FlexCap: Describe Anything in Images in Controllable Detail

Debidatta Dwibedi
Google Deepmind
debidatta@google.com

Vidhi Jain*
Carnegie Mellon University
vidhij@andrew.cmu.edu

Jonathan Tompson
Google Deepmind
tompson@google.com

Andrew Zisserman
Google Deepmind
zisserman@google.com

Yusuf Aytar
Google Deepmind
yusufaytar@google.com

Abstract

We introduce FlexCap, a vision-language model that generates region-specific descriptions of varying lengths. FlexCap is trained to produce length-conditioned captions for input boxes, enabling control over information density, with descriptions ranging from concise object labels to detailed captions. To achieve this, we create large-scale training datasets of image region descriptions with varying lengths from captioned web images. We demonstrate FlexCap's effectiveness in several applications: first, it achieves strong performance in dense captioning tasks on the Visual Genome dataset. Second, we show how FlexCap's localized descriptions can serve as input to a large language model to create a visual question answering (VQA) system, achieving state-of-the-art zero-shot performance on multiple VQA benchmarks. Our experiments illustrate FlexCap's utility for tasks including image labeling, object attribute recognition, and visual dialog. Project webpage: `https://flex-cap.github.io`.

1 Introduction

How does one describe the world around us, not just in broad strokes but with the ability to zoom in and out, capturing both the grand scene and the minute details? Imagine pointing at a bustling market scene and asking, "What's happening here?" and receiving a vivid description, not just of the market as a whole, but also a detailed account of the interactions between vendors and customers, the vibrant colors of the goods on display, or even a specific item held by a passerby. This ability to controllably focus and describe visual content is what we call *flexible captioning*.

Traditional image captioning models, while adept at capturing the gist of an image, often struggle to pinpoint specific objects or attributes. On the other hand, object detection systems excel at localizing elements but may lack the vocabulary to describe them comprehensively. Dense captioning [26] attempts to bridge this gap by generating captions for multiple regions, but its expressiveness is limited by existing datasets. In this work, we introduce a model called FlexCap that bridges the gap between holistic image understanding and localized inquiry. This enables the generation of captions that are both spatially precise and semantically rich (see Fig. 1 (left)) by specifying a region of interest and the desired level of detail in terms of number of words in the predicted caption. This allows us to integrate the strengths of image captioning, object detection, and dense captioning into one model.

To be able to train such a model, we require a dataset of images where many boxes are labeled with short and long descriptions. We propose a method to generate triplets of (i) image, (ii) a proposed region within the image, and (iii) a caption of a particular length, by using an open-vocabulary object detector to label regions from captions of an image-text pair dataset. We demonstrate this at two

*Work done as a student researcher at Google Deepmind.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-3530

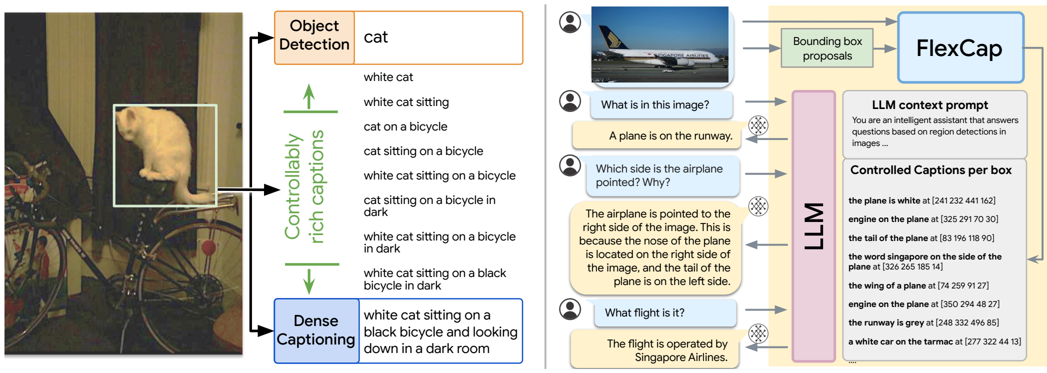


Figure 1: **FlexCap** generates controllably rich localized descriptions for any region in an image as shown on the left. It has the flexibility to produce captions in a controllable manner which allows the full spectrum of valid descriptions to be explored from short object category names to fully-detailed captions. On the right, we demonstrate that rich localized captions generated by FlexCap, when coupled with large language models (LLMs), enable zero-shot visual question answering.

scales: 200 million triplets using YFCC100M [47] captioned images; and 32 billion triplets using the WebLI [9] captioned dataset. Training FlexCap on these datasets enables the model to generate spatially and semantically rich representations (bounding boxes and their descriptions) that focuses on objects, their attributes, and their contextually changing descriptions.

We evaluate the effectiveness of the descriptions generated with FlexCap by representing images in detail by describing its constituent objects and regions. A popular and effective strategy is to directly provide visual features as input tokens to LLMs [2, 9, 14, 29, 30, 33, 59]. Instead we directly provide textual descriptions of images as input to the LLM to evaluate FlexCap similar to [21, 42, 57]. We show that this human interpretable representation, when combined with the power of LLMs, enables visual question answering and dialog. An example is shown in Figure 1(right). We also demonstrate that this combination can result in performance that is competitive with state-of-the-art VLM models on zero-shot image and video question answering benchmarks.

Our key technical contributions are: (i) controllable localized visual descriptions, using word count as a proxy for complexity to modulate the output of a generative language model (ii) a large-scale dataset generated for image-text-box triplets to enable training of our model; (iii) the human-interpretable representations produced by FlexCap with the help of LLMs is comparable or exceeds performance of state-of-the-art VQA methods; and (iv) demonstrating that our dense captioning methods outperform baselines under comparable scenarios.

2 FlexCap

Length conditioning. For the same region in an image, there may be multiple valid captions. In the input image shown in Figure 2, all the following descriptions are correct: *cat*, *grey and white cat*, *grey and white cat lying on shoes*. Clearly, the task of describing a bounding box in the image does not have only one right answer. We equip the model with the capability of producing outputs of a desired length by utilizing the idea of *length* conditioning. We condition the input by simply using an additional token indicating the desired length of the output caption. Training with length-conditioning is useful for three key reasons. First, the number of words used to describe is often proportional to the information content. We train the model to predict the next word in the sequence while accounting for the desired length, thereby the model learns to modulate the amount of information in the generated text. Second, length conditioning allows users to control the output of the model, further enabling the use of a single model for many diverse tasks. Third, the length prefix provides a better conditioned initial state for the captioner. Figure 3 shows how the same box might have more than one ground truth caption `<s> a dog <e>` or `<s> dog playing with a frisbee <e>`. If we use the first caption as ground truth and the words `<s> a dog` as the seen text, the next-word prediction loss encourages the model to increase its score for the `<e>` token and decrease the score for the word `playing` due to the softmax loss. Whether this occurrence will be a problem depends on the dataset statistics. To quantify the prevalence of this problem, we compute a statistic: for each image box, we consider all pairs of captions and measure the fraction of pairs sharing prefix words. For instance, consider a box that has three captions: `<s> dog <e>`, `<s> dog playing <e>`, and

https://doi.org/10.52202/079017-3530

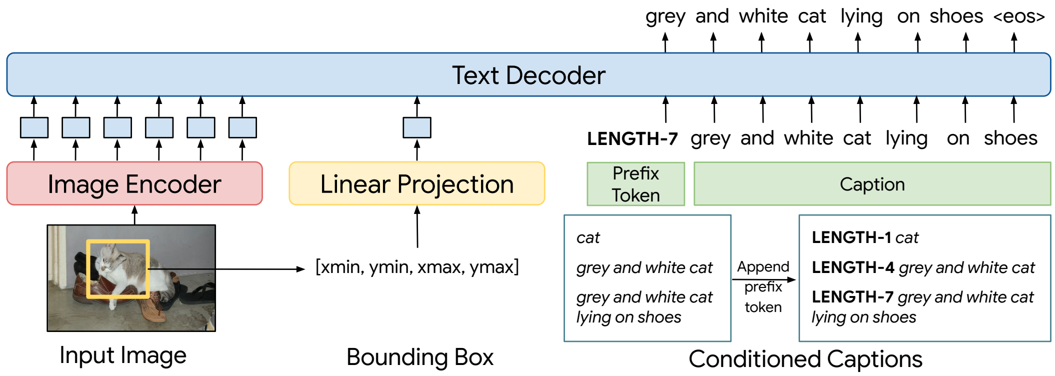


Figure 2: **Architecture and Training Setup.** FlexCap outputs a length-controlled caption of the object contained in the bounding box by taking (left) an image, (middle) coordinates of a bounding box and (right) the length prefix and caption, as inputs. The training loss is the standard next-word prediction loss that is used to train image captioning models.

`<s> dog playing with a frisbee <e>` which share a prefix `<s> dog`. While another caption for the same box, `<s> brown dog <e>`, does not share a prefix with captions beginning with `<s> dog <e>`. After averaging this metric across all images in our Localized Captions Dataset from WebLI (introduced later in Section 3), we found that 30.8% of all caption pairs share a prefix. Using a length conditioning token instead of `<s>`, the probability of prefix matching decreases from 30.8% to 11.1%. The length conditioning helps the model in distinguishing between captions with the same prefix while also providing the model with a novel capability during inference.

Architecture. Our objective is to train a model that takes as input an image and a region of interest and outputs a description *of a desired length* of the region spanned by the box. We present FlexCap's architecture in Figure 2. The model takes an image, the coordinates of a bounding box and the conditioning tokens as input, and outputs a textual description of visual contents within the specified bounding box. Our model mainly consists of an image encoder (i.e. SOViT-400M/14 [1]) and a transformer-based text-decoder. We pass the image through the vision model to produce outputs of dimensions $n \times d$ (where n is the number of patches and d is the embedding size). We pass the bounding box coordinates (of dimensions 1×4) through a linear layer to produce the coordinate features (of dimension $1 \times d$). Both vision features and normalized bounding box features are concatenated to form input of dimension $(n + 1) \times d$ to a text decoder. The text decoder consists of a stack of L Transformer layers. We use a decoder-only architecture in which all the vision and bounding box tokens remain unmasked but the text tokens are masked in a causal manner to enable next-word prediction training. We add all the vision tokens and bounding box coordinate tokens so that the text decoder can access all of the visual context in the image and the exact bounding box location. In this work, we train a text decoder composed of 12 self-attention transformer layers with a dimensionality of 768 and 12 attention heads.

In total, FlexCap has 590M parameters with 428M comprised of the image encoder (SOViT) and the remaining parameters in the text decoder. A linear layer transforms the 1152-dimensional output from the vision encoder into a 768-dimensional input for the text decoder. We initialize the vision encoder with SigLIP [60] weights, which is a contrastively trained image encoder using web-scale vision-text pairs from WebLI. We do not freeze the vision encoder during training.

Loss. We train FlexCap to predict the next token of the text. The text tokens are prefixed with the desired length of the caption and appended with an end of sentence token `<e>` to indicate the end of the caption. The target text tokens are obtained by shifting the padded text by 1. This is common training methodology for training generative language models like GPT [8] or SimVLM [51]. The loss is a classification loss over all the words present in the vocabulary. The loss is ignored over the padded tokens that are used to keep the size of the outputs same for all the captions in the batch. Formally, we represent a data sample as a triplet $T = (X, B, W)$ consisting of image X, bounding box B and captions W, where $W = \{\text{LENGTH-K}, w_1, w_2, ...w_k\}$. To enable batch training, we pad the tokenized captions to a fixed size M. For a given data triplet, our objective is to maximize the following log-likelihood. $l(X, B, W) = \sum_{i=1}^{M} \log p(w_i | w_{<i}, X, B)$. Assume that we have a dataset

 https://doi.org/10.52202/079017-3530

$\mathcal{D} = \{T_1, T_2, ..., T_N\}$. The overall loss function is:

$$L(D) = \sum_{j=1}^{N} l(X_j, B_j, W_j) = \sum_{j=1}^{N} \sum_{i=1}^{M} \log p((w_j)_i | (w_j)_{<i}, X_j, B_j)$$

Implementation. We implement this model using the JAX framework [7]. We train the entire model for about $400K$ steps using the AdamW optimizer with a cosine learning rate schedule. The maximum learning rate is 1.6×10^{-4} with 10K warm-up steps. We use a weight decay of 0.05. We train with a batch size of 4096 and image resolution of 224×224. We use a maximum text sequence length of 32 tokens. For each image in the batch, we sample a maximum of 8 bounding boxes.

Inference. At inference time, we provide an image, the target bounding box, and the desired length as input. We then decode in an auto-regressive manner till the end of caption token `<e>` is encountered or the maximum number of decoding steps is reached. While we can use standard sampling techniques used in text-generation like beam search, temperature sampling, or nucleus sampling [19] to generate multiple captions. We use greedy decoding in all experiments unless otherwise stated.

3 Localized Captions Dataset

In order to train the FlexCap model, we build a large scale dataset of image region descriptions of varying lengths. In the following section we describe how we produce such a dataset from existing image-text paired datasets. We leverage the web-based image-caption pairs datasets (like WebLI [9] and YFCC100M [47]) to create a localized captions dataset. The dataset generation pipeline is shown in Figure 3. First, we create *text queries* using n-grams from the caption of the image: e.g. “dog",“brown dog", “brown dog playing with a disc". We specifically create n-grams where $n = \{1, 2, \cdots, 8\}$ and then filter out incomplete captions like “with a red", “dog playing with". More details about the filtering step are mentioned in the appendix. Then we use the filtered n-grams as *text queries* for pre-trained region proposal models (i.e. OWL-ViT [36]) to extract boxes and select text-box pairs based on the similarity score (> 0.1). Multiple n-grams may match for a box, and this results in several ways of describing a box in the image as shown in Col. 4 in Figure 3.

WebLI. This data collection technique on the WebLI dataset results in 32 billion image-box-caption triplets from 2 billion images without requiring new manual annotation. Our captions show a rich vocabulary that is close to common language used to describe objects in the context of an image. If we use MS-COCO’s vocabulary then all humans in the dataset would get labeled as *person*. However by building our vocabulary in a bottom-up manner we end up with captions that contain more informative words such as *baby*, *nurse*, *policeman*, *firefighter*, or *baseball player* to describe the *person* class. Please refer to the appendix for details of dataset statistics and examples.

YFCC100M. We also create a localized captions dataset using YFCC100M images. Specifically we use the same 14M images as the CLIP paper. The dataset creation method results in ~ 11M images with at least one valid box. On average each image has 20 boxes, making the size of this dataset ~ 0.2B image-box-caption triplets. The number of YFCC100M triplets is 160 times smaller than the localized captions dataset created from WebLI.

As both OWL-ViT and the CLIP subset of YFCC100M are publicly available, the resulting localized captions dataset can be generated with open-source models and public datasets. Since the WebLI dataset is not publicly available yet, YFCC100M triplet-dataset can serve as a reproducible benchmark. Concurrently large-scale grounded image-text dataset generation pipelines have also been proposed in [38] and [50].

4 Experiments

4.1 Correctness and Compliance of Generated Captions

Correctness. In this experiment, we solely evaluate the recognition capabilities of our model in a zero-shot manner. We use the region classification task [53, 61] on the MS-COCO dataset to assess how well our model recognizes objects at different scales and under occlusion. In this task, the image and the ground truth bounding box are provided as input to the model such that it produces a short description of what is contained in the bounding box. Our region classification pipeline (Figure 4)

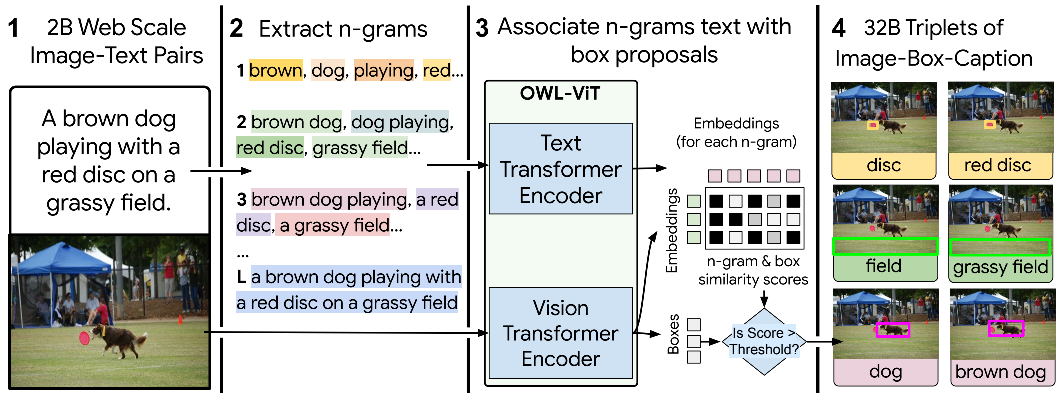


Figure 3: **Dataset Generation.** We use OWL-ViT to generate a dataset of triplets of image, bounding box and captions from a web-scale dataset of noisy image-text pairs. Increasing levels of richness in captions is captured through different length descriptions for each box.

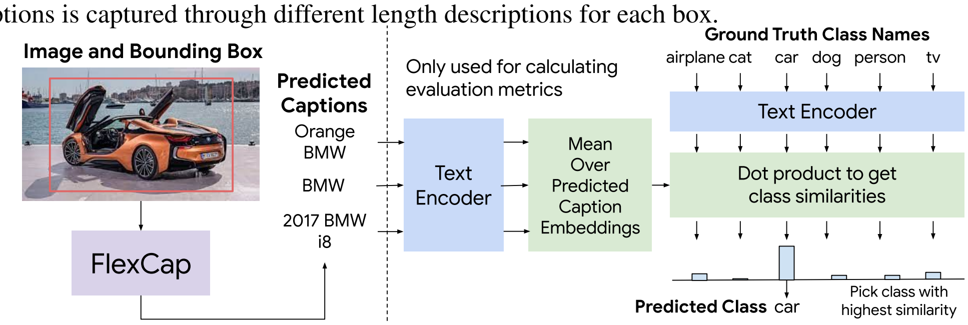


Figure 4: **Evaluating open-vocabulary outputs** from FlexCap using the CLIP [39] text encoder.

passes the input image (of size 448×448) through FlexCap which generates 20 captions each for 4 caption lengths (1, 2, 3, 4) via nucleus sampling. These captions are then mapped to object class names using CLIP's [39] text encoder. By comparing the mean of the text embeddings of predicted captions with those of ground truth class names, we obtain the classification scores. We report the results in Table 1a. We find FlexCap outperforms contrastively trained approaches used for region classification. Furthermore, we observe a significant boost (13% mAP) in performance by producing multiple captions for each box. One reason for the boost in performance may be directly comparing text embeddings to produce classification scores as compared to baselines which use dot product of image and text embeddings.

Compliance. In Figure 5, we show qualitative examples of the FlexCap model producing different length captions for the same box. Note how the model progressively adds more information about the object by incorporating context in the longer sentences (*in the jungle*), attributes (*pink flamingo kite*), and alternative nouns (*chevy*, *feline*). We also measure how well our model complies to the desired caption length. To do so, we take 1000 images from the MS-COCO dataset and use a random object in the image to produce descriptions with different lengths. We report the average length of the predicted caption, and the fraction of times the predicted caption has a length equal to the target length in Table 1b. We find FlexCap's outputs are mostly compliant with the target length.

4.2 Visual Question Answering

Visual question answering (VQA) often requires visually grounded rich semantic understanding of the content at multiple levels of granularity depending on the question. These properties make VQA a great test-bed for our method which can generate dense spatially grounded information on visual content with desired semantic complexity.

FlexCap-LLM. In Figure 1 (right), we show how we use FlexCap with an LLM to solve visual questions. First, we convert an image to a sequence of localized descriptions that describe the image in terms of the objects and regions present in the image. To do so, we need region proposals. We use OWL-ViTv2 [35] to localize important objects and regions in an image. We keep the top 128

https://doi.org/10.52202/079017-3530

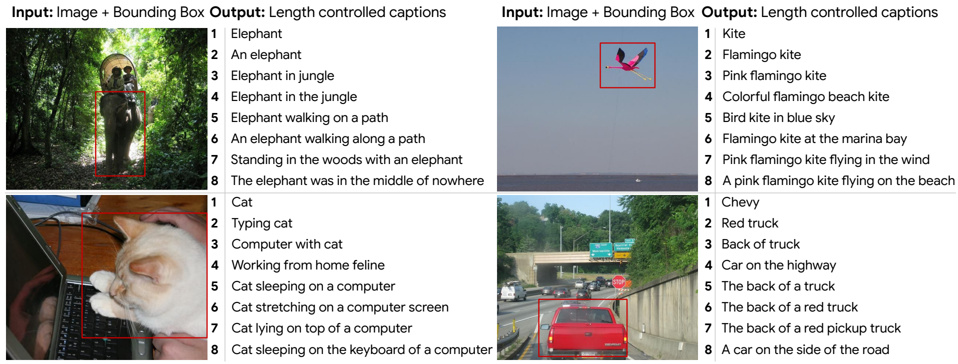


Figure 5: **Examples of length controlled captions generated by FlexCap.** Note that attributes ("pink flamingo kite") and context ("in the jungle") are generated as the length increases.

Method	mAP (↑)
CLIP [53]	29.2
CLIM [53]	62.2
RegionCLIP [61]	62.8
FlexCap (Top-1 caption)	72.0
FlexCap (Top-20 captions)	**85.0**

(a) **MS-COCO Region Classification** with ground truth bounding box.

Target Length	Mean Len of Pred.	Accuracy	Target Length	Mean Len of Pred.	Accuracy
1	1.02	0.99	5	5.06	0.94
2	2.05	0.96	6	6.01	0.97
3	3.06	0.95	7	7.02	0.95
4	4.04	0.96	8	8.02	0.95

(b) **Compliance metrics.** FlexCap produces length-compliant captions for different lengths.

Table 1: **FlexCap's outputs are accurate and length compliant.**

bounding boxes by their *objectness* scores. We then use FlexCap to describe each box in the image in the context of the entire image. In order to produce holistic descriptions, we use multiple prefixes for each region. These prefixes are a combination of length conditioning token and some initial text. We add the boxes and their descriptions to a *text preamble* as context to the LLM (see Figure 1) that defines the setup where we are using an LLM to answer questions about an image. In all the experiments, we use PALM2-S model [4] as the LLM of choice. We refer to this end-to-end system that takes an image and a question to output the answer as *FlexCap-LLM*.

To adapt the base FlexCap to have improved detection skills, output longer sentences, and identify OCR, we co-train FlexCap for 25k more steps on detection (COCO, VOC, OpenImages, LVIS), captioning (COCO Captions, Visual Genome) and OCR datasets (WebLI). For image captioning datasets, we use the bounding box that covers the whole image. We find this co-training step useful for downstream tasks using the LLM.

Hence we evaluate the effectiveness of FlexCap-LLM on several image VQA benchmarks such as OKVQA [34], VQAv2 [16], GQA [23], and VizWiz [18], and video question answering benchmarks such as MSRVTT [55] and MSVD [54]. Diverse characterictics of these datasets helps gaining better insight on FlexCap's capabilities. We report the commonly used accuracy metric for each dataset.

Image Question Answering. First we evaluate FlexCap-LLM on VQAv2 , GQA, OKVQA and VizWiz image VQA benchmarks in a zero-shot setting, meaning that our approach is not trained with the task or the corresponding dataset. The results on these benchmarks are presented in Table 2.

Standard VQA. The VQAv2 dataset is a standard for evaluating the performance of visual question-answering systems. In Table 2a, we present the results of our evaluation of FlexCap-LLM on this dataset. We find that FlexCap-LLM outperforms other zero-shot baselines, such as BLIP-2. This performance is achieved by providing object and region level information to LLMs without requiring multi-modal fine-tuning.

Compositional VQA. The GQA dataset is for evaluating the performance on complex compositional questions. As FlexCap produces information for multiple visual elements in the scene with their corresponding locations, it is quite well-suited for questions on compositional understanding of the image. On this benchmark, as shown in Table 2b, FlexCap-LLM outperforms all the recent baselines except InstructBLIP [11].

Method	Acc.(%) ↑
PalmE-562B [14]	80.0
VLMO [6]	82.8
BEIT-3 [49]	84.2
PaLI-17B [9]	84.3
FewVLM [25]	47.7
Flamingo [2]	56.3
BLIPv2 [29]	65.2
FlexCap-LLM	**65.6**

(a) **VQAv2 results** on test-dev set

Method	Acc.(%) ↑
LGCN [20]	55.8
LXMERT [45]	60.0
NSM [22]	63.0
CFR [37]	72.1
BLIPv2 [29]	44.7
ViperGPT [44]	48.1
FlexCap-LLM	48.8
InstructBLIP [11]	**49.5**

(b) **GQA results** on test-dev balanced set

Method	Acc.(%) ↑
PalmE-12B [14]	55.5
PalmE-562B [14]	66.1
PaLI-3B [9]	52.4
PaLI-17B [9]	64.5
BLIPv2 [29]	45.9
Flamingo [3]	50.6
ViperGPT [44]	51.9
FlexCap-LLM	**52.1**

(c) **OKVQA results** on val set

Method	Acc.(%) ↑
Flamingo 32 Shot [14]	49.8
Flamingo FT [14]	65.7
PaLI-3B [9]	67.5
PaLI-17B [9]	74.4
Flamingo [3]	31.6
Emu [43]	34.2
InstructBLIP [11]	34.5
FlexCap-LLM	**41.8**

(d) **VizWiz results** on test-dev set

Table 2: **Zero-shot image question answering results**. FlexCap-LLM is compared against recent baselines. Grayed out methods are trained on question answering datasets.

	Acc.(%) ↑	
Method	MSRVTT	MSVD
Emu [43]	8.3	18.8
Flamingo [3]	17.4	35.6
FlexCap-LLM	**25.0**	**39.5**

Table 3: **Zero-shot video question answering results** reported on MSRVTT-QA and MSVD-QA on the test set. FlexCap-LLM is better than other zero-shot baselines for video VQA benchmarks.

Method	mAP ↑
FCLN [26]	27.11
CAG-Net [58]	36.29
FlexCap	**46.9**

(a) Captioning GT Boxes

Method	mAP↑
FCLN [26]	5.39
JIVC [56]	9.31
COCG [31]	9.82
CAG-Net [58]	10.51
TDC+ROCSU [41]	11.49
GRiT [52]	15.52
FlexCap + GRiT Boxes	**16.2**

(b) Dense Captioning

Table 4: **Captioning boxes in Visual Genome dataset.** FlexCap exceeds performance of other methods. All methods have been fine-tuned on Visual Genome captions.

VQA with External Knowledge. OKVQA dataset is particularly designed for evaluating the ability to answer questions about images that require external knowledge which is not readily available on the image. Hence it requires multiple levels of understanding of the content, and reasoning with that information, which is well-suited for applying FlexCap. In Table 2c we show our performance on OKVQA is superior to strong baselines such as Flamingo and ViperGPT which highlights the effectiveness of the mix of generic and specific descriptions generated by FlexCap. Unlike other baselines which use the question, FlexCap generates captions without having access to the question.

VQA with atypical images. We also evaluate on VizWiz, which contains visual questions asked by people who are visually impaired. Unlike web content, in these images the objects and the scene are not always well-centered, hence this dataset contains many out-of-distribution samples compared to typical web-crawled datasets. We report the results of this experiment in Table 2d. Nevertheless, our approach significantly outperforms Flamingo [2] and InstructBLIP [11] in the zero-shot setting.

Video Question Answering. We also evaluate FlexCap-LLM on zero-shot video question answering datasets MSRVTT-QA and MSVD-QA [54]. The results on these benchmarks are presented in Table 3. For processing the video, we sample 8 frames uniformly from the video. We pass each of these frames through FlexCap to produce captions of objects and regions. We then combine all the object captions from the different frames into one prompt for the LLM. We observe FlexCap-LLM

https://doi.org/10.52202/079017-3530

significantly exceeds the performance of the Flamingo model in the zero-shot setting. These results highlight the zero-shot effectiveness of our method, which can solve tasks in the video domain even though both FlexCap and the LLM were not trained for those tasks.

4.3 Dense Captioning

Dataset and Evaluation Metrics. The dense captioning task is defined as producing both the regions and the corresponding descriptions for each region. For this experiment, we use the Visual Genome [28] dataset. In this dataset, each image is annotated with multiple bounding boxes and each box has a corresponding caption. We use the train-test splits and evaluation metric as proposed in [26]. The paper proposes to use a mean of Average Precisions (mAP) over pairwise thresholds of both IOU thresholds (0.3, 0.4, 0.5, 0.6, 0.7) and Meteor score thresholds (0.0, 0.05, 0.1, 0.15, 0.2, 0.25). We use the same preprocessing of text and boxes as mentioned in [52].

Fine-tuning FlexCap. We fine-tune the pretrained FlexCap model on the Visual Genome train split for 60k steps with a lower learning rate of $1e-6$ at a resolution of 448×448.

Captioning GT boxes. Following the evaluation procedure from [26], we evaluate captioning of the ground-truth boxes in Visual Genome. Since this setting removes the localization task, we have a cleaner evaluation of only the region captioning problem. The results of this experiment are provided in Table 4a in which we show that FlexCap achieves better performance compared to other approaches evaluated in this setting.

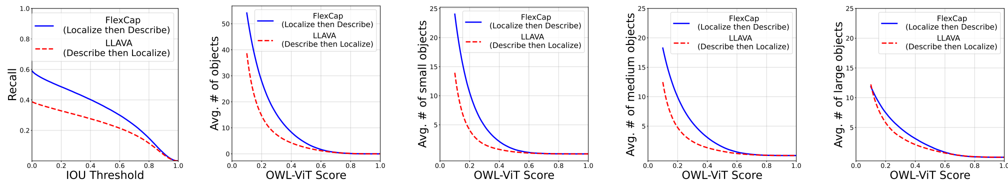
Captioning GRIT boxes. In this experiment, we want to compare against other approaches that perform both localization and captioning. We are measuring how well FlexCap performs when provided with box proposals. Table 4b shows FlexCap obtains better performance compared to other approaches evaluated in this setting. Since in our work we do not propose any localization module, we use GRIT's [52] region proposals as the input boxes for our model. This also allows us to directly compare our captioning capabilities against GRIT. We find that our approach outperforms GRIT at this task even though we test at a lower resolution of 448×448.

4.4 Open-Ended Object Detection

Open-ended object detection, that is identifying all the objects in the image with rich descriptions, can be achieved in two ways: (a) localize-then-describe or (b) describe-then-localize. With FlexCap, we employ a *localize-then-describe* approach where boxes are generated with a box proposal mechanism (i.e. OWL-ViT) and then described by FlexCap. We compare these results with a *describe-then-localize* approach where we first ask a state-of-the-art (SOTA) VLM (i.e. LLAVA [33]) for a comprehensive description of the image with all objects and then localize these descriptions using a SOTA open-vocabulary object detection method (i.e. OWL-ViT [36]). The *Recall* of all the objects in the image is a major criteria in open-ended object detection. As shown in Figure 6(a), we demonstrate that *localize-then-describe* approach powered with FlexCap performs significantly better compared to *describe-then-localize* approach using existing works. In Figure 6 (b-e), we observe that our *localize-then-describe* approach retrieves more parts of the image, particularly the small and medium size objects, compared to the *describe-then-localize* approach. This is mainly because most VLMs are trained to describe the salient objects in the image rather than exhaustively list all the objects present in the scene.

4.5 Discussion

Prefixing. Training on a large dataset also enables the extraction of different kinds of information from an image using *prefixes*. We can use this property to our advantage to extract object properties such as color and material. We show examples of this in Figure 7. Say we want to extract the color and material of objects, we design two prefixes: 1) "*LENGTH-4 The color is* ____" and 2) "*LENGTH-5 This is made of* ____". The pre-fixed length tokens guide the model to produce outputs that are just one more word. Note how the model is aware of the input prefix and completes based on the object in the bounding box without confusing it with other surrounding objects. We show more examples of using the prefix to define in Fig. 8. FlexCap can extract the following attributes with the corresponding prefix in the brackets: the actions of humans (*"The person is"*), the uses of objects (*"This is used for"*), OCR (*"The sign says"*), names of books (*"The book is called"*), authors of books



(a) Recall v/s IOU (b) Number of Objects Found (c) Number of Small Objects Found (d) Number of Med. Objects Found (e) Number of Large Objects Found

Figure 6: **Open-Ended Object Detection** on the Visual Genome dataset. We find that the localize-then-describe approach, which involves describing every detected box with FlexCap, achieves higher recall (see (a)) and produces more bounding boxes with good matching scores (see (b)) compared to describe-then-localize approach with SOTA VLMs and open-vocabulary detection. The difference between the two approaches is most stark for small and medium sized objects (see (c-e)).

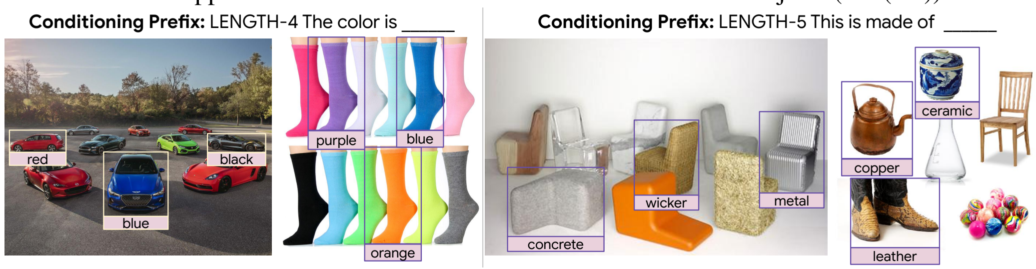


Figure 7: **Extracting properties by conditioning FlexCap with prefixes.** Examples of FlexCap extracting properties of objects of different categories by using relevant prefixes. Note how we are able to retrieve a one-word answer from the model by controlling the length of the caption.

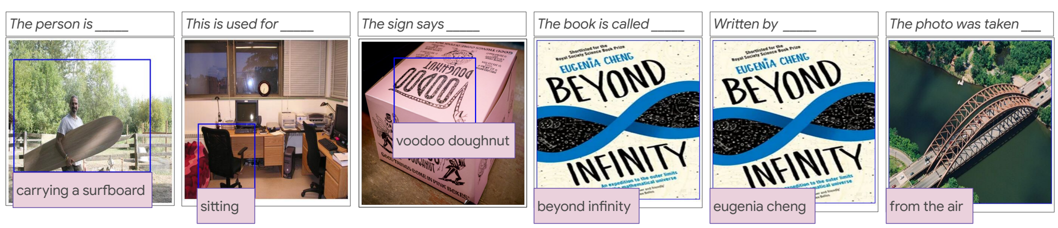


Figure 8: **Generating conditional captions using prefixes.** Conditional captions allow us to extract desired information from the input bounding box.

(*"Written by"*), and the locations where the photos were taken (*"The photo was taken in"*). Note the same model can extract different information from the same bounding box: the name of the book title and the author name based on the prefixing (col 4-5).

Limitations. We use OWL-ViT and alt-text to generate the training dataset for the model rather than relying on human annotations. While this approach allows us to scale, there exist biases in the model and alt-text which will reflect in the outputs of FlexCap. Another limitation is that the proposed FlexCapLLM system is not end-to-end trainable but is a composition of a VLM and LLM. This can be alleviated by training a VLM with the localized captioning dataset.

Broader Impact. Localized and controllable image captioning enabled by our model may benefit applications like accessibility tools for visual impairments and intuitive human-computer interaction. However, biases in training data raise concerns around potential misrepresentation or exclusion of certain demographics. Dual-use risks also exist if this model is employed for unethical surveillance.

5 Related Work

Visual question answering (VQA), a task designed to assess if a computer can answer questions about an image, often requires grounding visual concepts and reasoning. Although initially introduced for supervised evaluation of the task [5], most recently, VQA also become one of the most powerful benchmarks for evaluating task and dataset independent visual dialog. Several existing models such

https://doi.org/10.52202/079017-3530

as ViperGPT [44], Flamingo [2], BLIP [29, 30], PaLI [10] show convincing zero-shot performance on the VQA benchmarks that rivals the supervised approaches. Unlike most previous zero-shot approaches, which tightly couple vision and language components in a single model, FlexCap generates a high-level human interpretable representation of an image and demonstrates that, through straight-forward application of LLMs, we can achieve comparable performance with state-of-the-art results across VQA benchmarks. Unlike others, ViperGPT [44], also decouples vision and language components and reinterprets visual questions with LLM generated programs, and executes them using existing visual perception tools. Whereas, in our case we use only one powerful vision tool, i.e. FlexCap, to generate all the necessary information and leave the reasoning to an LLM. In that sense, FlexCap is quite complementary to ViperGPT as it can be one of the powerful tools that can improve the controllable visual understanding of the image for ViperGPT.

Open vocabulary object detection models like OWL-ViT [36] and ViLD [17] enable the user to query any given text on the image and obtain matched bounding boxes for those queries. In these models the text is often encoded by a text encoder like CLIP [39] and T5 [40]. The text embeddings are compared with the category-agnostic box proposals coming from the visual backbone. In this work, we use OWL-ViT's text and vision encoders to associate bounding boxes with text-queries to produce our training data. By training a localized captioning model, we remove the manual step of providing per-dataset or per-image text queries to use OWL-ViT. RegionCLIP [61] obtained good performance on open-vocabulary object detection by utilizing region-level vision-language contrastive learning on large scale data. We differ from this work as we generate the description for each bounding box instead of associating text queries (defined manually) with bounding boxes.

Dense captioning involves localizing salient regions of the image and describing them with natural language sentences, introduced in [26]. In practice, the existing work often produces longer and more informative descriptions of objects or their compositions using visual attributes of objects [27, 58] or contextual and global image cues [31, 56]. However, the richness of descriptions in this line of work are often limited to existing image captioning datasets [28, 32]. By utilizing a large scale dataset of billions of noisy image-text pairs collected from the web (similar to [9, 24]), we aim to generate more diverse sentences with a focus on describing the visual content in controllable detail using a richer visual descriptive space learned from the web.

Length-controlled image captioning has been explored in ZeroCap [46] and LIC [13]. ZeroCap [46] implements length control as a post-processing step by changing the probability of sampling the end-of-sentence token. Hence the model is not naturally trained with word length conditioning in mind and cannot guarantee fine-grained length control at the level of number of words. On the other hand, LIC [13] generates length-controllable captions by conditioning the model with learned tokens that represent different length intervals. However there are considerable differences compared to FlexCap. First, our approach allows for controllability at the level of image regions, while LIC only provides full image captions. This is a significant difference, as it allows us to generate concise or detailed captions for all the objects in the image. Second, our approach has a more precise level of caption-length control. LIC uses a coarse subjective level of control with four or five levels of length (e.g. short, medium, long, and longer), while our approach allows for an exact number of words to be specified. [48] also propose an approach to produce variable length descriptions for objects localized interactively. They use a pre-trained image captioner to produce descriptions of objects and ChatGPT in post-hoc to output length-conditioned captions. While in our FlexCap model, the length tokens modulate the output produced by the captioner.

6 Conclusion

In this work, we introduce FlexCap, a flexible captioning model that can describe localized regions in an image with controllably rich captions. To train FlexCap, we generate a large-scale image-box-caption dataset that is rich in diversity of visual descriptions and their length. We achieve this by utilizing existing web-scale noisy image-text pairs and open-vocabulary object detection models. We show how localized rich descriptions provided by FlexCap can help us connect images and videos to LLMs and achieve strong performance on visual question answering and dense captioning tasks. We also show that our FlexCap model benefits from contrastive pretraining, localized captions dataset size scaling, model size scaling, and is compliant to length conditioning. We also demonstrate the effectiveness of FlexCap-enabled *localize-then-describe* approach over the *describe-then-localize* approaches.

https://doi.org/10.52202/079017-3530

Acknowledgements

The authors would like to thank Matthias Minderer, Lisa Anne Hendricks, Andy Zeng, Matthias Bauer, Anastasija Ilić, Alexey Gritsenko, Vincent Vanhoucke, Jonathan Hoech, Alex Irpan, and Shefali Umrania for their feedback and helpful discussions.

References

[1] I. Alabdulmohsin, X. Zhai, A. Kolesnikov, and L. Beyer. Getting vit in shape: Scaling laws for compute-optimal model design. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[2] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022.

[3] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan. Flamingo: a visual language model for Few-Shot learning. *ArXiv*, abs/2204.14198, 2022.

[4] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. Palm 2 technical report. *arXiv e-prints*, pages arXiv–2305, 2023.

[5] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[6] H. Bao, W. Wang, L. Dong, Q. Liu, O. K. Mohammed, K. Aggarwal, S. Som, S. Piao, and F. Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35:32897–32912, 2022.

[7] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

[10] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer, A. Kolesnikov, J. Puigcerver, N. Ding, K. Rong, H. Akbari, G. Mishra, L. Xue, A. V. Thapliyal, J. Bradbury, W. Kuo, M. Seyedhosseini, C. Jia, B. K. Ayan, C. R. Ruiz, A. P. Steiner, A. Angelova, X. Zhai, N. Houlsby, and R. Soricut. PaLI: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*, 2023.

[11] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi. Instructblip: towards general-purpose vision-language models with instruction tuning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

[12] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual dialog. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 326–335, 2017.

[13] C. Deng, N. Ding, M. Tan, and Q. Wu. Length-controllable image captioning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 712–729. Springer, 2020.

[14] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

 https://doi.org/10.52202/079017-3530

[15] R. A. Google, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, E. Chu, J. H. Clark, L. E. Shafey, Y. Huang, K. Meier-Hellstern, G. Mishra, E. Moreira, M. Omernick, K. Robinson, S. Ruder, Y. Tay, K. Xiao, Y. Xu, Y. Zhang, G. H. Abrego, J. Ahn, J. Austin, P. Barham, J. Botha, J. Bradbury, S. Brahma, K. Brooks, M. Catasta, Y. Cheng, C. Cherry, C. A. Choquette-Choo, A. Chowdhery, C. Crepy, S. Dave, M. Dehghani, S. Dev, J. Devlin, M. Díaz, N. Du, E. Dyer, V. Feinberg, F. Feng, V. Fienber, M. Freitag, X. Garcia, S. Gehrmann, L. Gonzalez, G. Gur-Ari, S. Hand, H. Hashemi, L. Hou, J. Howland, A. Hu, J. Hui, J. Hurwitz, M. Isard, A. Ittycheriah, M. Jagielski, W. Jia, K. Kenealy, M. Krikun, S. Kudugunta, C. Lan, K. Lee, B. Lee, E. Li, M. Li, W. Li, Y. Li, J. Li, H. Lim, H. Lin, Z. Liu, F. Liu, M. Maggioni, A. Mahendru, J. Maynez, V. Misra, M. Moussalem, Z. Nado, J. Nham, E. Ni, A. Nystrom, A. Parrish, M. Pellat, M. Polacek, A. Polozov, R. Pope, S. Qiao, E. Reif, B. Richter, P. Riley, A. C. Ros, A. Roy, B. Saeta, R. Samuel, R. Shelby, A. Slone, D. Smilkov, D. R. So, D. Sohn, S. Tokumine, D. Valter, V. Vasudevan, K. Vodrahalli, X. Wang, P. Wang, Z. Wang, T. Wang, J. Wieting, Y. Wu, K. Xu, Y. Xu, L. Xue, P. Yin, J. Yu, Q. Zhang, S. Zheng, C. Zheng, W. Zhou, D. Zhou, S. Petrov, and Y. Wu. Palm 2 technical report, 2023.

[16] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *International Journal of Computer Vision*, 127:398 – 414, 2016.

[17] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

[18] D. Gurari, Q. Li, A. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3608–3617, 2018.

[19] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[20] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko. Language-conditioned graph networks for relational reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10294–10303, 2019.

[21] Y. Hu, H. Hua, Z. Yang, W. Shi, N. A. Smith, and J. Luo. Promptcap: Prompt-guided image captioning for vqa with gpt-3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2963–2975, 2023.

[22] D. Hudson and C. D. Manning. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32, 2019.

[23] D. A. Hudson and C. D. Manning. Gqa: a new dataset for compositional question answering over real-world images. *ArXiv*, abs/1902.09506, 2019.

[24] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[25] W. Jin, Y. Cheng, Y. Shen, W. Chen, and X. Ren. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. *arXiv preprint arXiv:2110.08484*, 2021.

[26] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016.

[27] D.-J. Kim, J. Choi, T.-H. Oh, and I. S. Kweon. Dense relational captioning: Triple-stream networks for relationship-based captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6271–6280, 2019.

[28] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[29] J. Li, D. Li, S. Savarese, and S. Hoi. BLIP-2: Bootstrapping Language-Image pre-training with frozen image encoders and large language models. Jan. 2023.

[30] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[31] X. Li, S. Jiang, and J. Han. Learning object context for dense captioning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8650–8657, 2019.

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[33] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[34] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3190–3199, 2019.

[35] M. Minderer, A. Gritsenko, and N. Houlsby. Scaling open-vocabulary object detection. *arXiv preprint arXiv:2306.09683*, 2023.

[36] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby. Simple open-vocabulary object detection. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, page 728–755, Berlin, Heidelberg, 2022. Springer-Verlag.

[37] B. X. Nguyen, T. Do, H. Tran, E. Tjiputra, Q. D. Tran, and A. Nguyen. Coarse-to-fine reasoning for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4558–4566, 2022.

[38] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, Q. Ye, and F. Wei. Grounding multimodal large language models to the world. In *The Twelfth International Conference on Learning Representations*, 2024.

[39] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[40] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[41] Z. Shao, J. Han, D. Marnerides, and K. Debattista. Region-object relation-aware dense captioning via transformer. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[42] Z. Shao, Z. Yu, M. Wang, and J. Yu. Prompting large language models with answer heuristics for knowledge-based visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14974–14983, 2023.

[43] Q. Sun, Q. Yu, Y. Cui, F. Zhang, X. Zhang, Y. Wang, H. Gao, J. Liu, T. Huang, and X. Wang. Emu: Generative pretraining in multimodality. In *The Twelfth International Conference on Learning Representations*, 2024.

[44] D. Sur'is, S. Menon, and C. Vondrick. ViperGPT: Visual inference via python execution for reasoning. *ArXiv*, abs/2303.08128, 2023.

[45] H. Tan and M. Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[46] Y. Tewel, Y. Shalev, I. Schwartz, and L. Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928, 2022.

[47] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: the new data in multimedia research. *Commun. ACM*, 59(2):64–73, jan 2016.

[48] T. Wang, J. Zhang, J. Fei, Y. Ge, H. Zheng, Y. Tang, Z. Li, M. Gao, S. Zhao, Y. Shan, et al. Caption anything: Interactive image description with diverse multimodal controls. *arXiv preprint arXiv:2305.02677*, 2023.

[49] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

https://doi.org/10.52202/079017-3530

[50] W. Wang, M. Shi, Q. Li, W. Wang, Z. Huang, L. Xing, Z. Chen, H. Li, X. Zhu, Z. Cao, et al. The all-seeing project: Towards panoptic visual recognition and understanding of the open world. *arXiv preprint arXiv:2308.01907*, 2023.

[51] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

[52] J. Wu, J. Wang, Z. Yang, Z. Gan, Z. Liu, J. Yuan, and L. Wang. Grit: A generative region-to-text transformer for object understanding. *arXiv preprint arXiv:2212.00280*, 2022.

[53] S. Wu, W. Zhang, L. Xu, S. Jin, W. Liu, and C. C. Loy. Clim: Contrastive language-image mosaic for region representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6117–6125, 2024.

[54] D. Xu, Z. Zhao, J. Xiao, F. Wu, H. Zhang, X. He, and Y. Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017.

[55] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[56] L. Yang, K. Tang, J. Yang, and L.-J. Li. Dense captioning with joint inference and visual context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2193–2202, 2017.

[57] Z. Yang, Z. Gan, J. Wang, X. Hu, Y. Lu, Z. Liu, and L. Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3081–3089, 2022.

[58] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, and J. Shao. Context and attribute grounded dense captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6241–6250, 2019.

[59] H. You, H. Zhang, Z. Gan, X. Du, B. Zhang, Z. Wang, L. Cao, S.-F. Chang, and Y. Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023.

[60] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023.

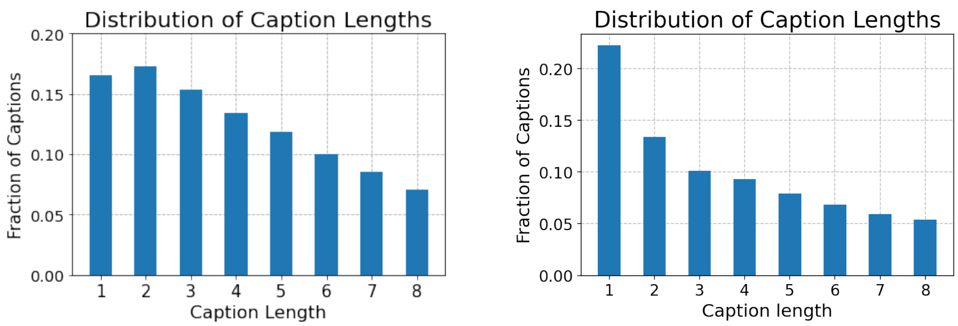
[61] Y. Zhong, J. Yang, P. Zhang, C. Li, N. Codella, L. H. Li, L. Zhou, X. Dai, L. Yuan, Y. Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, 2022.

Appendix

A Localized Captions Dataset Details

WebLI. Our dataset creation process on the WebLI dataset results in $\sim$32 billion Image-Box-Caption triplets. In Figure 9a, we show the distribution of caption lengths in the generated dataset using WebLI. We observe that the distribution is not uniform. This is due to the fact that there are more n-grams of length 1 to sample than length 8. The average number of unique boxes in an image is 4.19, and the average number of captions per box is 4.04.

YFCC100M. Our dataset creation process on the CLIP subset of the YFCC100M dataset results in $\sim$0.2 billion Image-Box-Caption triplets. In Figure 9b, we show the distribution of caption lengths in the generated dataset using YFCC100M. The average number of unique boxes in an image is 6.3, and the average number of captions per box is 3.3.



(a) WebLI Dataset

(b) YFCC100M Dataset

Figure 9: **Distribution of caption lengths in the Localized Captions Dataset.** We show the distribution of caption length for the region-captions obtained using paired image-texts from two datasets: WebLI [9] and YFCC100M [47]

.

N-gram Filtering. Before matching n-grams with boxes, we filter out n-grams that do not form informative or grammatically correct captions for boxes. This is done with three steps: 1) Removing any captions composed only of uninformative words (image, jpg, background, wallpaper, hd wallpaper etc.); 2) Removing n-grams that begin with words with which sentences usually do not start (of, on, in etc.); 3) Removing n-grams that finish with words with which sentences usually do not end (a, the, to, on etc.). This step is essential to reduce noise present in the large-scale image-text pair dataset obtained from the web. It is also important for the captioning model to produce grammatically correct informative sentences.

Dataset Samples. We show some samples from the dataset in Figure 10. The alt-text from which the box captions are generated is provided as the title of the image. Note the alt-text gets clipped due to display-length limits which is why the detected boxes might have captions not visible in the displayed alt-text directly. We next discuss how captions of varying lengths are matched with different objects in an image.

B Ablations and Analysis

Large-scale Contrastive Pre-Training. In this section, we study the impact of using a contrastively pre-trained vision encoder or training from scratch using only box-caption objective. For this experiment, we use ViT-B/16 as our backbone and train on the WebLI Region Captioning dataset. We report the results on ground truth (GT) box captioning and dense captioning on GRiT proposals on the Visual Genome dataset in Table 5a. In Row 1, we observe that a length-conditioned region captioning objective can be used for training both the backbone and text decoder from scratch and achieve competitive performance. Second, we observe using a large-scale contrastively pre-trained (CPT) model results in better performance. Note that these CPT weights have already been open-sourced [60].

 https://doi.org/10.52202/079017-3530

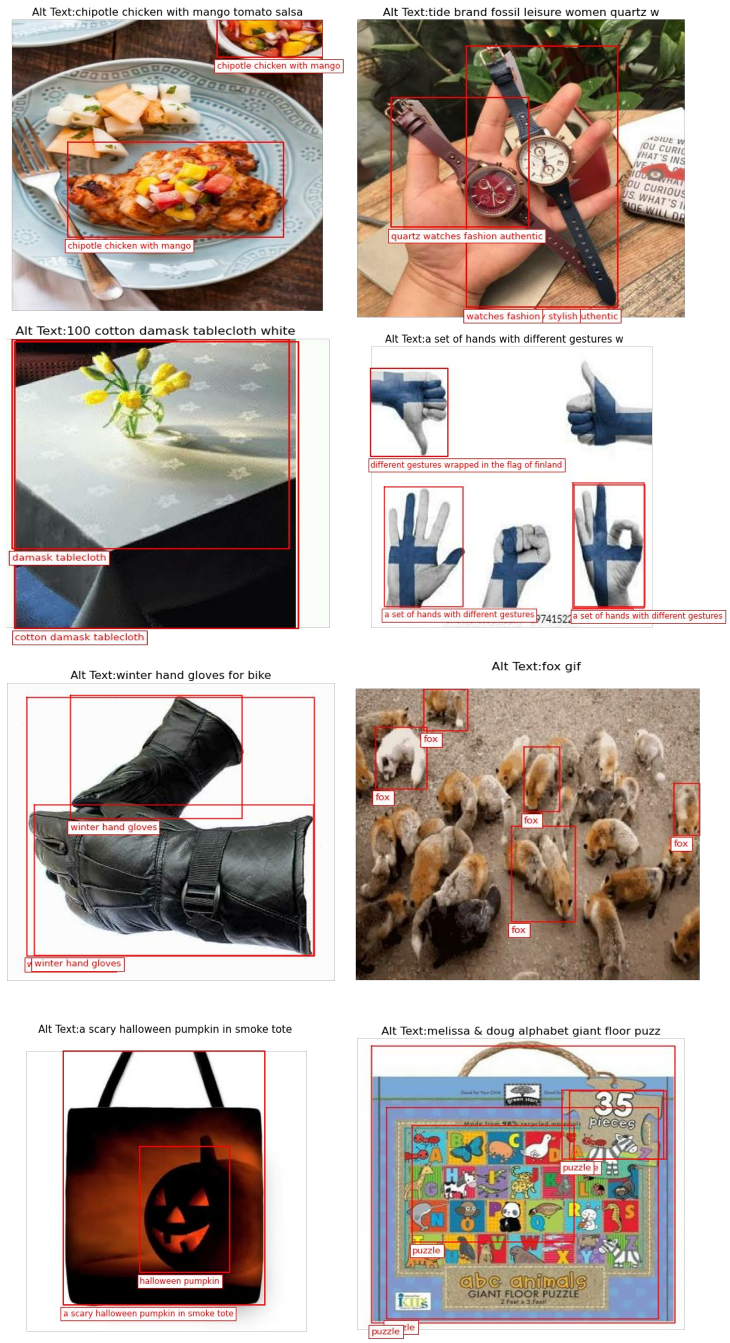


Figure 10: **Samples from the Localized Captions dataset with images from the WebLI dataset [9]**. We only visualize a maximum of 5 boxes for each image to avoid clutter.

	VG mAP ↑	
CPT	GT	GRiT
	43.6	15.6
✓	**45.1**	**15.8**

(a) **Contrastive Pre-training**

		VG mAP ↑	
Dataset	Size	GT	GRiT
YFCC100M	0.2B	38.5	14.2
WebLI	32B	**45.1**	**15.8**

(b) **Data Scaling**

		VG mAP ↑	
Backbone	Params	GT	GRiT
ViT-B/16	86M	45.1	15.8
SO-ViT/14	428M	**46.9**	**16.2**

(c) **Model Scaling**

Table 5: **Ablations.** We vary different aspects of model pre-training, dataset size, and model size and measure its effect on region captioning task on the Visual Genome dataset. In (a) and (b) the visual backbone is ViT-B/16 and in (b) and (c) all models start from contrastively pre-trained weights.

Figure 11: **Objects missed by LLAVA but found by FlexCap.** We show some typical examples of objects (5 per image to avoid clutter) that FlexCap can find with OWL-ViT but LLAVA does not.

Data Scaling. In this experiment we measure how scaling the localized captions dataset affects performance. We first take ViT-B/16 encoder contrastively pre-trained with a large image-text dataset (WebLI). We now train it on two different region captioning datasets of different scales. We show the results of this experiment in Table 5b. We find that even after contrastive pre-training on a large scale, a large region-caption dataset can result in significant performance boost (∼5% mAP for GT box captioning). This shows the importance of designing a scalable dataset creation method such as the one introduced in Section 4.3. We also note that the dataset and pre-trained weights required to reproduce the model with YFCC100M are available publicly.

Model Scaling. Next, we measure the impact of changing the model size. We train two models: ViT-B/16 (85M) v/s SO-ViT/14 (428M). Just like the data scaling experiments, both of these have been pre-trained on the WebLI dataset in a contrastive manner. We report the results of this experiment in Table 5c. We find that the larger model results in better performance.

C Open-Ended Object Detection

In this experiment, we compare the effectiveness of a *describe-then-localize* method (specifically LLAVA 1.5 [33] 7B + OWL-ViTv2 [35]) with a *localize-then-describe* method (specially FlexCap + OWL-ViTv2) for the task of open-ended object detection. For the *describe-then-localize* method, we generate a list of objects using the following prompt:`Describe object names and regions in this image`. We extract the nouns and use them as text queries for OWL-ViTv2. For the *localize-then-describe* approach, we take the top 128 objects ranked by objectness score by OWL-ViTv2 and describe them with different length prompts. In Fig. 5 in the main paper, we compare recall of the ground-truth regions annotated in the Visual Genome dataset for both these approaches. We find that while *describe-then-localize* with LLAVA can be an effective approach for finding large objects, *localize-then-describe* with FlexCap is significantly better at medium and small sized objects and marginally better for large-sized objects. We show some typical examples of missed detections in Fig. 11.

https://doi.org/10.52202/079017-3530

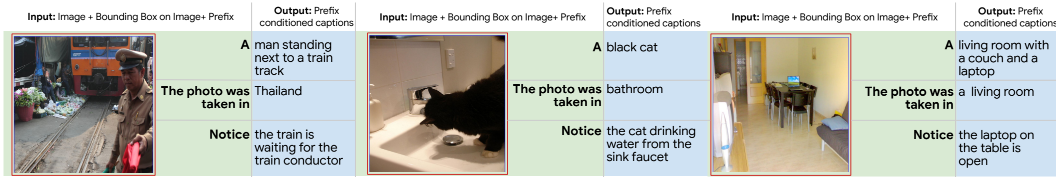


Figure 12: **Diverse captioning with Prefixes.** FlexCap can be used to perform conditional captioning of images. We show three prefixes: *"a", "the photo was taken in ", "notice"* resulting in diverse captions for the same image. Note the input red bounding box is around the full image.

	Maximum Length of Captions			
Dataset	1	2	4	8
VQAv2 (*val* split)	57.4	59.9	66.4	67.8
GQA (*testdev-balanced* split)	43.6	45.3	47.7	48.8

Table 6: **Effect of caption lengths on FlexCap-LLM**

D Different prefixes with same image

In Fig. 12, we highlight how different prefixes can be used to generate diverse captions. Note how in column 1 the model correctly identifies the country even, possibly using the logo on the train. We find the prefix *"Notice"* leads to captions highlighting noteworthy aspects in an image which the un-prefixed caption does no.

E Visual Dialog

FlexCap-LLM can be used for the task of visual dialog [12]. We first caption all the objects in the image using FlexCap and OWL-ViTv2. We retain the top 128 boxes according to the objectness score from OWL-ViTv2 and describe each box using FlexCap. See Fig. 1 and Fig. 4 in the main paper for more details. Once the image has been represented as the list of objects in text, we can interact with a LLM by providing the conversation turns as additional context for each query to the LLM. We show some examples of conversations with the FlexCap-LLM system in Figure 13. Note how the model is able to read text in the image in the leftmost figure, recognize material in the middle figure, and localize objects of interest in the rightmost figure. As we compute the object captions only once at the beginning of the conversation, there is no additional overhead of querying a large VLM for each additional turn in the conversation. We show more examples of this on the webpage in the FlexCap-LLM section of `index.html`.

F Effect of Length on FlexCapLLM

In this experiment, we study the effect of generating captions of different lengths on downstream visual question answering tasks. We evaluate this by using captions of increasing lengths on the val split of the VQAv2 dataset. We report the results of this experiment in Table 6. We find that increasing the length of generated captions results in better VQAv2 and GQA performance.

G FlexCapLLM Details

In Fig. 14 we show the whole pipeline how FlexCap interfaces with LLMs. We convert an image into a list of localized texts. These texts are passed onto the LLM with a preamble indicating the task and the VQA task's question. We use the following prompts for the LLM in the question-answering experiments.

VQAv2, OK-VQA and GQA.

Preamble:

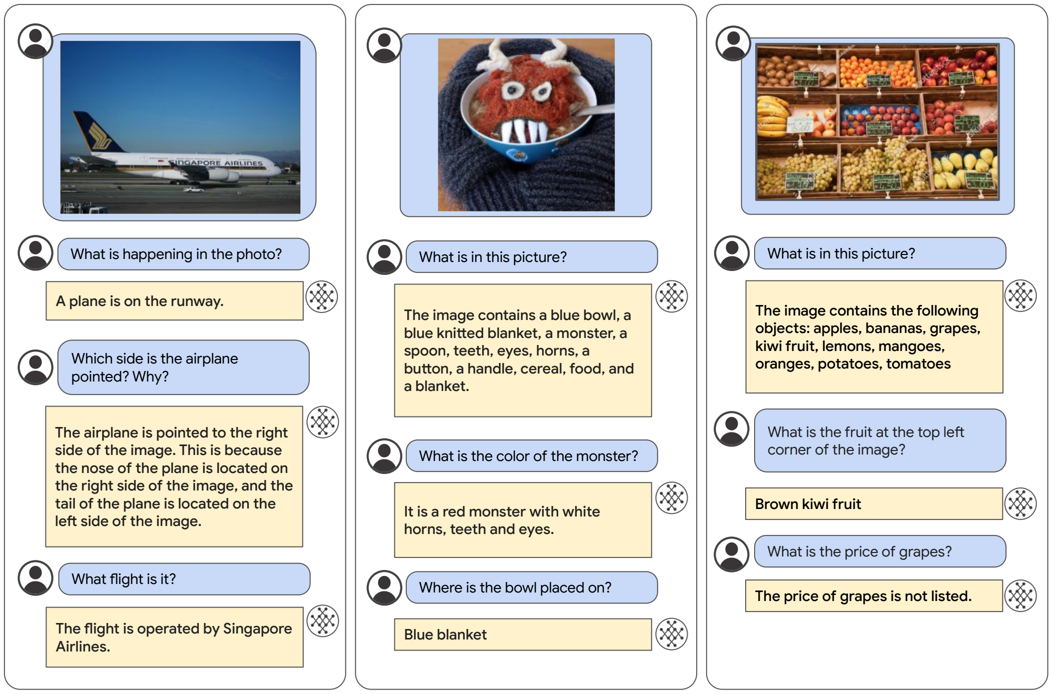


Figure 13: **FlexCap-LLM for Visual Dialog**

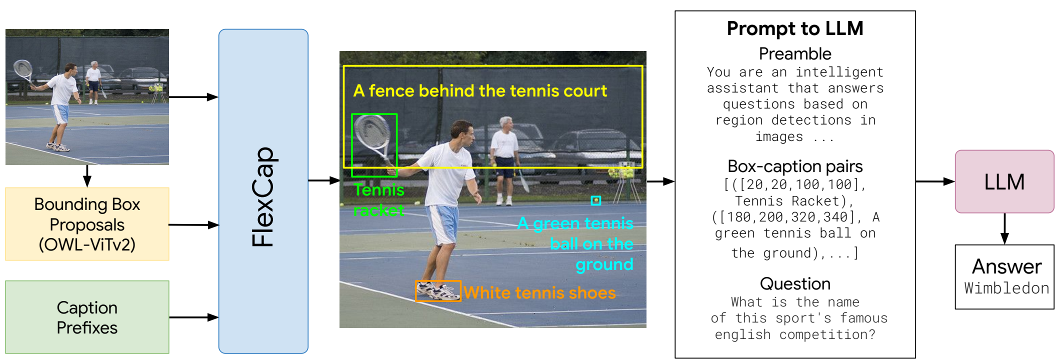


Figure 14: **FlexCap for VQA with bounding box proposals and an LLM**. FlexCap generates captions for different regions in a given image. To answer any open-ended questions, we prompt an LLM [15] with FlexCap's detections (box-caption pairs).

```
preamble = "You are a helpful assistant answering questions about images to people. You can look at the
list of object detections in the image and answer questions. The image content may not be sufficient to
answer the questions, and you may need to rely on external knowledge resources or commonsense. In an image,
 many objects were detected. They are listed in the following format: [object descriptions] [cx, cy, w, h],
 where cx is x coordinate of the center, cy is the y coordinate of the center, w is the width and h is the
 height of the bounding box of that object in the image."
```

Image Size:

```
image_size_prompt = f"The height of the image is {image_height} and width of the image is {image_width}".
```

Image description:

```
image_description = f"Full images descriptions for this image are: {image_captions}".
```

Object representation:

 https://doi.org/10.52202/079017-3530

```
objects_description = "The list of objects is as follows: "
for captions, (cx, cy, w, h) in zip(object_captions, object_boxes):
  objects_description += f"{captions} [{cx}, {cy}, {w}, {h}],"
```

Question prompt:

```
question_prompt = f"Q: {question} Answer in one word. A:"
```

Full prompt:

```
full_prompt = (preamble + image_size_prompt + image_description + objects_description + question_prompt)
```

VizWiz

Preamble:

```
preamble = "You are a helpful assistant answering questions about images to people. You can look at the
list of object detections in the image and answer questions. The image content may not be sufficient to
answer the questions, and you may need to rely on external knowledge resources or commonsense. In an image,
 many objects were detected. They are listed in the following format: [object descriptions] [cx, cy, w, h]
 [score], where cx is x coordinate of the center, cy is the y coordinate of the center, w is the width, h
is the height and score is the confidence score for the object detection. Low score means the detection is
 likely inaccurate, and this often makes the question unanswerable. You can answer questions as ’
unanswerable’."
```

Image Size:

```
image_size_prompt = f"The height of the image is {image_height} and width of the image is {image_width}".
```

Image description:

```
image_description = f"Full images descriptions for this image are: {image_captions}".
```

Object representation:

```
objects_description = "The list of objects is as follows: "
for captions, (cx, cy, w, h) in zip(object_captions, object_boxes):
  objects_description += f"{captions} [{cx}, {cy}, {w}, {h}],"
```

Question prompt:

```
question_prompt = f"Q: {question} Answer in one word. A:"
```

Full prompt:

```
full_prompt = (preamble + image_size_prompt + image_description + objects_description + question_prompt)
```

MSRVTT and MSVD

Preamble:

```
preamble = "You are a helpful assistant answering questions about videos to people. You can look at the
list of object detections in each frame and answer questions."
```

Object representation:

```
objects_description = "In a video, many objects were detected in each frame."

for frame_idx in frame_idxes:
  objects_description = f"In frame {frame_idx}, following objects were detected"
  for captions in object_captions_in_frame_idx:
    objects_description += f"{captions},"
```

Question prompt:

```
question_prompt = f"Q: {question} Answer in one word. A:"
```

Full prompt:

```
full_prompt = (preamble + objects_description + question_prompt)
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim that we can train a vision language model conditioned on bounding box and desired caption length. We show in the experiments that we can train such a model. Furthermore, we claimed that using this model we can describe an image densely using just words. We show that connecting these descriptions to an LLM leads to better dense captioning performance and zero-shot visual question answering performance.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the Discussion section we have included limitations of the current work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

https://doi.org/10.52202/079017-3530

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification:No theoretical claims.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes the dataset is produced using a public model. The weights of our vision backbone are also open-sourced. The decoder is trained from scratch. We include results in the ablations in the appendix section for models trained on open-source YFCC100M data. We will release training code and model inference code. The part that cannot be open-sourced is the WebLI dataset. We will attempt to release the model trained on WebLI dataset though.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

https://doi.org/10.52202/079017-3530

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The base open-source dataset with which results can be replicated is YFCC100M. The object detector used is OWL-ViTv2. The vision backbone is SigLIP pretrained SO400M.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We mention these details in the implementation section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: Error bars are not included as conducting a single run of pre-training is very resource intensive due to the large size of the dataset.

   Guidelines:

https://doi.org/10.52202/079017-3530

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Dataset size, model size, batch size, hardware and number of training steps have been mentioned in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the code of ethics and this work does not violate its guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Refer to Broader Impact section of the paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Models will be released after safeguards are put in place.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All work that is directly used in development of data and models used in this paper have been appropriately cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

https://doi.org/10.52202/079017-3530

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: New assets are not released right now.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work did not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work did not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-3530