricco, Candice Schumann, Ken Burke, Simon Wang, Rachel Hornung, Yichang Chen, Utsav Prabhu, Abhijit Ogale, Kristina Toutanova, Weicheng Kuo, Jihyung Kil, Xiangning Chen, Liang Chen, Rich Lee, Elizabeth Adkison, James Cockerille, Eric Ni, Erica Moreira, Victor Gomes, Jeremiah Harmsen, Claire Cui, Slav Petrov, Tania Bedrax-Weiss, Joelle Barral, Tom Duerig, Paul Natsev, Fernando Pereira, Jeff Dean, and Zoubin Ghahramani for helpful discussions, feedback, and support.

## A Additional Model Details and Examples

### A.1 PaLI-X Architecture Illustration

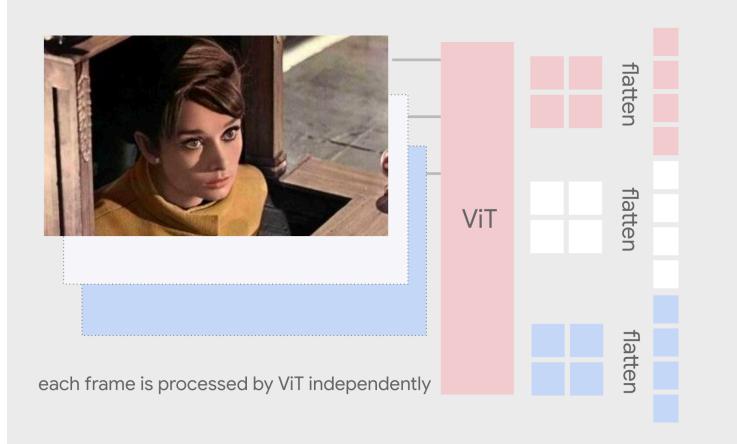


Figure 4: Visual input for videos: each frame is independently processed by ViT; patch embeddings are flattened and concatenated together to form the visual representation. (The example input image is in the public domain).

### A.2 Tuning ViT-22B for better OCR capabilities

The vision encoder’s ability to understand text is crucial to several downstream tasks and general usability. JFT-based pre-training is insufficient to cover this, and so we tuned ViT-22B on WebLI-OCR data. In order to stay true to the original discriminative classification-based objective used for ViT-22B, we turn OCR into a bag-of-words prediction task. OCR texts are tokenized using the mT5 tokenizer [95] across all languages, and the model is trained to predict whether or not a given token occurs in an image. This is treated as multilabel classification, with an expanded classification head.

In the ablation study shown in Table 22, we confirm that this extra tuning step indeed has a significant improvement on Scene-Text understanding capabilities, demonstrated by the performance on ST-VQA and TextVQA. Meanwhile, the performance on regular VQA tasks such as those in the VQAv2 benchmark also improves.

### A.3 Illustrative PaLI-X Examples

Table 14 shows representative examples of PaLI-X, illustrating improved abilities related to counting (both of the simple and complex variety), in context text-reading capabilities, and spatial awareness.



Image Credit: Wikimedia Commons [CC BY-SA 4.0]

Q: how many coins are there?

A: 12

Q: how many one-dollar coins are there?

A: 2

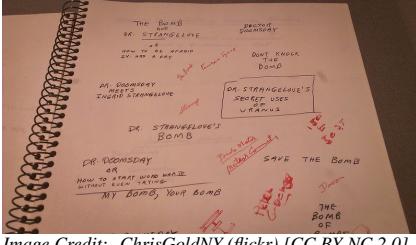


Image Credit: ChrisGoldNY (flickr) [CC BY-NC 2.0]

Q: what is written inside the box?

A: dr. stranglove's secret uses of uranus

Q: what is written on the top-left corner of the page?

A: the bomb and dr. strangelove

Q: what is written on the top-right corner of the page?

A: doctor doomsday



Public Domain Image

Captioning:

a movie poster for rebecca starring laurence olivier and joan fontaine.

Q: who directed this movie?

A: alfred hitchcock

Q: who produced this movie?

A: david o. seznick

Table 14: Examples of counting, text reading capabilities with context and spatial awareness. Results are generated by the multi-task-finetuned models using the model’s inherent OCR capabilities (i.e., without the use of an external OCR system).

## B Additional results: Image Captioning and VQA

### B.1 Information of Downstream Image Benchmarks

Table 15 summarizes the Image Captioning and VQA benchmarks. For benchmarks modeled only end-to-end without OCR pipeline input (Table 1 and Table 16), fine-tuning is performed with resolution 672×672. For Scene-Text and Document Understanding tasks presented in Table 2, fine-tuning is performed with resolution 756×756.

### B.2 Extended Tables of Image Benchmarks

An extended table of results on some Image Benchmarks is shown as Table 16.

Benchmark	Visual Domain	Description	Metric
COCO Captions		Captioning of natural images	CIDEr
NoCaps		Captioning of natural images	CIDEr
TextCaps		Captioning of natural images containing text	CIDEr
VizWiz-Cap		Captioning of photos taken by people who are blind	CIDEr
VQAv2		VQA on natural images	VQA accu.
OKVQA	Natural Images	VQA on natural images requiring outside knowledge	VQA accu.
TextVQA		VQA on natural images containing text	VQA accu.
VizWiz-QA		VQA on photos taken by people who are blind	VQA accu.
ST-VQA		VQA on natural images containing text	ANLS
TallyQA		VQA with counting questions	EM
OVEN		VQA on natural images for visual entity recognition	EM
InfoSeek		VQA on natural images for visual info-seeking questions	Relaxed EM
OCR-VQA		VQA on images of book covers	EM
ChartQA	Illustrations	VQA on images of charts	RA
AI2D		VQA on images of scientific diagrams	EM
DocVQA		VQA on images of scanned documents	ANLS
InfographicsVQA	Documents	VQA on images of infographics	ANLS
Screen2Words	UIs	Captioning a UI screen to describe functionality	CIDEr
Widget Captioning		Captioning a UI component on a screen	CIDEr

Table 15: Summary of Image Captioning and VQA benchmarks used for evaluating PaLI-X

Model	COCO	NoCaps		VQAv2		OKVQA	TallyQA	
	Karp.-test	val	test	test-dev	test-std	val	simple	complex
SimVLM	143.3	112.2	110.3	80.03	80.34	-	-	-
CoCa (2.1B)	143.6	122.4	120.6	82.3	82.3	-	-	-
GIT (0.7B)	144.8	125.5	123.4	78.56	78.81	-	-	-
GIT2 (5.1B)	145.0	126.9	<b>124.8</b>	81.74	81.92	-	-	-
OFA (0.9B)	145.3	-	-	82.0	82.0	-	-	-
Flamingo (80B)	138.1	-	-	82.0	82.1	57.8*	-	-
BEiT-3 (1.9B)	147.6	-	-	84.2	84.0	-	-	-
PaLM-E (562B)	138.7	-	-	80.0	-	<b>66.1</b>	-	-
MoVie	-	-	-	69.26	-	-	74.9	56.8
PaLI (17B)	149.1	<b>127.0</b>	124.4	84.3	84.3	64.5	81.7	70.9
PaLI-X (55B)	<b>149.2</b>	126.3	124.3	<b>86.0</b>	<b>86.1</b>	<b>66.1</b>	<b>86.0</b>	<b>75.6</b>

Table 16: Results on COCO Captions (Karpathy split), NoCaps, VQAv2, OKVQA, and TallyQA with end-to-end modeling without OCR pipeline input. The “simple” and “complex” are test subsplits.

### B.3 Multi-lingual Captioning

**Multilingual captioning on XM-3600** The Crossmodal-3600 (XM3600) benchmark contains a geo-diverse set of 3600 images with human-annotated reference captions in 36 languages [23]. Table 17 presents multilingual results for both PaLI (current SoTA on XM-3600) and PaLI-X, both finetuned with  $224 \times 224$  resolution. Overall, PaLI-X improves on the SoTA performance across 5 of the 7 languages we report here (and for 14 of the total 35 languages considered); notably, the performance on English is 4 CIDEr points lower compared to PaLI. The 35-language average CIDEr score is in the same ballpark between PaLI and PaLI-X, with a slight +0.5 advantage for PaLI.

Model	en	fr	hi	iw	ro	th	zh	35-lang avg.
PaLI	<b>98.1</b>	75.5	31.3	46.8	35.8	72.1	<b>36.5</b>	<b>53.6</b>
PaLI-X	94.2	<b>78.7</b>	<b>32.0</b>	<b>46.9</b>	<b>36.9</b>	<b>75.3</b>	36.1	53.1

Table 17: CIDEr scores on image captioning for the Crossmodal-3600 benchmark for seven diverse languages (English, French, Hindi, Hebrew, Romanian, Thai, and Chinese), as well as the average of the 35 languages covered by the benchmark. Both models are finetuned with  $224 \times 224$  resolution.

#### B.4 TallyQA and the emergence of complex counting capability

We present in Table 18 the performance of similar models across a wide range of capacity – from 700M parameters to 55B parameters for PaLI-X. The graphs in Fig. 5 illustrate how simple counting appears to follow a more linear progression as parameter-size increases, while complex counting appears to show emergence somewhere before the datapoint provided by the performance of PaLI 17B. This corresponds to our intuition that complex counting is a true multimodal task that requires additional capabilities from a model, in terms of the alignment that is required between the visual information and the prompt specification.

Model	TallyQA simple	TallyQA complex	Weighted average
PaLI (700M)	66.9	55.6	62.4
PaLI (3B)	72.0	56.7	65.9
PaLI (17B)	76.2	65.5	71.9
PaLI-X (55B)	81.3	71.0	77.2

Table 18: Performance on TallyQA splits for simple and complex questions. All models use  $224 \times 224$  image resolution.

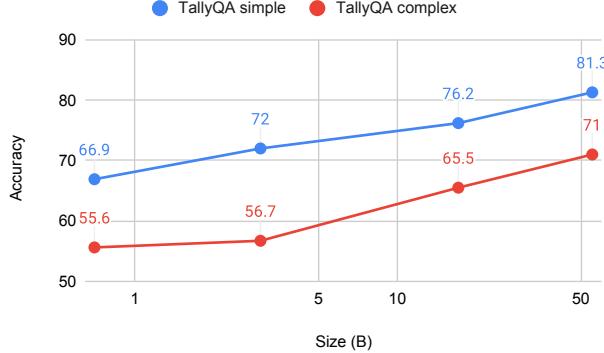


Figure 5: Performance on TallyQA splits for simple and complex using PaLI variants and PaLI-X. All models use  $224 \times 224$  image resolution. The emergent behavior on complex counting beyond the 3B size is made clear with PaLI-X.

#### B.5 Details on Few-shot Modeling

##### B.5.1 Few-shot Formulation

Figure 6 illustrates the network flow of a few shot model. The text and prompt part of each shot is embedded and concatenated as text features for the PaLI-X model. Each shot’s images and the target image are independently encoded by the ViT component, and the ViT features are concatenated along the sequence axis as visual features. Conditioned on that sequence, the PaLI-X decoder autoregressively makes the predictions for the target image.

**Encoder shot and Decoder shots** While images for all few-shot examples and target example are given as input to the model, text information can be provided in different ways. During inference time, all text information related to the few-shot examples is given to the encoder; in the case of a Multi-answer VQA task, for example, this includes both the prompts that contain the questions, and the expected answers. Prompt for the target example is also given to the encoder, and the decoder is tasked with generating an answer for the target example. During training, however, we increase the training efficiency by making the model predict answers for both the target example and selected shots (the *decoder shots*). That is, we partition the  $N$  shots in two sets: encoder shots ( $N_e > 0$ ) and decoder shots ( $N_d \geq 0$ ), such that  $N_e + N_d \leq N$ . We use up to 4 shots in total during pre-training (i.e.  $N = 4$ ), and sample  $N_e$  uniformly at random from 1 to  $N$ . Text input for encoder shots contain

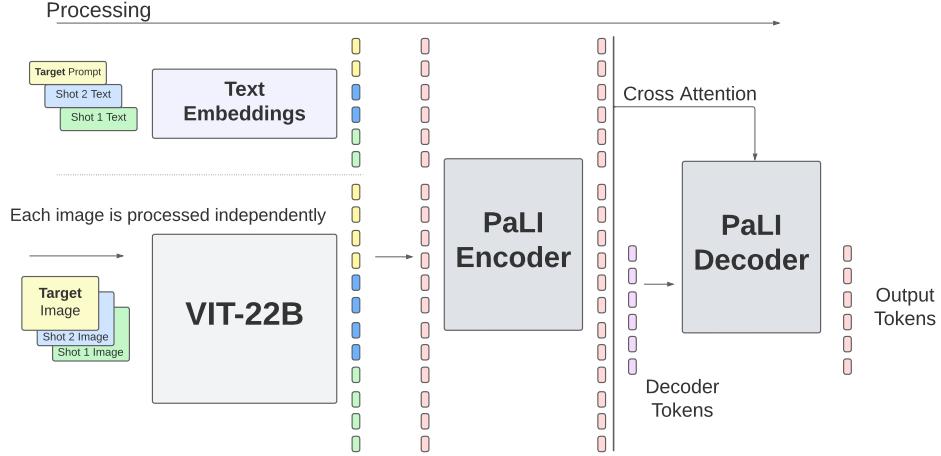


Figure 6: A detailed view on how the few-shot exemplars are fed to the model components.

both prompts and answers. The decoder shots, however, act as if they were target examples: their text input to the encoder contains only the prompt, and the decoder needs to predict answers for the decoder shots in addition to the target example.

**Attention re-weighting** Increasing the number of shots turned out to be challenging, potentially due to cross-attention to target example input tokens getting diluted by the large number of shots. To address this, we introduce an attention re-weighting mechanism. As shown in Figure 7, we explicitly boost the weights for cross attention between decoder tokens and encoded tokens from the target example (that is, the target image and the target text prompt).

Specifically, if there are  $N$  shots in total, when decoding each token we multiply the cross attention weights by  $N$  for the target image and text tokens from the encoder outputs. We observe this attention re-weighting technique is especially helpful when we provide the model with many shots (e.g. 32 shots). [96] introduces a technique along similar lines to manipulate attention weights when gathering them from different threads of encoded shots at inference time.

### B.5.2 Additional Few-shot Results

**Multilingual captioning results** Table 19 reports the CIDEr scores for 7 languages and an average over 35 languages to demonstrate PaLI’s multilingual captioning capabilities on the XM3600 benchmark in the few-shot setting. The pre-trained model (no few-shot finetuning) achieves an average score of 22.7. The PaLI-X model achieves an average score of 45.1 for 4 shots and 47.1 for 32 shots. Note that the 32-shot PaLI-X average CIDEr score is only 6 points behind the fully finetuned model, which uses roughly 600k training examples per language (while the few-shot approach does not update the model parameters).

**Qualitative results** Figure 8 shows 3 examples on few-shot captioning and VQA tasks for qualitative analysis. The first row shows captions for the images using the images’ original language,

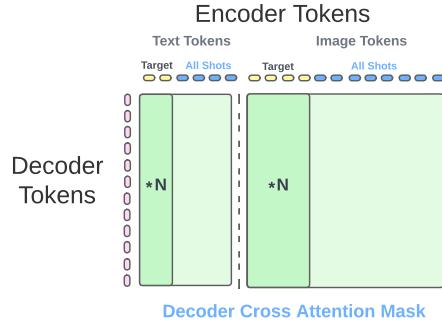


Figure 7: Re-weighted attention with few-shots.

<sup>5</sup>Equivalent with the Flamingo “0-shot” setting.

Input few shot examples + target image	Output
 <p>Public Domain Image The current population of Australia is 26 million people.</p>  <p>Wikimedia Commons [CC BY-SA 3.0] Deutschland hat 83 Millionen Einwohner.</p>  <p>Wikimedia Commons [CC BY-SA 2.0] Argentina tiene 46 millones de habitantes.</p>  <p>Wikimedia Commons [CC BY-SA 3.0]</p>	la population actuelle de la france est de 67 millions d'habitants
 <p>Tim Klapdor (flickr) [CC BY-NC 2.0] Home to the best burgers and fried chicken.</p>  <p>Wikimedia Commons [CC BY-SA 4.0] Home to unbeatable fish and chips.</p>  <p>Pelican (flickr) [CC BY-SA 2.0] Home to outstanding ramen.</p>  <p>Wikimedia Commons [CC BY-SA 2.5]</p>	Home to the best pizza in the world
 <p>Wikimedia Commons [CC BY-SA 3.0] Q: Is this a new technology? A: Yes, the typewriter is invented in 19 Century, fairly new.</p>  <p>Wikimedia Commons [CC BY-SA 3.0] Q: Is this a new technology? A: No, the compass is invented 2,000 years ago, pretty old.</p>  <p>Wikimedia Commons [CC BY-SA 2.0] Q: Is this a new technology? A: Hard to tell, the Newtonian telescope is invented around 17 century.</p>  <p>The IBM PC Gautier Poupeau (flickr) [CC BY 2.0] Q: Is this a new technology? A:</p>	yes the personal computer is invented in 1970s fairly new

Figure 8: Qualitative Results on few-shot captioning (first two rows) and VQA (the last row) tasks.

	Crossmodal-3600 Captioning							
	en	fr	hi	iw	ro	th	zh	35-lang avg.
PaLI-X 0-shot	48.8	25.0	10.5	20.1	13.0	33.3	18.4	22.7
PaLI-X (2 text-only shots <sup>5</sup> )	54.5	46.7	12.0	22.2	9.4	40.3	23.7	25.8
PaLI-X 4 shots	77.8	62.5	22.2	38.7	30.2	56.0	27.7	45.1
PaLI-X 32 shots	81.4	66.1	25.6	40.6	32.4	59.4	29.7	47.1
PaLI-X (finetuned)	94.2	78.7	32.0	46.9	36.9	75.3	36.1	53.1

Table 19: Few-shot performance of the PaLI-X model on multilingual captioning tasks.

demonstrating the cross multilingual transfer of the few-shot capability. The second row captions the images with a country’s popular food, showing that the few-shot approach can access the model’s world knowledge. The last row shows a VQA with an explanation-like scenario where we ask if the technologies in the images are “new”. Generally speaking, the shown personal computer was produced more than 40 years ago and could be regarded as old technology considering the fast pace of the current high-tech development. However, the 3 input shots provide the detailed calibration for the concept of “new” and the few-shot model successfully take the context and output “new” with plausible explanation to the very old PC.

### B.5.3 Few-shot ablation results

In this section, we present and discuss some ablation results for few-shot we explored in order to inform our final design choices on PaLI-X. Unless otherwise specified, we use a 700M-parameter model with the same encoder-decoder architecture, consisting of a ViT-B/16 vision encoder and a mT5-base encoder-decoder language model.

**Pooling vs not pooling image tokens** To mitigate the computational burden that arises with many shots, we can pool (for example, average) the per-image tokens before concatenating all input tokens. This pooled image tokens model achieved a CIDEr score of 56.3 for 4-shots COCO captioning, which is substantially lower than the full model’s CIDEr score of 61.7. This highlights the importance of keeping all the tokens coming out of the ViT encoder, despite the computational overhead.

**Limited-range Encoding Attention.** We explore per-example image-text attention, as proposed and applied in [10]. Under this approach, the image query tokens for each example can only attend

to its corresponding text tokens, while the text query tokens can attend to all tokens. By using this per-example attention model, we achieved a CIDEr score of 59.6, which is 2.1 points lower than the full attention model’s CIDEr score of 61.7 for 4-shots COCO captioning.

**Attention re-weighting for large number of shots.** We report the few-shot results on COCO captioning from early-stopped PaLI-2 3B models; in this case, we did not apply normalized attention in training. We provide the test results with and without attention re-weighting during *inference* for a different number of encoder shots. Attention re-weighting achieves increasing CIDEr scores of 82.1, 84.3 and 84.5 with 4, 8 and 16 shots respectively. On the other hand, the model achieves 83.4, 76.5 and 66.3 without attention re-weighting. The decreasing performance may suggest that the model fails to locate the target image and text prompt among the large number of shots, whereas the attention re-weighting helps the model to focus on the target features. Accordingly, we decided to include attention re-weighting during finetuning for PaLI-X.

**Distributing shots between encoder and decoder.** We explore the use of both encoder and decoder shots during pre-training. We pretrain the PaLI-2 700M model on PaLI-2 mixtures with varying number of encoder shots (between 1 and 4). The remaining shots (up to exactly 4) are used as decoder shots. Using only encoder shots leads to a 64.0 CIDEr score for 4 shots in COCO captioning. The best mix of encoder and decoder shots achieves a CIDEr score of 65.2. This suggests splitting shots leads to a more challenging pre-train task that helps the model learn more efficiently.

## B.6 Finetuning hyperparameters

The hyperparameter choices for downstream finetuning experiments are summarized in Table 20. As mentioned in the Main Text, for all of the downstream finetuning experiments, we used a reduced set of hyperparameters, without heavy per-task optimization.

Benchmark	learning rate schedule	Steps before LR decay to 0	batch size
COCO		10k	256
VQAv2		20k	256
OCRVQA		20k	256
Multitask-VQA	linear decay from 1e-4	20k	256
Multitask-Captioning		20k	256
All other		5k	128

Table 20: Hyperparameter used for finetuning PaLI-X.

## B.7 Multi-task finetuning

We deduplicated every training set mixture over the test sets of every task in order to prevent leakage of any test-set examples into the training set. The mixture is formed by putting the training examples of each subtask together, with heuristic adjustments for a better balance. Following the resolutions for the single-task finetuning, the multi-task captioning and VQA finetuning are done with 672 and 756 image resolutions, respectively. The multitask finetuning covers just about 5M examples, which is 20k steps with a batch size of 256. For scene-text and document understanding tasks, the multi-task finetuning uses the end-to-end setting without OCR pipeline input.

The following aspects made multitask finetuning particularly challenging: (i) all tasks used the same prompt without task-specific indicators; the model is thus required to adapt to the style of multiple benchmarks simultaneously. 2) We do not perform per-task validation set optimization. All subtasks are evaluated using the same checkpoint, but tasks converge to their optimal value at a different pace.

## B.8 Ablation studies

We first show in Table 22 the advantage brought by the OCR co-training stage of ViT-22B. We pair the vanilla ViT-22B and the ViT-22B with additional OCR co-training with a small language model mT5-base and pretrain these models on 40M of WebLI-OCR data with the splitOCR objective, before finetuning on ST-VQA. Co-training on image and OCR classification has a significant advantage on

Model	VQA v2	OK VQA	Text VQA	VizWiz VQA	ST VQA	OCR VQA	Info VQA	Doc VQA	Chart QA	Avg.
Split	test-dev	val	val	test-dev	val	test	test	test	test	-
Previous Multi-task SOTA	84.3	64.5	68.4	71.6	75.1	71.3	40.0	76.6	70.5	-
Single-task FT	<b>86.0</b>	<b>66.1</b>	<b>71.9</b>	<b>72.6</b>	<b>80.2</b>	<b>75.9</b>	49.2	80.0	<b>70.9</b>	-
Multi-task FT	84.3	63.5	71.4	71.4	79.0	73.4	<b>50.7</b>	<b>80.9</b>	70.6	-
Multi-task (+/-)	-1.7	-2.6	-0.5	-1.2	-1.2	-2.4	+1.5	+0.9	-0.3	-0.8

Table 21: Scores from multi-task finetuning compared with those from single-task finetuning for VQA. Validation or test-dev set numbers are reported for some tasks.

ST-VQA and TextVQA. In the meantime, the performance on VQAv2, which is not very scene-text heavy, is improved as well. Moreover, we found that making the top left patch white, which helped the co-training of image classification and ocr classification on ViT-22B, is not required for the subsequent training of PaLI-X.

For ablation of the PaLI-X training procedure, we used a 5B model with UL2-3B and ViT-G with 2B parameters, which is roughly a 10:1 down-scale of the PaLI-X 55B model.

Model	OCR-task Indicator	ST-VQA	TextVQA	VQAv2	3-task avg.
mT5-base + Vanilla ViT-22B	No	42.6	36.1	68.9	49.2
mT5-base + ViT-22B-OCR	No	<b>47.0</b>	38.9	69.8	<b>51.9</b>
mT5-base + ViT-22B-OCR	Yes	46.2	<b>39.4</b>	<b>70.2</b>	<b>51.9</b>

Table 22: Advantage of the OCR co-training stage of ViT-22B. Pretraining is performed with resolution 224×224 and finetuning is with 448×448. Numbers reported are on validation split.

For stage 1 training, we show in Table 23 that adding image token generation does not harm the performance on the main image+language understanding tasks.

Mixture	COCO	VQAv2
without ViT-VQGAN	139.3	77.3
with 10% ViT-VQGAN	139.7	77.1

Table 23: Ablation experiment showing adding ViT-VQGAN tokens does not harm understanding performance (captioning and VQA tasks).

## C Additional results: Video Captioning and QA

Below we give a brief description of each video data set we used for evaluation. Note that we freshly collected the data when performing the experiments, which led to different effective numbers of videos in different splits in some cases, see Table 24.

These descriptions refer to the original dataset size, but we train on (sometimes significantly) fewer videos — the exact numbers are given in Table 24. This is because not all videos in the datasets were available online at the time of writing (e.g., due to user deletion).

### C.1 Datasets & Benchmarks

**MSR-VTT [55]:** This dataset consists of 10K open domain video clips for video captioning, with 20 captions each. The duration of each video clip is between 10 and 30 seconds. We follow the standard splits proposed by [55] and report results on the test set.

**VATEX [56]:** VATEX includes captions for 41K videos sampled from the Kinetics-600 dataset, with 10 English captions each. We report results on the English public test set.

**ActivityNet Captions [57]:** This dataset consists of 100K temporally localized sentences for 20k videos. We follow the standard split containing 50/25/25% of the dataset for training, validation and testing, and use ground truth temporal proposals at evaluation following [57]. Note that following other works [62], we use the val\_1 split for validation and val\_2 split for testing.

**Spoken Moments in Time (SMIT) [58]:** This dataset consists of long captions obtained via audio recordings for 500k short video clips. While this dataset has been traditionally only used for text to video retrieval, we find that it is a strong benchmark for captioning as it is the largest manually annotated set of videos with text captions.

**ActivityNet-QA [61]:** The dataset contains 58,000 question-answer pairs for videos in the ActivityNet dataset [97]. We report accuracy (using exact string match) on the test split. Note that we do open-ended generation for all VideoQA datasets.

**MSR-VTT-QA [60]:** This dataset was created using a semi-automatic pipeline on top of the MSR-VTT dataset. We report accuracy (using exact string match) on the test split.

**NExT-QA [59]:** We focus on the Open-Ended QA task, which consists of 52,044 question-answer pairs for a total of 5,440 videos (sampled from the VidOr dataset[98]). Exactly following Next-QA [59] and Flamingo [10], we report the Wu-Palmer Similarity (WUPS) on the test set.

		MSR-VTT	VATEX	ANet-Cap	SMIT	M-V-QA	ANet-QA	NExT-QA
Original size	train	6513	25991	37421	481094	158581	32000	37523
	valid.	497	3000	17505	14604	12278	18000	5343
	test	2990	6000	17031	3513	72821	8000	9178
Dataset size	train	4768	22902	30982	481094	116943	28020	37523
	valid.	327	2657	14604	8096	8215	15890	5343
	test	2144	5276	14234	3513	53014	7050	9178
% Remaining	train	73.21	88.12	82.79	100.00	73.74	87.56	100.00
	valid.	65.79	88.57	83.43	100.00	66.91	88.28	100.00
	test	71.71	87.93	83.58	100.00	72.80	88.13	100.00

Table 24: We freshly collect the data sets from the respective data sources. In cases where there are multiple question-answer pairs per video we report the number of question-answer pairs. Similarly, for ActivityNet Captions we report the number of captions. Due to missing videos which were removed after the original data sets were defined, most of our data sets are missing 10% of the videos or more.

## D Additional results: Image Classification

**Setup for zero-shot and finetuning evaluation** The setup used for the experiments here uses the PaLI-X model to generate directly the (English) class name using the captioning prompt. The output is considered correct if it matches exactly the class name (apart from ImageNet-REAL, where we check if the class corresponding to the output is in the set of correct labels).

**Zero-shot Evaluation results** We use the same scoring technique as in PaLI [5] to evaluate PaLI-X in zero-shot setting (without training on any Imagenet data). We use the PaLI-X model obtained after the first stage of training (using the base 224 image resolution).

The results are presented in Table 25. We compare the results to PaLI [5] - previous zero-shot generative SOTA, and Flamingo [10] - another generative model of similar architecture with comparable 1-shot and 5-shot results. Overall, we report that the results between PaLI and PaLI-X for 0-shot are similar.

Model (ImageNet data)	INet	REAL	INet-R	INet-A	INet-Sketch	INet-v2	ObjNet
Flamingo-80B (1-shot)	71.9	-	-	-	-	-	-
Flamingo-80B (5-shot)	77.3	-	-	-	-	-	-
PaLI (17B) (0-shot)	<b>72.11</b>	<b>76.43</b>	81.97	44.70	<b>63.83</b>	<b>64.46</b>	42.62
PaLI-X (0-shot)	71.16	75.75	<b>82.96</b>	<b>46.13</b>	61.58	63.91	<b>44.58</b>

Table 25: Top 1 accuracy results of 0-shot image classification on ImageNet [66], ImageNet-REAL [67], ImageNet-R [68], ImageNet-A [69], ImageNet-Sketch [70], Imagenet-v2 [71] and ObjectNet [99].

**Finetuning** To test image classification capabilities, we finetune PaLI-X on ImageNet [66] and evaluate the resulting model on ImageNet-REAL [67] and out-of-distribution datasets: ImageNet-R [68], ImageNet-A [69], ImageNet-Sketch [70], ImageNet-v2 [71].

We use the model from the first training stage (at resolution 224) and the one from the last training stage (at resolution 756). We use the same training hyperparameters for all of runs (selected without any hyperparameter tuning).

The results can be seen in Table 26. We compare the results to generative model with open vocab – GiT2 [9] (using 384 image resolution), which is the current SOTA for full-finetuning on ImageNet. PaLI-X achieves close to SOTA results for generative models on Imagenet, and other datasets.

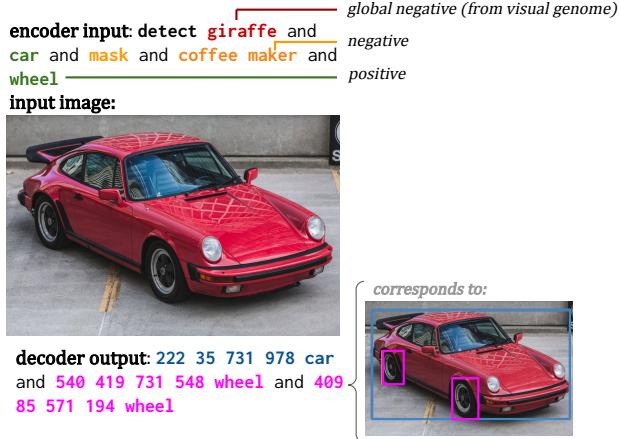
Model (resolution)	INet	REAL	INet-R	INet-A	INet-Sketch	INet-v2
GIT2 (384)	<b>89.22</b>	-	-	-	-	-
PaLI 3B (224)	85.11	88.71	<b>81.11</b>	45.71	70.00	78.23
PaLI 17B (224)	86.13	88.84	78.21	50.00	71.21	78.91
PaLI-X (224)	88.22	90.36	77.66	55.97	72.56	81.42
PaLI-X (756)	88.82	90.80	79.97	<b>73.47</b>	<b>73.39</b>	83.48
PaLI-X <sup>†</sup> (756)	89.19	<b>90.98</b>	80.06	72.57	73.37	<b>83.66</b>

Table 26: Classification (top-1) accuracy with Imagenet [66] fine-tuning on: ImageNet, ImageNet-REAL [67], ImageNet-R [68], ImageNet-A [69], ImageNet-Sketch [70], Imagenet-v2 [71] (resolution in parentheses). PaLI-X <sup>†</sup> fine-tuned for 2.2x more steps.

## E Object Detection

### E.1 Object detection as a VLM task

Object detection is framed similarly to Pix2seq [72], with two key differences: the use of a natural language vocabulary, and class-conditioning. Prompt classes are fed to PaLI-X’s text encoder, in the format `detect class1 and class2 and class3`. The model is trained to only output bounding boxes corresponding to classes in this prompt. We represent bounding boxes as coordinates in the same style as pix2seq [72]; that is, 4 integers  $y_{\min} \ x_{\min} \ y_{\max} \ x_{\max}$  ranging from 0 to 999. Figure 9 shows an example input.



*Image credits: Matthew Henry, burst, <https://burst.shopify.com/photos/vintage-red-porsche>*

Figure 9: An example training pair, consisting of the text prompt, the image and the expected output. The prompt consists of multiple classes; we show a hypothetical Open Images V4 example, with positives ‘car’ and ‘wheel’, negative ‘giraffe’ and global negatives ‘mask’ and ‘coffee maker’ (sampled from the visual genome label space).

**Prompt sampling hyperparameters** During training, a prompt for each example. We construct prompts from three pieces of information:

- *Positives*: These are the bounding boxes for objects definitely present in the image. During training, per example we sample  $p^+ \sim \mathcal{U}(0, P_{\max}^+)$ , and keep that proportion of positives.
- *Negatives*: These are the known instance negatives i.e. bounding boxes for objects definitely not present. For exhaustively labelled datasets like COCO, this is simply classes not labelled as positives. For non-exhaustively labelled datasets like LVIS, these are the classes not labelled as positives, which were presented to raters. During training sample  $f^- \sim \mathcal{U}(0, 5.0)$ , and use up to  $f^- \times n^+$ , where  $n^+$  is the number of positives after sampling  $p^+$ .
- *Global negatives*: These are negatives which are not explicitly labelled as negatives. They are taken from a wider label space combining multiple detection datasets. For a given example, valid global negatives consist of classes from the wider label space not explicitly labelled as positives or negatives. During training, we sample  $f^{GN} \sim \mathcal{U}(0, 5.0)$  and append  $f \times n_+$  global negatives, where  $n_+$  is the number of positives after sampling  $p^+$ .

By default, the combined label spaces of Visual Genome, Objects365 and OpenImagesV4 was used as the global label space, with the exception of detection finetuning, where LVIS and COCO label spaces were also added.

We truncate the number of total classes to  $n_{\max}$ .  $n_{\max}$  and  $P_{\max}^+$  are tuned per dataset to meet sequence lengths. After truncation, we shuffle classes in the prompt.

## E.2 Preprocessing

During pre-training, data is preprocessed to remove all LVIS-rare labels, following the protocol of OwlViT [28]. This is not done for detection finetuning. Images are randomly flipped horizontally, and randomly resized to between  $0.3$  and  $2.0 \times$  their original size, followed by selecting a random square crop of the current training resolution. If the image is resized to be smaller than the current resolution, it is left as is. Images are finally padded to a square.

## E.3 Licenses and attribution for images used in Main Text Figure 2

- Watermelon: Credit: Sarah Pflug  
<https://burst.shopify.com/photos/cutting-watermelon>.
- Bowls:  
<https://www.flickr.com/photos/ariesandrea/502826051/> CC-BY-NC-ND 2.0
- Business cat Credit: Sarah Pflug,  
<https://burst.shopify.com/photos/business-cat-in-office>
- Wall Credit: Matthew Henry  
<https://burst.shopify.com/photos/man-walking-in-front-of-this-is-paradise-wall?c=urban-life>

## References

- [1] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.
- [2] Tom B. Brown, Benjamin Mann, Nick Ryder, Jared Kaplan, Melanie Subbiah, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Christopher Hesse Clemens Winter, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *NeurIPS*, 2020.
- [3] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellar, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In *ICML*, 2022.
- [4] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Diaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellar, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu,

Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. Palm 2 technical report, 2023.

- [5] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Alexander Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. **PaLI: A jointly-scaled multilingual language-image model**. In *ICLR*, 2023.
- [6] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Patrick Collier, Alexey Gritsenko, Vighnesh Birodkar, Cristina Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. **Scaling vision transformers to 22 billion parameters**. In *ICML*, 2023.
- [7] Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. **UL2: Unifying language learning paradigms**. In *ICLR*, 2023.
- [8] Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhui Chen, Nigel Collier, and Yasemin Altun. DePlot: One-shot visual language reasoning by plot-to-table translation. *arXiv preprint arXiv:2212.10505*, 2022.
- [9] 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. *TMLR*, 2022.
- [10] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.
- [11] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023.
- [12] OpenAI. Gpt-4 technical report, 2023.
- [13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [14] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.
- [15] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.
- [16] Denny Zhou, Nathanael Schärlí, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models. In *ICLR*, 2023.
- [17] Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.
- [18] Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. Meta-learning via language model in-context tuning. In *ACL*, 2022.
- [19] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-IO: A unified model for vision, language, and multi-modal tasks. In *ICLR*, 2023.
- [20] Gang Li and Yang Li. Spotlight: Mobile UI understanding using vision-language models with a focus. In *ICLR*, 2023.

- [21] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. In *ICLR*, 2022.
- [22] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.
- [23] Ashish V. Thapliyal, Jordi Pont-Tuset, Xi Chen, and Radu Soricut. Crossmodal-3600: A massively multilingual multimodal evaluation dataset. In *EMNLP*, 2022.
- [24] Jihyung Kil, Soravit Changpinyo, Xi Chen, Hexiang Hu, Sebastian Goodman, Wei-Lun Chao, and Radu Soricut. PreSTU: Pre-training for scene-text understanding. *arXiv preprint arXiv:2209.05534*, 2022.
- [25] Soravit Changpinyo, Doron Kukliansky, Idan Szpektor, Xi Chen, Nan Ding, and Radu Soricut. All you may need for VQA are image captions. In *NAACL*, 2022.
- [26] AJ Piergiovanni, Weicheng Kuo, and Anelia Angelova. Pre-training image-language transformers for open-vocabulary tasks. In *T4V: Transformers for Vision Workshop, Conference on Computer Vision and Pattern Recognition*, 2022.
- [27] Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In *ICML*, 2023.
- [28] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection with vision transformers. In *ECCV*, 2022.
- [29] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *ICLR*, 2022.
- [30] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.
- [31] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8948–8957, 2019.
- [32] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. TextCaps: a dataset for image captioning with reading comprehension. In *European conference on computer vision*, pages 742–758, 2020.
- [33] Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 417–434. Springer, 2020.
- [34] Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. Screen2Words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology, UIST ’21*, page 498–510, New York, NY, USA, 2021. Association for Computing Machinery.
- [35] Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. Widget Captioning: Generating natural language description for mobile user interface elements. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5495–5510, Online, November 2020. Association for Computational Linguistics.
- [36] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.
- [37] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.
- [38] Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8076–8084, 2019.

- [39] 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.
- [40] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. VizWiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018.
- [41] Ali Furkan Biten, Rubèn Tito, Andrés Mafla, Lluís Gomez, Marçal Rusiñol, C.V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. Scene text visual question answering. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4290–4300, 2019.
- [42] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019.
- [43] 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.
- [44] 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.
- [45] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016.
- [46] Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of ACL*, 2022.
- [47] Hexiang Hu, Yi Luan, Yang Chen, Urvashi Khandelwal, Mandar Joshi, Kenton Lee, Kristina Toutanova, and Ming-Wei Chang. Open-domain visual entity recognition: Towards recognizing millions of wikipedia entities. *arXiv preprint arXiv:2302.11154*, 2023.
- [48] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? *arXiv preprint arXiv:2302.11713*, 2023.
- [49] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhajit Som, et al. Image as a foreign language: BEiT pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.
- [50] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. In *arXiv*, 2023.
- [51] Duy-Kien Nguyen, Vedanuj Goswami, and Xinlei Chen. Revisiting modulated convolutions for visual counting and beyond. *ICLR*, 2021.
- [52] Yixuan Qiao, Hao Chen, Jun Wang, Yihao Chen, Xianbin Ye, Ziliang Li, Xianbiao Qi, Peng Gao, and Guotong Xie. Winner team Mia at TextVQA challenge 2021: Vision-and-language representation learning with pre-trained sequence-to-sequence model. *arXiv preprint arXiv:2106.15332*, 2021.
- [53] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikanth Appalaraju, and R. Manmatha. LaTr: Layout-aware transformer for scene-text VQA. In *CVPR*, 2022.
- [54] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. Unifying vision, text, and layout for universal document processing. In *CVPR*, 2023.
- [55] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

- [56] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. VATEX: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, 2019.
- [57] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.
- [58] Mathew Monfort, SouYoung Jin, Alexander Liu, David Harwath, Rogerio Feris, James Glass, and Aude Oliva. Spoken Moments: Learning joint audio-visual representations from video descriptions. In *CVPR*, 2021.
- [59] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021.
- [60] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *MM*, 2017.
- [61] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. ActivityNet-QA: A dataset for understanding complex web videos via question answering. In *AAAI*, 2019.
- [62] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6847–6857, 2021.
- [63] Feng Cheng, Xizi Wang, Jie Lei, David Crandall, Mohit Bansal, and Gedas Bertasius. VindLU: A recipe for effective video-and-language pretraining. In *CVPR*, 2023.
- [64] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17959–17968, 2022.
- [65] Mario Fritz Mateusz Malinowski. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NeurIPS*, 2014.
- [66] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009.
- [67] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with ImageNet? *arXiv preprint arXiv:2006.07159*, 2020.
- [68] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.
- [69] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021.
- [70] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019.
- [71] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *International Conference on Machine Learning*, pages 5389–5400, 2019.
- [72] Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey E. Hinton. Pix2Seq: A language modeling framework for object detection. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.
- [73] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5356–5364. Computer Vision Foundation / IEEE, 2019.

- [74] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022.
- [75] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. Regionclip: Region-based language-image pretraining. In *CVPR*, 2022.
- [76] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *ECCV*, 2018.
- [77] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334), 2017.
- [78] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *EMNLP*, 2017.
- [79] Zeyu Wang, Clint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *CVPR*, 2020.
- [80] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *FAccT*, 2018.
- [81] Jessica Deuschel, Bettina Finzel, and Ines Rieger. Uncovering the bias in facial expressions. *arXiv preprint arXiv:2011.11311*, 2020.
- [82] Ben Hutchinson, Jason Baldridge, and Vinodkumar Prabhakaran. Underspecification in scene description-to-depiction tasks. In *AACL / IJCNLP*, 2022.
- [83] Terrance De Vries, Ishan Misra, Changhan Wang, and Laurens Van der Maaten. Does object recognition work for everyone? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 52–59, 2019.
- [84] Priya Goyal, Adriana Romero Soriano, Caner Hazirbas, Levent Sagun, and Nicolas Usunier. Fairness indicators for systematic assessments of visual feature extractors. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 70–88, 2022.
- [85] Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1548–1558, 2021.
- [86] Alyssa Lees, Vinh Q Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. A new generation of perspective API: Efficient multilingual character-level transformers. In *KDD*, 2022.
- [87] Candice Schumann, Susanna Ricco, Utsav Prabhu, Vittorio Ferrari, and Caroline Rebecca Pantofaru. A step toward more inclusive people annotations for fairness. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2021.
- [88] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science*, 2012.
- [89] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.
- [90] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.
- [91] Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [92] Ellis Monk. Monk skin tone scale, 2019.
- [93] Foad Hamidi, Morgan Klaus Scheuerman, and Stacy M Branham. Gender recognition or gender reductionism? the social implications of embedded gender recognition systems. In *Proceedings of the 2018 chi conference on human factors in computing systems*, pages 1–13, 2018.
- [94] Os Keyes. The misgendering machines: Trans/hci implications of automatic gender recognition. *Proceedings of the ACM on human-computer interaction*, 2(CSCW):1–22, 2018.

- [95] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, June 2021.
- [96] Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv preprint arXiv:2212.06713*, 2022.
- [97] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.
- [98] Xindi Shang, Donglin Di, Junbin Xiao, Yu Cao, Xun Yang, and Tat-Seng Chua. Annotating objects and relations in user-generated videos. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 279–287, 2019.
- [99] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Joshua Tenenbaum, and Boris Katz. ObjectNet: a large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 9453–9463, 2019.