What does CLIP know about a red circle? Visual prOmpt engineering for VLMs

Aleksandar Shtedritski Christian Rupprecht Andrea Vedaldi
Visual Geometry Group, University of Oxford
{suny, chrisr, vedaldi}@robots.ox.ac.uk

Abstract

Large-scale Vision-Language Models, such as CLIP, learn powerful image-text representations that have found numerous applications, from zero-shot classification to text-to-image generation. Despite that, their capabilities for solving novel discriminative tasks via prompting fall behind those of large language models, such as GPT-3. Here we explore the idea of visual prompt engineering for solving computer vision tasks beyond classification by editing in image space instead of text. In particular, we discover an emergent ability of CLIP, where, by simply drawing a red circle around an object, we can direct the model's attention to that region, while also maintaining global information. We show the power of this simple approach by achieving state-of-the-art in zero-shot referring expressions comprehension and strong performance in keypoint localization tasks. Finally, we draw attention to some potential ethical concerns of large language-vision models.

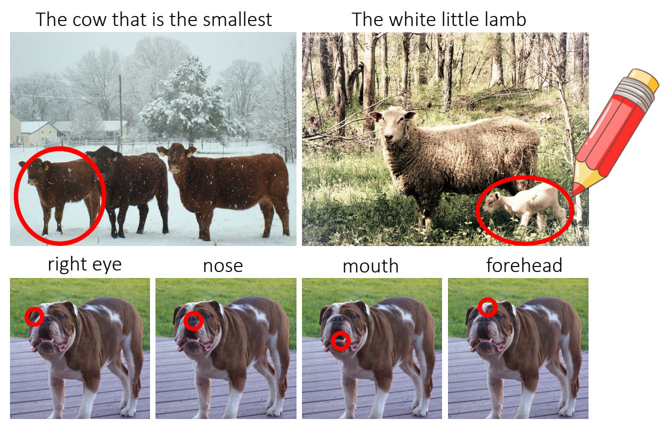

Figure 1: **Visual Prompt Engineering.** We draw multiple annotations over an image and have CLIP choose the correct one given a caption. Here we show predictions for the given expressions. Top: Examples from RefCOCOg on referring expressions detection. Bottom: Example from SPair71k on keypoint localization.

1. Introduction

Large Language Models (LLMs) such as GPT-2/3 [7, 39] and ChatGPT [1] have demonstrated surprising emerging behaviours. For example, these models can perform language translation without being explicitly trained for it, in a zero-shot manner. This can be partially explained by the fact that occurrences of the desired behaviours, such as translating between two languages, naturally occur in their enormous training corpus, which is, essentially, the Internet.

Interesting emergent behaviours have been observed in large Vision-Language Models (VLMs) like CLIP [38] too. For example, CLIP can be used for zero-shot classification by checking the compatibility of a given image with prompts such as "an image of a X", where X is one of a set of class hypotheses to be tested.

Emergent behaviours are elicited by supplying suitably crafted inputs to the VLMs, often called *prompts*. As in the example above, researchers have mostly focused on *engineering textual prompts*, manipulating the textual input of the model. This approach is inspired by LLMs, where manipulating the textual modality is the only available option. However, VLMs are inherently multimodal and offer the possibility of manipulating both modalities, textual and visual. While the textual modality is the natural choice for expressing semantics, the visual modality can be better for expressing geometric properties such as location.

In this paper, we thus explore *visual prompt engineering* [1]. We do so with two goals. The first goal is to contribute one more practical tool for extracting useful information from VLMs in a zero-shot manner. We demonstrate this by obtaining state-of-the-art zero-shot results in referring expressions comprehension by engineering visual prompts. The second goal is to characterise interesting and

[1]Note the difference between visual prompt *tuning*, a setting previously explored, where the prompts are task-specific learnable tokens, and visual prompt *engineering*, where we apply a fixed augmentation in pixel space.

unexpected properties of the VLMs and their training data, including identifying some behaviours that can raise ethical concerns.

Perhaps the most surprising of our findings is the effectiveness of a particular type of visual prompting: **drawing a plain red circle** on top of the image (Fig. 1). We show that this simple intervention steers the VLM to analyse/talk about the image region contained in the circle. This behaviour can then be used for tasks such as naming a specific object or object part or detecting particular image regions based on a description. The latter, for instance, is achieved by marking each object proposal with a red circle and using the VLM to find the best match with respect to the provided referring expression, achieving strong results on multiple benchmarks in the unsupervised regime. Furthermore, we show that prompting with a circle also works for finer-grained localization, marking specific object parts or keypoints instead of just whole objects.

We further contrast marking an image with the alternative of cropping it, which, from sliding window classifiers to region neural networks, is the canonical approach to steer the focus of an image-level predictor to a particular image region. We show that, for VLMs at least, marking is significantly more effective than cropping, possibly because it does not lose contextual information like the latter.

Apart from the practical applications, our findings reveal unexpected and intriguing properties of VLMs. We show empirically, that marking with a red circle is optimal among a selection of possible markers (variants of the circle, boxes, arrows, etc.). Presumably, the VLMs understand red circles out of the box because these appear sufficiently frequently in the training corpus, i.e., the Internet. While we do not have access to the full training data of CLIP, we corroborate this intuition by seeking examples of such images in YFCC15M, a dataset of CC-BY images.

Our analysis shows that red circles are indeed present even in a (comparatively small) dataset of images like YFCC15M, but they are *rare*. It is a testament to the extraordinary capacity of VLMs that such a behaviour can be learned from such rare events, without an explicit focus on doing so. We test models of different sizes/capacities and show that only the larger models exhibit this behaviour reliably, which corresponds to our intuition.

Finally, we note that the ability of VLMs to learn even from rare events such "red circles" can acquire both desirable and undesirable behaviours. Red circles, in particular, can have a negative connotation in the training data as they are often used by news outlets to mark missing people or criminals and, evidently, the model learns from such examples. As a result, we show that drawing a red circle in an image increases the probability that the model would characterise a person as a criminal or as a missing person.

To summarise, we make the following main contributions: (1) We propose marking as a new form of visual prompt engineering that is effective in extracting useful emergent behaviours in VLMs like CLIP; (2) We use the latter to achieve state-of-the-art zero-shot referring expressions comprehension using a VLM; (3) We provide an analysis of why marking is effective for these models, and link that to the training data and large model capacity; (4) We show that visual prompt engineering can also elicit unwanted behaviours, such as triggering problematic biases in the VLMs, revealing potential ethical issues.

2. Related work

Emergent Behaviour from Large Scale Pretraining has mainly been observed in Large Language Models (LLMs). Most notably, GPT-2 [39], GPT-3 [7], and ChatGPT [1] have been shown to be capable of tasks such as zero-shot translation, question answering, arithmetic, as well as planning actions for embodied agents [18]. Finetuning LLMs can also lead to models that can generate code from docstrings [8] or solve math problems [13, 23]. Only a few emergent zero-shot behaviours have been reported for VLMs like CLIP, mainly for classification [38] and OCR [33]. Generative VLMs like FLAMINGO [3] and BLIP [24] excel in captioning and visual question-answering tasks, but also have no way of solving pixel-level computer vision tasks.

Prompting VLMs is most commonly performed by prepending a set of learnable tokens to the text input [16, 21, 65, 66], vision input [19, 46, 64], or both text and vision inputs [41, 60], in order to easily steer a frozen CLIP model to solve a desired task. [4] learn augmentations in pixel space, such as padding around the image, or changing a patch of the image, which are optimized with gradient descent on a downstream task. [5] cast image inpainting as a visual prompting task, using a generative model trained on figures from academic papers. Coloring regions of an image has been used for the VCR task [61], where a model is fine-tuned on annotated images [62]. Colorful Prompt Tuning (CPT) [57] color regions of an image and use a captioning model to predict which object in an image an expression refers to by predicting its color. Similarly to CPT, we augment the input image in pixel space and perform zero-shot inference. However, we *annotate* the image in a human-like manner and show that our method is more powerful and more flexible than CPT.

Referring Expression Comprehension (REC) aims to localize a target object in an image that corresponds to a textual description. Most approaches to REC start with object proposals, for example, generated with Faster-RCNN [40], and learn to score them [17, 30, 31, 48, 54]. REC is sometimes considered together with referring expression generation — the task of generating a description of a given region. [31] use a comprehension model to guide a generator, whereas [9] jointly train a detector with a caption generator. Some works model the scene as a graph [30, 48, 54] or use language parsers and grammar-based methods [12, 29], leading to a more interpretable result. More recently, transformer architectures have been used [14, 22, 25, 50, 55]. [14, 22, 25, 55] perform text-modulated object detection, where a transformer decoder takes the referring expression as an input and predicts a bounding box. [50] train with a text-to-pixel contrastive loss, which allows for a text-driven segmentation or detection at test time.

Unsupervised Referring Expression Comprehension is a less explored area, only made possible with the introduction of large pre-trained models such as CLIP [38]. ReCLIP [44] crops object proposals and ranks them using CLIP before an ad-hoc postprocessing step to take into account relations such as left/right, smaller/bigger, etc. CPT [57] colors object proposal boxes and use a pretrained captioning model [63] to auto-regressively predict which colored proposal corresponds to the query description. Pseudo-Q [20] generates descriptions for multiple objects in an image, which is used to train a REC network. However, this model is not fully unsupervised as the pseudo descriptions it uses are generated using a captioning model trained on COCO.

Visual Reasoning Using Large Pretrained Models has been an area of significant interest in the last few years. In addition to referring expression detection [44], CLIP [38] has been used for semantic segmentation [27, 37]. [37] use CLIP to assign text labels to object parts after doing part co-segmentation in the latent space of a GAN. [27] utilize CLIP for open-vocabulary segmentation by using a general-purpose mask proposal network and CLIP as a classifier. CLIP has also been used for unsupervised object proposal generation [42] and open-set detection [15]. Semantic segmentation also emerges from image only [34, 49] or image-text [53] self-supervision.

Bias of VLMs is an increasingly popular area of research, as downstream applications come with the risk of perpetuating biases and stereotypes existing in the training data. However, methods for assessing the bias of a VLM are still not well established. [2] measure the misclassification rate of CLIP of faces of people of different races with non-human and criminal categories, whereas [6, 11, 47] measure fairness in retrieval results. Here, we show a different kind of bias, where the addition of a red circle over a person can trigger a negative connotation.

3. Method

Our goal is to develop visual prompting in *Vision-Language Models* (VLMs). VLMs solve prediction tasks that involve jointly processing text and images. For example, models such as CLIP are trained to *match* text and image samples. The input to such a VLM is an image $i \in \mathbb{R}^{3\times H\times W}$ and text $t \in \Sigma^*$, where Σ is an alphabet. The output is a score $s(i,t)$ that expresses the degree of compatibility between the supplied image and text.

3.1. Prompt engineering

One of the most striking capabilities of VLMs is their ability to solve a variety of classification tasks with little to no further training at all, in a zero-shot manner. This is done by reducing the task of interest to that of evaluating the VLM on suitably-engineered image and text pairs.

For example, given an image-caption pair (i,t), consider the problem of localizing a named object keypoint in the image. We can cast this as a question-answer problem, where the question $q \in Q$ is the name of the object keypoint (e.g., "right ear", "front left leg", ...) and the answer $a \in A$ is one of a discrete set of image locations.

Because the VLM computes a compatibility score $s(i,t)$ between an image i and the text t, it cannot be used to map the question q to the answer a directly. However, via prompt engineering, we can use the VLM to construct a compatibility score $s(q,a|i,t)$ between question and answer, conditioned on the input image-text pair (i,t). This score is in general given by the expression

$$s(q,a|i,t) = s(i_{qa}, t_{qa}), \tag{1}$$

where i_{qa} and t_{qa} are versions of the input image and text, obtained by transforming the latter to reflect the question-answer pair (q,a).

The specific way Eq. (1) should be applied to a problem depends on the specific nature of the latter. For example, in the problem of localizing the named keypoints, it is natural to encode the name of the keypoint via the textual modality and its 2D location via the visual modality. For instance, in order to answer the question $q =$ "right ear" for a given input image i with caption $t =$ "dog", we can engineer the textual prompt $t_{qa} = t_q =$ "an image of the right ear of a dog" to encode a description of the named entity. Likewise, we can engineer the visual prompt $i_{qa} = i_a$ in such a way as to 'select' the location a in the image, using one of the methods discussed in Section 3.2. With this, we can answer the question by finding $\hat{a}(q|i,t) = \operatorname{argmax}_{a\in A} s(q,a|i,t)$, that maximizes the score $s(q,a|i,t) = s(i_a, t_q)$, which specializes Eq. (1).

In the following sections, we provide further details and apply these ideas to a few concrete tasks.

3.2. Visual prompting via marking

The usual way of encoding location information in a visual prompt is to crop the image around the desired location, meaning that i_a is the image cropped around a. This idea has been used extensively with VLMs, including to interpret referring expressions, where maximizing a score of the

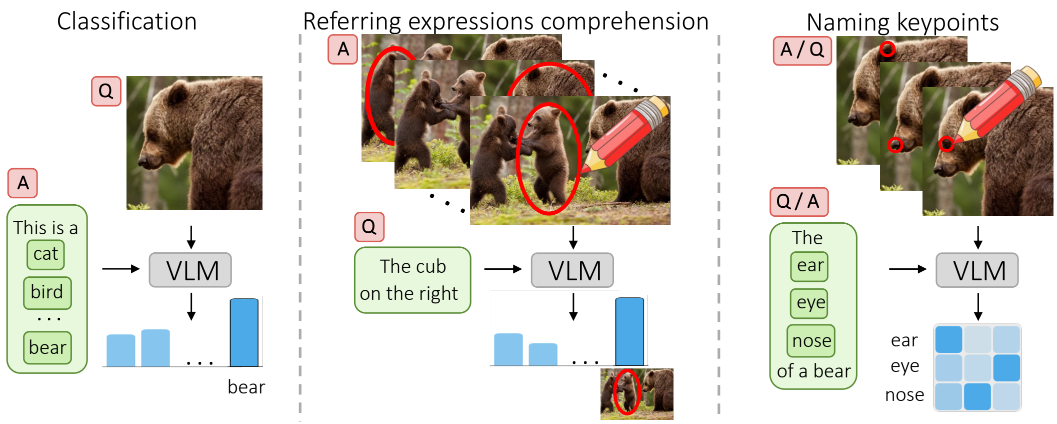

Figure 2: **Prompt engineering for VLMs.** We cast zero-shot inference with VLMs as a Q/A problem, each requiring specific prompt engineering. In the figure, Q is *Question* and A is *Answer* (a set of possible answers). Left: text prompt engineering for classification. This widely used method can be interpreted as follows in our framework: The image is the question, and classes are the available answers, which are engineered into prompts. Middle: visual prompt engineering for referring expressions comprehension. The question is the referring expression, and the available answers are the box proposals, which we engineer into visual prompts. Right: visual and text prompt engineering for keypoint matching. For keypoint localization, we use a similar setup to referring expressions, where the question is a keypoint in plain text and the possible answers are all 2D locations in the image.

form $s(i_a, t_q)$ seeks for the image crop that best matches the referring expression t_q.

In this paper, we explore an alternative approach for visual prompting that uses the concept of *marking* the desired region in the image. Marking quite literally means overlaying to the image i_a a circle, a box, or an arrow, which visually indicates the desired location a.

While the idea of marking may sound strange, it is interesting for two reasons. First, differently from cropping, a marked image i_a preserves almost all the information contained in the input image i, including contextual information that crops lack. Second, we show that marking works well with VLMs, *outperforming* cropping-based prompt engineering in some prediction tasks.

While the simplest marking consisting of a red circle is particularly effective, in Section 4 we explore several different ways of generating markings. We refer the reader to that section for further details and examples.

3.3. Tasks

We study the idea of mark-based prompt engineering by considering several zero-shot prediction tasks, from simple tasks such as matching keypoints to their names to more complex ones such as referring expression comprehension.

Naming Keypoints. The first and simplest task that we consider is matching the name of the keypoints of an object to their 2D locations in an image. The input is an image i, a set of keypoint names Q, and a set of corresponding keypoint locations $A \subset \{0, \dots, H-1\} \times \{0, \dots, W-1\}$. The number of names and locations is the same ($m = |Q| = |A|$) and the goal is to match the two. We express the latter as predicting the square permutation matrix $\Pi \in S_m$ that associates each name q to its corresponding location a (i.e., $\Pi_{qa} = 1$).

In order to predict Π, we use Eq. (1) to define the cost of associating name q to location a as $C_{qa} = s(i_a, t_q)$ where i_a is obtained either via cropping or marking and t_q is just the name of the keypoints prefixed by the string "an image of". For this problem, the role of questions and answers is symmetric and we decode the cost matrix C into a permutation matrix Π via optimal transport:

$$\hat{\Pi}(i, Q, A) = \operatorname*{argmax}_{\Pi \in S_m} \sum_{q \in Q, a \in A} \Pi_{qa} \exp\left(-\tau C_{qa}\right), \quad (2)$$

where $\tau > 0$ is a temperature parameter. This optimization problem is solved efficiently via the Sinkhorn-Knopp algorithm [43], which renormalizes matrix C.

Keypoint Localization. The second task is a more useful and difficult variant of the first. The goal is still to localize a named keypoint q in an image, but this time the locations A are a subset of a $m \times m$ regular grid. These are further restricted to a salient image region extracted by using the unsupervised saliency method of [49] to avoid testing irrelevant locations in the background. The difference compared to naming keypoints is that this version of the problem does not assume prior knowledge of the possible locations of the keypoints. Given the name q of a keypoint, its location a is then obtained as $\hat{a}(i, q) = \operatorname{argmax}_{a \in A} s(i_a, t_q)$ where i_a and t_q are as defined previously.

Method	Name-to-keypoint							Keypoint-to-name						
	CUB	**Spair71k**						**CUB**	**SPair71k**					
Random	8.2	16.8	15.0	10.5	9.4	15.1	11.9	8.2	16.8	15.0	10.5	9.4	15.1	11.9
Crop w/o SK	15.8	28.5	28.5	28.5	20.1	26.1	29.7	18.7	22.4	19.1	24.0	14.9	27.3	25.1
Crop w/ SK	25.5	35.1	37.5	34.6	23.9	32.9	36.3	25.8	36.1	32.5	32.7	19.8	35.3	32.5
Red Circle w/o SK	46.5	54.8	53.1	51.6	40.1	47.4	45.2	29.5	26.8	24.9	36.9	18.8	31.8	28.9
Red Circle w/ SK	**58.2**	**67.6**	**60.1**	**59.3**	**53.1**	**56.7**	**52.8**	**56.5**	**67.2**	**54.4**	**59.7**	**49.8**	**56.6**	**53.0**

Table 1: **Naming keypoints results on CUB and SPair71k**. On SPair71k, we show results on all animal classes — bird, cat, dog, horse, sheep, cow. We show the percentage of correctly matched keypoints and names, given a list of them. We compare to randomly guessing the correct correspondence and cropping around the region of interest, rather than drawing an annotation. We compare all methods with and without normalization with the Sinkhorn-Knopp (SK) algorithm.

Referring Expression Comprehension. Comprehending a referring expression means detecting an object in an image that corresponds to a textual specification that explicitly refers to it (e.g., "fourth dog from the right"). Similarly to prior work [20, 44, 57], given an image i, we approach this problem by extracting first a set of object proposals using the method from [58] and interpret those as the set of possible answers A. The set of questions Q is instead a collection of referring expressions extracted from a given benchmark dataset. For each referring expression, the best matching proposal is then given by

$$\hat{a}(i,q) = \operatorname*{argmax}_{a\in A} \left[s(i_a, t_q) - \frac{1}{|Q|}\sum_{\bar{q}\in Q} s(i_a, t_{\bar{q}}) \right]. \quad (3)$$

The engineered prompts i_a and t_q are defined as in Section 3.3. In this case, we found it useful to subtract from the score the average with respect to all possible referring expressions Q. This weighs down hypotheses a such as faces that are visually very salient and tend to respond very strongly to all questions q.

4. Experiments

We study the properties of visual marking in VLMs by considering first the three tasks of Section 3.3: naming keypoints, localizing keypoints, and referring expression comprehension.

4.1. Naming Keypoints

Naming keypoints is a comparatively simple problem that has no direct application; however, it is simpler and faster to evaluate than the other tasks, so we use it to ablate various aspects of our method.

Data and implementation details. For this task, we consider the CUB-200-2011 (CUB) [51] and SPair71k [35] datasets. The first contains named keypoint annotations for each image, whereas the second only annotates matching keypoints in pairs of images, but does not name them. We

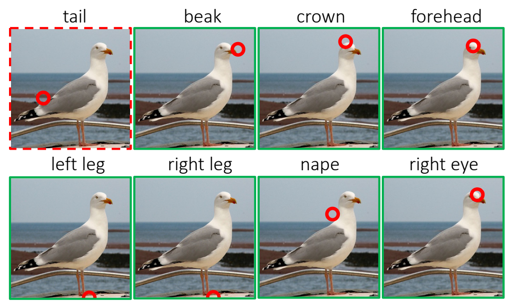

Figure 3: **Qualitative Results on Localizing Keypoints** on an image from SPair71k. Green and red (dashed) borders are for correct and wrong predictions according to PCK with $\alpha = 0.1$. The red circle shown is thicker than the one used; see the sup. matt. for examples of the actual thickness.

thus augment the latter, manually naming each keypoint instance in each animal image. We further crop the images from SPair71k with the provided bounding boxes. For the VLM, we use the ViT-L/14@336px backbone. Please see the sup. matt. for details.

Results. Recall that, in this task, the output of the predictor is a permutation matrix Π associating each keypoint location to a corresponding name. We report (i) the ratio of keypoint names that are mapped to the correct location and (ii) the ratio of keypoint locations that are mapped to the correct names. To the best of our knowledge, there are no prior works that associate keypoints with their names. We thus compare the result of this new task to (a) random choice and (b) a baseline where i_a is obtained by cropping.

As seen in Table 1, prompting via visual marking (red circles) significantly outperforms the baselines, achieving almost twice the accuracy. Using the Sinkhorn-Knopp (SK) algorithm to normalize the matching score further boosts results, mainly improving results for points that are ambiguous and close to each other, *e.g.*, mouth and nose.

What is the best visual marker? We compare the use of (i) different shapes for highlighting a location: circle,

Marker shape	Mean	Best
Circle	**33.5 ± 4.5**	**46.5**
Arrow	28.3 ± 3.1	36.3
Square	24.1 ± 3.6	36.3
Cross	21.5 ± 6.3	34.5

Circle color	Mean	Best
Red	**36.4 ± 5.1**	**46.5**
Green	34.3 ± 4.2	43.3
Purple	34.0 ± 3.7	41.9
Blue	32.7 ± 3.9	41.1
Yellow	32.4 ± 4.0	40.8

Table 2: **Ablation of annotation types for naming keypoint.** We evaluate on CUB and present results across a variety of colors and sizes (for marker shape) and sizes (for circle color). For full results refer to the sup. matt.

Figure 4: **Example Annotations for Keypoints.** We show how the annotations we use look for different shapes and colors. We also experiment with multiple thicknesses and sizes. The red circle annotation on the left is the one we use throughout all evaluations unless stated otherwise.

rectangle, cross, arrow, (ii) different sizes, and (iii) different colors of the annotations, and show some examples in Fig. 4. We compare different shapes and colors in Table 2 and find that red circles perform best. Red is the best color despite the fact that it is a commonly occurring color in images, unlike colors like purple which can be found less often in nature and can thus be more distinctive, but lead to worse performance. We attribute this to the fact that this emergent capability of CLIP exists due to human-centric manipulations of its training data, and humans are likely to annotate using red circles, as shown next.

Are there visual markers in the training data? To explore the hypothesis that CLIP can zero-shot classify annotations on images because of similar examples seen during training, we find images in YFCC15M that contain markers (YFCC15M is a subset of the CLIP training data). To this end, we train a binary classifier using an ensemble of a ViT-B/16 and RN50x16 CLIP vision encoders to classify images in YFCC15M that contain annotations. We then use this to filter a 6M subset of YFCC15M and take the top 10k images with the highest score. Finally, we manually examine the 10k images and find 70 images that have annotations drawn on top of them. We show 3 such images in Fig. 5. Hence, the training data contains examples of markers, but they are very rare (∼0.001%), suggesting that such behavior can only be learned from very large datasets by high-capacity models. This is further explored next.

How do different VLMs differ? We compare a number of CLIP models in Fig. 6. In general, we observe

Figure 5: **Discovered annotations in YFCC15M.** We show images from YFCC15M that have been human-annotated, and their corresponding captions. Discovered by training a simple detector.

Method	Backbone	Data	Params	Name-to-keypoint CUB	Name-to-keypoint Spair71k	Keypoint-to-name CUB	Keypoint-to-name SPair71k
CLIP [38]	ViT-B/32	400M	87M	19.1	26.7	19.1	25.6
CLIP [38]	ViT-B/16	400M	86M	22.2	34.0	22.1	33.6
CLIP [38]	RN50x16	400M	167M	30.6	41.0	29.7	40.0
CLIP [38]	ViT-L/14	400M	304M	47.9	54.3	48.0	51.2
CLIP [38]	ViT-L/14⋆	400M	304M	**58.2**	**58.3**	**56.5**	**56.8**
OpenCLIP [10]	ViT-B/32	2B	87M	19.4	27.5	20.7	27.2
OpenCLIP [10]	ViT-L/14	2B	304M	33.9	42.4	33.3	41.5
OpenCLIP [10]	ViT-H/14	2B	632M	45.0	53.7	42.8	50.5
OpenCLIP [10]	ViT-g/14	2B	1.01B	44.2	47.2	42.5	43.6
OpenCLIP [10]	ViT-G/14	2B	1.84B	50.4	52.5	48.9	48.6
SLIP [36]	ViT-S/16	15M	22M	13.0	17.8	12.0	16.7
SLIP [36]	ViT-B/16	15M	86M	17.3	16.5	16.7	17.1
SLIP [36]	ViT-L/16	15M	303M	24.6	26.3	24.1	25.0
CLIP† [36, 38]	ViT-S/16	15M	22M	11.4	16.5	12.4	15.8
CLIP† [36, 38]	ViT-B/16	15M	86M	14.0	18.4	14.8	17.8
CLIP† [36, 38]	ViT-L/16	15M	303M	15.0	20.5	15.8	20.4
FILIP‡ [26, 56]	ViT-B/32	15M	90M	8.9	15.6	8.7	15.6
DeFILIP [26]	ViT-B/32	15M	90M	12.5	19.5	12.5	19.4
CLIP‡ [26, 38]	ViT-B/32	15M	90M	10.6	14.1	11.0	14.3
DeCLIP [26]	ViT-B/32	15M	90M	15.8	19.9	15.8	18.7
DeCLIP [26]	ViT-B/32	88M	90M	19.4	23.6	19.7	22.3

Table 3: **Comparison of backbones and pretraining methods**. We compare several VL pretraining methods, implementations, and backbones. † and ‡ : we evaluate the models trained by [36] and [26], respectively. ⋆ : we evaluate a model trained on images of input size 336 × 336. For SPair71k, we present the average across all animal classes. The 400M, 2B, 15M, and 88M datasets are WIT-400M, LAION-2B, YFCC-15M and DeCLIP-88M, respectively.

that the performance of keypoint matching improves with (i) the size of the pretraining dataset and (ii) the size of the vision encoder. The former holds true for CLIP models trained on WIT-400M vs YFCC-15M (which is a subset of WIT-400M). However, using the LAION-2B dataset for pretraining leads to worse results. We suspect this result comes from differences in filtering when creating WIT-400M and LAION-2B, where in the latter, examples of annotations might have been discarded due to a stronger focus on *aesthetic* images for generative models. Similarly, we see big gains in performance as we increase the size of the vision encoder, and the gains do not seem to converge with the biggest available models. We emphasize on the dramatic increase in performance of the WIT-400M pretrained CLIP — the biggest models improve on the performance of

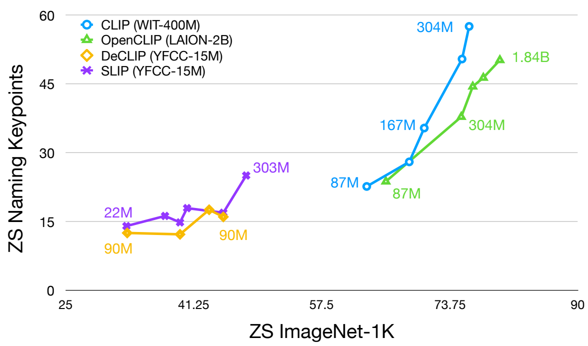

Figure 6: **Comparison of Zero-shot (ZS) Naming Keypoints to ZS ImageNet-1K.** We show results for CLIP [38], OpenCLIP [10], DeCLIP [26] and SLIP [36]. We compute the average text-to-image and image-to-text matching score on SPair71k and CUB and compare it against the reported zeros-shot ImageNet-1K accuracy for each model. For all methods, we show the pretraining dataset and the number of parameters of the vision encoders of some of the models.

Method	Mask	CUB	Spair71k					
Random	✗	1.1	3.1	1.8	3.6	3.0	3.0	3.4
Random	✓	8.3	6.7	4.7	4.6	5.5	3.3	4.3
Crop	✗	16.9	25.5	27.3	22.6	16.1	20.0	22.7
Crop	✓	21.3	28.4	28.8	21.9	16.0	23.4	24.5
Red Circle	✗	32.3	50.7	52.7	55.9	38.3	42.4	38.0
Red Circle	✓	**45.2**	**53.4**	**54.5**	**56.4**	**40.9**	**43.2**	**42.2**

Table 4: **Named Keypoint Localization Results.** We report PCK of predicted keypoint location and compare to a baseline where we crop around the region.

the smallest by 250% on ZS keypoint matching, whereas the improvement on ZS ImageNet-1K classification is just 20%. We draw similarities between this task and tasks in the domain of NLP, such as zero-shot or one-shot arithmetic, where only the largest GPT models perform well [7]. We argue that in a similar fashion, the vision encoder needs sufficient capacity and data in order to show this emergent behavior.

4.2. Localizing Keypoints

For this experiment, we use the same data and network architecture as for the previous one, but report the *percentage of correct keypoints* (PCK) as a metric, as the latter is widely used when evaluating semantic correspondences. Given a set of ground-truth points $\mathcal{P} = \{p_m\}_{m=1}^M$ and predictions $\hat{\mathcal{P}} = \{p_m\}_{m=1}^M$, PCK is given by:

$$\text{PCK}(\mathcal{P}, \hat{\mathcal{P}}) = \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}[\|\hat{p}_m - p_m\| \leq \delta].$$

Method	ZS	RefCOCO Val	RefCOCO TestA	RefCOCO TestB	RefCOCO+ Val	RefCOCO+ TestA	RefCOCO+ TestB	RefCOCOg Val	RefCOCOg Test
DTWREG [45]	✗	39.2	41.1	37.7	39.2	40.1	38.1	–	–
Pseudo-Q [20]	✗	56.0	58.3	54.1	38.9	45.1	32.1	46.3	47.4
CPT [57]	✓	32.2	36.1	30.3	31.9	35.2	28.8	36.7	36.5
ReCLIP † [44]	✓	42.0	43.5	39.0	47.4	50.1	43.9	57.8	57.2
ReCLIP ‡ [44]	✓	45.8	46.1	**47.1**	47.9	50.1	45.1	59.3	**59.0**
Red Circle	✓	**49.8**	**58.6**	39.9	**55.3**	**63.9**	**45.4**	**59.4**	58.9

Table 5: **Comparison with state-of-the-art on REC.** We report top-1 accuracy (%). † is the crop-based baseline of [44] and ‡ is their method that adds relational resolving. *"ZS"* refers to zero-shot approaches. Drawing red circles outperforms other zero-shot approaches on most benchmarks, including ReCLIP‡ that post-processes results using manually designed relational rules. On RefCOCO+ and RefCOCOg, a red circle also outperforms Pseudo-Q and DTWREG that are not zero-shot and use weak supervision.

Here, δ is a distance threshold given by $\delta = \alpha \max(H, W)$, where $0 < \alpha < 1$ is a ratio and (H, W) is the bounding box size. For all datasets we use $\alpha = 0.1$. Keypoint localization also utilizes an unsupervised saliency mask to ignore background locations.

Similarly to the naming task, we compare keypoint localization to random guessing and the crop-based baseline. As shown in Table 4, using red circles significantly outperforms both; as expected, results are further improved by using saliency to further filter keypoint locations. We show qualitative results in Fig. 3.

4.3. Referring Expression Comprehension

Datasets and implementation details. Referring expression comprehension is commonly evaluated on the RefCOCO [59], RefCOCO+ [59], and RefCOCOg [32] datasets, all of which consist of images from the MS-COCO dataset [28] together with expressions that refer to a unique object in the image, which are also annotated with a bounding box. RefCOCO+ only contains appearance-based expressions, whereas RefCOCO and RefCOCOg contain relation-based expressions (e.g., containing the words left/closer/bigger). The test sets of RefCOCO and RefCOCO+ are split in two, where "testA" and "testB" contain only people and non-people, respectively. We evaluate using the percentage of correct predictions, where a box is correctly predicted if its intersection-over-union with the ground-truth box is over 0.5.

For the referring expressions task, we use an ensemble RN50x16 and ViT-L/14@336 CLIP backbones. Following prior work [44, 57], we score the bounding box proposals of MAttNet [58].

Results. Using a red circle, we achieve state-of-the-art on most referring expressions comprehension baselines in the zero-shot setting, as shown in Table 5. Interestingly, this even outperforms ReCLIP [44], which is based on scor-

Model	Red Circle	FaceSynthetics			COCO		
		Positive	Neutral	Criminal	Positive	Neutral	Criminal
ViT-L/14@336px	✗	0.5%	35.6%	63.9%	22.7%	47.5%	29.8%
ViT-L/14@336px	✓	0.0%	19.1%	80.9% (+17.0%)	1.6%	505.%	47.9 (+18.1%)
ViT-L/14	✗	1.3%	40.8%	57.9%	25.0%	46.5%	28.6%
ViT-L/14	✓	0.0%	32.5%	67.5% (+9.6%)	2.7%	43.1%	54.2 (+25.6%)
ViT-B/16	✗	0.7%	49.1%	50.2%	19.4%	56.7%	23.9%
ViT-B/16	✓	0.0%	37.1%	62.9 (+12.7%)	4.7%	39.3%	56.0 (+32.1%)
ViT-B/32	✗	0.0%	85.2%	14.8%	15.3%	61.8%	22.9%
ViT-B/32	✓	0.0%	70.5%	29.5% (+14.7%)	4.9%	48.2%	46.9 (+24.0%)
RN50x16	✗	0.7%	46.4%	52.9%	26.5%	30.5%	43.0%
RN50x16	✓	0.1%	52.2%	47.7% (-5.2%)	13.6%	39.5%	46.9 (+3.9%)

Table 6: **Classification rate into criminal categories.** We report ZS classification results in criminal categories on synthetic faces from FaceSynthetics and persons from COCO, with and without red circles. In brackets is the absolute difference from the crop without a red circle.

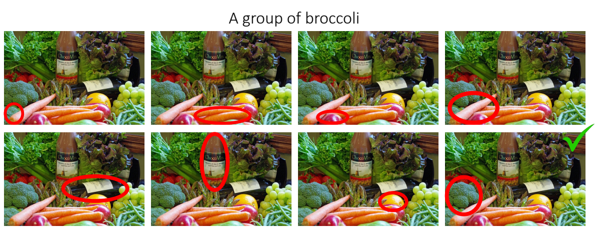

Figure 7: **Qualitative Evaluation on REC** on an image from RefCOCOg. CLIP correctly chooses the bottom right image. The circle is shown thicker than actually used for clarity (see the sup. matt. for the actual thickness).

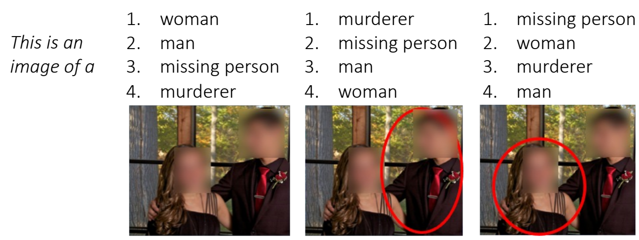

Figure 8: **Bias of CLIP.** We zero-shot classify a random COCO image with a male-looking and a female-looking individual, also adding a circle over each of them. We score the following sentences to the images: *This is an image of a* {*woman, man, missing person, suspected murderer* }. The apparent gender resolution is correct, but the circled images tend to be scored higher as missing or murderers. Blur added for privacy.

ing image crops, followed by post-processing with manually designed relations rules. A red circle also outperforms Pseudo-Q on most benchmarks, even though Pseudo-Q explicitly trains for this task.

4.4. Model biases and ethics

While drawing circles on images can extract useful behaviors from a VLM for a wide variety of legitimate image analysis tasks, it can also extract unwanted ones and **must not be used for the analysis of sensitive data**.

To demonstrate this fact, in Fig. 8 we take a random image from COCO that contains a male-looking and a female-looking individual and zero-shot classify the image, as well as the image with circles over each individual, into 4 categories: male, female, missing person, and suspected murderer. While this leads to a correct resolution of the apparent gender of the annotated person, the annotated images are more likely to be classified as containing a missing person or a murderer. While we cannot know for certain, we hypothesize that this is due to the presence in the CLIP model training data of missing person reports, police footage, or similar, where people have been marked.

We further quantify these biases following [2], using synthetic faces from FaceSynthetics [52] and person crops from COCO [28]. For FaceSyntetics, we take 1000 random synthetic faces, and for COCO, we crop all bounding boxes for the class *person* from the validation set that have an area of at least 10% of the total area of the image, which comes down to 1352 crops. Following [2], we measure zero-shot classification rates into criminal categories. We introduce a "positive" category (honest man/woman/person), "neutral" category (man/woman/person) and a "criminal" category (criminal/thief/suspicious person). Finally, we zero-shot classify the original images and the images with circles. In Table 6 we present classification rates into criminal categories. We see that for all ViT encoders, the rate at which people are classified as criminals is significantly higher. This is problematic as such existing biases can lead to harmful consequences. Note that there are various limitations in this analysis, including the usage of binary gender.

5. Conclusions

We have shown that visual prompt engineering via marking can extract useful behavior from VLMs such as CLIP in a zero-shot manner, achieving state-of-the-art zero-shot referring expression comprehension performance, and significantly outperforming traditional techniques like image cropping. Our analysis suggests that this behavior emerges because relevant samples of marking exist in the training data of the VLMs, but these samples are very rare. As a consequence, the behavior can only be learned by very large models trained on very large datasets. The analysis also shows that VLMs acquire undesirable behaviors too, where the mere addition of a red circle to an image increases the model's belief that the image has a negative connotation.

Dataset Ethics. We use the RefCOCO, RefCOCO+, MS-COCO, FaceSynthetics, YFCC15M, CUB, SPair71k in a manner compatible with their terms. Some of these images may contain personal data (faces). In Sections 4.1 to 4.3 there is no extraction of biometric data. In Section 4.4 we use MS-COCO to demonstrate that such a method cannot reliably extract information about people due to the bias in the pre-trained CLIP model (there is no identification). The FaceSynthetics, used for the same purpose, is a dataset of synthetic faces, so it does not raise privacy concerns. For further details on ethics, data protection, and copyright please see https://www.robots.ox.ac.uk/~vedaldi/research/union/ethics.html.

Acknowledgements. We thank Luke Melas-Kyriazi, Tim Franzmeyer, Rhydian Windsor and Bruno Korbar for proofreading A. Shtedritski is supported by EPSRC EP/S024050/1. A. Vedaldi and C. Rupprecht are supported by ERC-CoG UNION 101001212. C. Rupprecht is also partially supported by VisualAI EP/T028572/1.

References

[1] Chatgpt. https://chat.openai.com/. 1, 2

[2] Sandhini Agarwal, Gretchen Krueger, Jack Clark, Alec Radford, Jong Wook Kim, and Miles Brundage. Evaluating clip: towards characterization of broader capabilities and downstream implications. *arXiv preprint arXiv:2108.02818*, 2021. 3, 8

[3] 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. *arXiv preprint arXiv:2204.14198*, 2022. 2

[4] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 2022. 2

[5] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A Efros. Visual prompting via image inpainting. *arXiv preprint arXiv:2209.00647*, 2022. 2

[6] Hugo Berg, Siobhan Mackenzie Hall, Yash Bhalgat, Wonsuk Yang, Hannah Rose Kirk, Aleksandar Shtedritski, and Max Bain. A prompt array keeps the bias away: Debiasing vision-language models with adversarial learning. *arXiv preprint arXiv:2203.11933*, 2022. 3

[7] 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. 1, 2, 7

[8] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. 2

[9] Yi-Wen Chen, Yi-Hsuan Tsai, Tiantian Wang, Yen-Yu Lin, and Ming-Hsuan Yang. Referring expression object segmentation with caption-aware consistency. *arXiv preprint arXiv:1910.04748*, 2019. 3

[10] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *arXiv preprint arXiv:2212.07143*, 2022. 6, 7

[11] Ching-Yao Chuang, Varun Jampani, Yuanzhen Li, Antonio Torralba, and Stefanie Jegelka. Debiasing vision-language models via biased prompts. *arXiv preprint arXiv:2302.00070*, 2023. 3

[12] Volkan Cirik, Taylor Berg-Kirkpatrick, and Louis-Philippe Morency. Using syntax to ground referring expressions in natural images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 3

[13] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 2

[14] Zi-Yi Dou, Aishwarya Kamath, Zhe Gan, Pengchuan Zhang, Jianfeng Wang, Linjie Li, Zicheng Liu, Ce Liu, Yann LeCun, Nanyun Peng, et al. Coarse-to-fine vision-language pre-training with fusion in the backbone. *arXiv preprint arXiv:2206.07643*, 2022. 3

[15] Sepideh Esmaeilpour, Bing Liu, Eric Robertson, and Lei Shu. Zero-shot open set detection by extending clip. *arXiv preprint arXiv:2109.02748*, 2021. 3

[16] Zixian Guo, Bowen Dong, Zhilong Ji, Jinfeng Bai, Yiwen Guo, and Wangmeng Zuo. Texts as images in prompt tuning for multi-label image recognition. *arXiv preprint arXiv:2211.12739*, 2022. 2

[17] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4555–4564, 2016. 2

[18] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022. 2

[19] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Vi-

sual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 2

[20] Haojun Jiang, Yuanze Lin, Dongchen Han, Shiji Song, and Gao Huang. Pseudo-q: Generating pseudo language queries for visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15513–15523, 2022. 3, 5, 7

[21] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 105–124. Springer, 2022. 2

[22] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021. 3

[23] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*, 2022. 2

[24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 2

[25] Muchen Li and Leonid Sigal. Referring transformer: A one-step approach to multi-task visual grounding. *Advances in Neural Information Processing Systems*, 34:19652–19664, 2021. 3

[26] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021. 6, 7

[27] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. *arXiv preprint arXiv:2210.04150*, 2022. 3

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence 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. 7, 8

[29] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4673–4682, 2019. 3

[30] Yongfei Liu, Bo Wan, Xiaodan Zhu, and Xuming He. Learning cross-modal context graph for visual grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11645–11652, 2020. 2, 3

[31] Ruotian Luo and Gregory Shakhnarovich. Comprehension-guided referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7102–7111, 2017. 2

[32] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 7

[33] Joanna Materzyńska, Antonio Torralba, and David Bau. Disentangling visual and written concepts in clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16410–16419, 2022. 2

[34] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8364–8375, 2022. 3

[35] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543*, 2019. 5

[36] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 529–544. Springer, 2022. 6, 7

[37] Daniil Pakhomov, Sanchit Hira, Narayani Wagle, Kemar E Green, and Nassir Navab. Segmentation in style: Unsupervised semantic image segmentation with stylegan and clip. *arXiv preprint arXiv:2107.12518*, 2021. 3

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 6, 7

[39] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1, 2

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[41] Sheng Shen, Shijia Yang, Tianjun Zhang, Bohan Zhai, Joseph E Gonzalez, Kurt Keutzer, and Trevor Darrell. Multitask vision-language prompt tuning. *arXiv preprint arXiv:2211.11720*, 2022. 2

[42] Hengcan Shi, Munawar Hayat, Yicheng Wu, and Jianfei Cai. Proposalclip: Unsupervised open-category object proposal generation via exploiting clip cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9611–9620, 2022. 3

[43] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967. 4

[44] Sanjay Subramanian, Will Merrill, Trevor Darrell, Matt Gardner, Sameer Singh, and Anna Rohrbach. Reclip: A strong zero-shot baseline for referring expression comprehension. *arXiv preprint arXiv:2204.05991*, 2022. 3, 5, 7

[45] Mingjie Sun, Jimin Xiao, Eng Gee Lim, Si Liu, and John Y Goulermas. Discriminative triad matching and reconstruction for weakly referring expression grounding. *IEEE*

transactions on pattern analysis and machine intelligence, 43(11):4189–4195, 2021. 7
[46] Cheng-Hao Tu, Zheda Mai, and Wei-Lun Chao. Visual query tuning: Towards effective usage of intermediate representations for parameter and memory efficient transfer learning. *arXiv preprint arXiv:2212.03220*, 2022. 2
[47] Jialu Wang, Yang Liu, and Xin Eric Wang. Are gender-neutral queries really gender-neutral? mitigating gender bias in image search. *arXiv preprint arXiv:2109.05433*, 2021. 3
[48] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1960–1968, 2019. 2, 3
[49] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *arXiv preprint arXiv:2209.00383*, 2022. 3, 4
[50] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11686–11695, 2022. 3
[51] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 5
[52] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Matthew Johnson, Virginia Estellers, Thomas J. Cashman, and Jamie Shotton. Fake it till you make it: Face analysis in the wild using synthetic data alone, 2021. 8
[53] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022. 3
[54] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4644–4653, 2019. 2, 3
[55] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Crossing the format boundary of text and boxes: Towards unified vision-language modeling. *arXiv preprint arXiv:2111.12085*, 2021. 3
[56] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021. 6
[57] Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Cpt: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*, 2021. 2, 3, 5, 7
[58] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018. 5, 7
[59] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016. 7
[60] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022. 2
[61] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731, 2019. 2
[62] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651, 2021. 2
[63] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2021. 3
[64] Sheng Zhang, Salman Khan, Zhiqiang Shen, Muzammal Naseer, Guangyi Chen, and Fahad Khan. Promptcal: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery. *arXiv preprint arXiv:2212.05590*, 2022. 2
[65] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 2
[66] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2