# Representational Task Bias in Zero-shot Recognition at Scale

**Anonymous authors**
Paper under double-blind review

## Abstract

Research from the last year has demonstrated that vision-language pre-training at scale from incidental supervision on the Internet can result in representations with clear advantages over traditional supervised training for many computer vision tasks. We conduct an in-depth exploration of the CLIP model, and find that the interface that language creates to these learned representations – by the same token as enabling zero-shot application for many tasks – leads the model to solve tasks that may not have been intended by the user in realistic scenarios. We call the inherent uncertainty of which task a user intends to solve in zero-shot recognition *task ambiguity*. We demonstrate that the representation produced for a given image tends to be strongly biased towards features for a particular task over others; in other words, they exhibit *task bias*. Moreover, which task a particular image will be biased towards is unpredictable, with little consistency across images. We construct a dataset of images where each image has labels for multiple semantic recognition tasks to evaluate task bias. Our results show that we can learn visual prompts to serve as effective conditioning mechanisms for which task is desired, and can even improve performance for the task when used outside the context of evaluating task ambiguity.

## 1 Introduction

Time for a quiz! Which of the choices in Fig. 1 is correct for the image shown? With only the information provided, there is no clear answer; the action being performed is *lift*, the text reads *football*, and the most prominent object is a *weight*. It is common sense that one cannot answer a question without a question. Yet in the current application paradigm, vision-language models (VLMs) such as CLIP (Radford et al., 2021) face decisions like this regularly in applied settings – and, unknown to users, the models have already decided which question to answer solely from the image. As shown on the right of Fig. 1, when CLIP is given these choices, it has a strong bias to prefer one of them, even if they are all equally correct.

The computer vision community often leverages large, hand-annotated datasets, such as the ImageNet benchmark (Russakovsky et al., 2015). However, a new paradigm has emerged that capitalizes

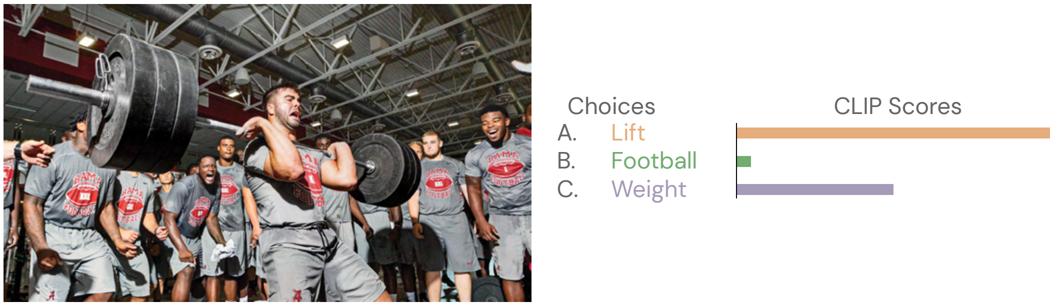

Figure 1: The CLIP visual representation of this image is strongly predisposed towards solving action recognition over scene text and object recognition.

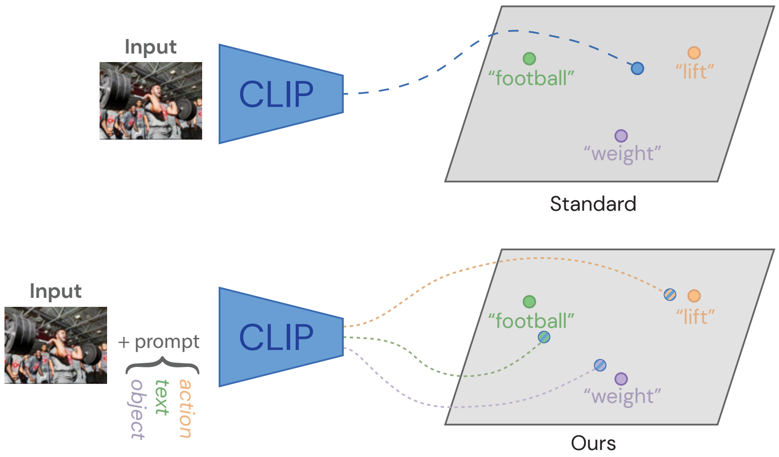

Figure 2: (top) We show the standard approach for CLIP where an image is embedded into a multi-modal space, and the nearest word embedding is retrieved to produce the final answer. However, when there are multiple correct options, this approach has a *task bias* where it prefers one task over the others. (bottom) We show to overcome this problem by learning a visual prompt that steers the representation towards the intended task.

on corresponding image-text pairs on the Internet to learn deep visual representations without strong human supervision. Natural language supervision enabled the field to graduate beyond pre-defined categories, achieved excellent zero-shot performance, and even powered compositional generalization to out-of-distribution samples on various visual recognition tasks Radford et al. (2021).

Language has made computer vision accessible for many applications (Ramesh et al., 2022; Majumdar et al., 2022; Ha & Song, 2022). By naming a few categories in natural language (and not even labeling any images) the ascendant paradigm of "zero-shot inference" using these vision-language models enables the field to quickly build systems to recognize objects, actions, text, and anything else that can be described in language. It is this approach that has set the state-of-the-art on benchmarks. However, typically in these benchmarks, they will be given a classification problem where multiple answers are valid responses, the correct one only depending on the task. Consequently, there can be multiple answers that are equally correct, but associated with entirely different tasks.

This paper shows that models such as CLIP have a *task bias* where, given an image, their representations will prefer to solve one task over another one. This task bias is not an isolated phenomenon, and we show it occurs for most images. Following the standard zero-shot application procedure that language enables, models can thus solve tasks that were not intended by the user in realistic scenarios. Such errors are a symptom of a deeper problem: it highlights that the visual representations of images with multiple potential tasks to be solved are specialized to one of these tasks, posing an issue for any downstream use of these representations (e.g., for visual similarity). We show that attempting to engineer better text prompts to clarify the task does not help with guiding the predictions to a desired task, and, of course, it cannot adjust the visual representations.

We propose a straightforward solution to the task bias problem that requires no modification to the model parameters, making the fix widely applicable. We show that we can learn a visual prompt for each task that guide the image representations towards the task of interest, thereby resolving task ambiguity. Figure 2 illustrates the method. The prompt is independent of the image, and experiments show it steers the visual representation to focus on the task-relevant parts of the image.

## 2 Preliminaries: Vision-Language Models (VLMs) and CLIP

For the purposes of this paper, when we discuss VLMs, we refer to the paradigm of 1) training deep neural networks using contrastive learning with image-text pairs collected from the Internet and 2) applying them in zero-shot settings to various tasks. We center our investigation around the popular CLIP model (Radford et al., 2021). We review both of these steps below.

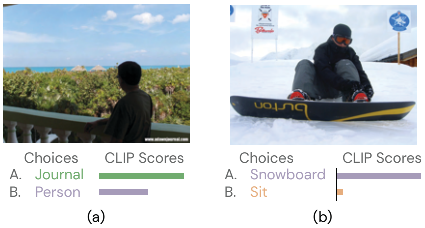

Figure 3: (a) Scene text recognition causing the wrong answer for object recognition. (b) Object recognition causing the wrong answer for action recognition.

### 2.1 Vision-Language Contrastive Pretraining

The goal of vision-language pretraining is to utilize pairs of images and text to learn effective visual representations. Recent progress in this area has been driven by advances in contrastive learning, in large part due to its computational efficacy allowing for easier scalability than exact prediction tasks (Radford et al., 2021). For each batch consisting of $N$ image-text pairs, the images and text are passed through independent vision and language encoders to obtain $N$ embeddings of each modality. The contrastive pretraining task is to identify which images go with which text. This can be thought of as a classification task: for each image in the batch, classify which text option in the batch it belongs to by maximizing the cosine similarity of embeddings of corresponding pairs and minimizing it for every other combination.

### 2.2 Zero-Shot Application

We review the typical zero-shot application pipeline for CLIP. (Note that we describe the steps for selecting a text option matching a given single image; this is symmetric with selecting an image option matching a given piece of text.) First, an image is embedded using the vision encoder. Similarly, the text options for the answer to the task are embedded using the language encoder. The cosine similarity in latent space between the image embedding and all text embeddings is used as the similarity metric. The softmax function is applied to these similarities, like the logit in typical classification, and finally the text option with the highest score is returned as the output. The steps of this pipeline are often taken for granted, but as we will see, they can have unintended consequences.

## 3 The Problem of Task Ambiguity

Training with language enables a zero-shot application procedure that can solve multiple tasks with no change in the method. However, this is a double-edged sword – it means we can obtain the solution to a task we did not intend to solve. The zero-shot application procedure only allows it to pick a single answer, regardless of how many may be "correct" in some sense. To make matters worse, we also do not know *which* of the (potentially many) tasks it is solving for a given decision: there is no mechanism to inform the user which task it is even trying to solve. In Figure 1, a zero-shot model presented with these choices could choose to solve action recognition, scene text recognition, or object recognition, depending on what parts of the input the model pays attention to, with each being valid – and these only represent three of the potentially unbounded set of reasonable tasks that could be defined for a given image.

This leads us to *task ambiguity*, or the inherent impossibility of knowing which task a user wants to solve in a zero-shot language-based framework without further input. It is already clear by construction that this zero-shot application could solve multiple tasks, and would be justified in picking a correct answer to any task.

Task ambiguity motivates the question: given the model is capable of many tasks, which task will it solve for a given image? We find that for given images, the visual representations CLIP produces strongly prefer one task's solution over other correct answers pertaining to the image, as discussed in Section 3.1. We call this tendency *task bias*. Can we characterize when these preferences come into play, and to what extent? We note that while this is a form of data bias – which is well-studied

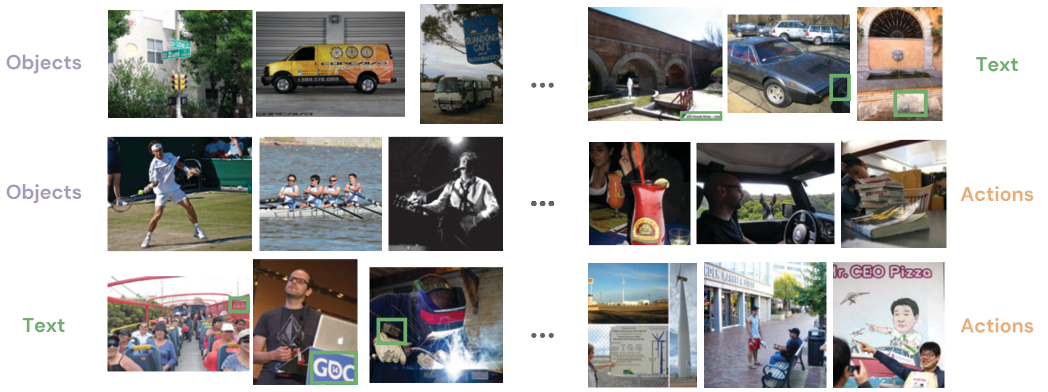

Figure 4: Example images of the most extreme per-image task bias. Best viewed with zoom. The bias is often unintuitive, picking minor text for large images (highlighted by the green boxes), obscure or small objects for clear actions, and more.

– it is a distinct form unique to zero-shot application of VLMs that has thus far not been identified or deeply explored in its own right.

On the surface, this may seem to have a deceptively simple solution: try and make sure you never provide the answers to multiple questions among your options. There are two issues with this approach. First, it does not address the underlying problem with the visual representations; if the model is not paying attention to parts of the input relevant for tasks other than the one it is biased towards, the representation will simply not contain the information needed for the task. Secondly, it is often unavoidable to encounter task ambiguity in realistic settings. Consider Figure 3. The options provided would be reasonable for the desired tasks across a dataset, but cause problems in the given images. Such examples are common when applying these models, thus making this question inevitable. In fact, this even occurs in the categories used for the ImageNet benchmark. Figure 6 shows a case where CLIP picks the ImageNet category “swing” instead of “baseball player,” solving a task other than the intended one.

This bias results in undesired behavior with these models. For example, it gives rise to the “typographic attacks” identified by (Goh et al., 2021), where a model can be tricked into choosing an incorrect category for classification by instead being offered the choice of solving the scene text recognition task as an adversary adds text to an image. We expect a new wave of work on adversarial attacks that aim to influence a VLM’s choices by affecting what task it will solve or exploiting its confusion of linguistic concepts. These can be thought of as *ambiguity attacks*, with “typographic attacks” representing the specific (interesting) case of an ambiguity attack by incentivizing the scene text recognition task over the one the user actually wants.

### 3.1 Evaluation

As task bias is a unique issue arising from zero-shot application of VLMs, we lack established ways of evaluating it. To address this, we develop a new task explicitly designed to probe task bias. Given correct text responses for multiple potential tasks, which will the visual representation from CLIP be closest to? None of the answers is more true than the others; as such, there is no behavior that is ideal. **Just the fact that we can construct this task itself shows a critical flaw in the way we are using VLMs.** Our evaluation lets us understand *how* the issue presents itself rather than whether it is an issue or not. It prompts the development of new application paradigms that allow for task-conditioning in zero-shot application of VLM learned representations.

While there exist many such high level semantic tasks in natural images, we identified the following three tasks as particularly relevant, co-occurring frequently in natural images: object recognition, action/activity recognition, and scene text recognition. Previous work gives us good reason to expect CLIP to perform well on each of these separately zero-shot (Radford et al. (2021); Goh et al. (2021)). We present the first work that examines CLIP’s behavior when presented with all of them together.

To conduct this evaluation, we need a dataset with multiple semantic labels for each image, similar to the choices given in Figure 1. Multitask datasets in vision do exist, such as the Taskonomy dataset

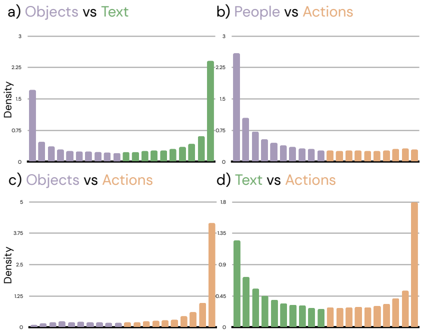

Figure 5: Task bias evaluation. We measure the distance in CLIP latent space between every image and the correct answer to both tasks. Closer to the left end implies closer to the first task listed while the right implies the second. We compute histograms of these (normalized) distances, showing that the most density can be found at both extremes.

(Zamir et al., 2018); however, these datasets have primarily dealt with low-level vision tasks, such as surface normal estimation, that are not as relevant for VLMs, which excel at higher-level semantic tasks. We collected a large dataset of images where the previously mentioned tasks have clear, reliable human labels for every image. We set up our datasets to evaluate pairwise task comparisons. We obtain this data by building on the large, publicly available OpenImages-V6 repository of $\sim$ 9 million images (Kuznetsova et al., 2020; Benenson et al., 2019). We obtain labels for objects by processing the provided detection labels; for actions by processing the provided labels for inter-object relationships and attributes; and for scene text by merging the images with open-source scene text labels created for a previous iteration of OpenImages (Krylov et al., 2021). When considering action recognition, there are typically two “objects” of interest in each image – actual objects as well as the people acting on those objects. Thus, we make our analysis of comparisons involving action recognition more fine-grained by considering people separately from objects at large. Figure 3 shows several example images. Please see Appendix A for further details on dataset construction.

### 3.2 Establishing per-image task bias

We probe every image by first embedding the text corresponding to correct answers for both tasks with labels, then determining which text embedding the image embedding is closer to and by how much. This follows the standard zero-shot classification protocol for VLMs (described in Section 2.2) to do binary classification between the correct answers for both the tasks. CLIP thus assigns a score for each text option, as visualized in Figure 3. Our analysis follows from these outputs.

Our results show that most embeddings are biased towards a task. Figure 5 plots (normalized) histograms to display this trend across the datasets. We observe a distribution with highest density at the ends (max task bias). This suggests that for a given image, the predictions are strongly biased towards one particular task. One may hypothesize this bias may be predictable; for example, if an image is primarily text with a small object in it, it may be understandable to expect to solve scene text recognition. Unfortunately, Figure 4 shows the opposite. We emphasize all of the answers are *correct* – but the preferred task is highly unintuitive and not aligned with what we may expect. The model “sees” minor objects over obvious actions, subtle actions over obvious objects, and so on.

### 3.3 Text Prompt Engineering

The first hope one might have is that we can use our existing apparatus of language prompts to indicate to the model which task we want to solve. The only immediate way the existing structure affords us to specify a task in words is through the text encoder. However, the CLIP text encoder and visual encoder are independent, only sharing an output space. Modifying our text input will only influence the text embeddings. This means our visual embedding will remain unmoved; if the visual encoder has learned to only pay attention to part of an image that solves one task (e.g., image text), the representation will only encode the information relevant to that task (e.g., text recognition), regardless of the input passed through the text encoder.

As text prompting cannot guide our visual representations, it cannot resolve problems of task ambiguity for downstream tasks such as image generation with CLIP embeddings (e.g. Ramesh et al. (2022), Crowson et al. (2022)). However, we may think it could be of interest for our secondary goal: solving the intended recognition task. Say we wanted to solve scene text recognition for Figure 1. Instead of the text choices on their own, could we guide the model by using a task-specific prompt? This would add a clarifying prefix to every option – e.g., "This is a photo of text that reads [CHOICE]." Manual prompting has proven one of the most successful methods in guiding large language models (Liu et al., 2021; Brown et al., 2020), so it seems reasonable to believe that it may help us here. To evaluate this task-directed prompting for CLIP, we created such prefixes for each task (see Appendix B for details). We added these prefixes to try and guide the decision, recomputing the respective scores. We then considered the change in proportion of data points for which the model produces the label for the intended task. We find that manual text prompts produce ambiguous results. For text vs. actions, the performance increases 14.7% for the former, but 7.28% for the latter. For objects vs. text, manual text prompting gives a 9.22% decrease for objects and 11% increase for text. For objects vs. actions, it leads to a 37.5% decrease for objects and a 29.6% increase for actions. Overall, we do not find any consistent way to guide the model towards desired tasks using manual text prompts.

This indicates that adding manual text prompts fails to steer the task consistently and significantly. Further, it suggests that the task bias in the visual representations cannot be trivially overcome, even for the retrieval task, by simply modifying the representations of the "answer choices."

## 4 Resolving Task Ambiguity

### 4.1 Method

Given a particular task of interest, we would like to be able to change the visual representations extracted from images to primarily pertain to input features relevant to that task. We can measure this by considering, over a large set of images, how frequently the representations are closer to the solutions for the goal task than for other tasks.

In this section, we provide a simple but effective technique that allows us to reliably direct CLIP towards representations relevant to a given task without modifying the pre-trained model. To this end, we learn a single set of parameters for each task of interest that we combine with the visual input before it is passed through the visual encoder. This can be thought of as learning a per-task visual prompt that indicates what question we are asking CLIP, guiding the information encoded in its representation.

We are given a dataset where each input image $x$ has task labels $y_1, y_2 \ldots y_n$. For instance, in Fig. 1a), we might have $y_1$ = "lift", $y_2$ = "football", $y_3$ = "weight", where task 1 is action recognition, task 2 is scene text recognition, and task 3 is object recognition. Given a fixed pre-trained model parametrized by $\theta$, the goal is to learn a set of parameters $\phi$ that minimizes the objective

$$\max_{\phi} \sum_{y=1}^{n} 1[y = y_k] \log \frac{e^{f_V(g_\phi(x)) \cdot f_T(y)}}{\sum_{k'=1}^{n} e^{f_V(g_\phi(x)) \cdot f_T(y'_k)}} \tag{1}$$

where $k$ is the task of interest for building our visual representations, $f_\phi$ is a function that applies the prompt parameters $\phi$ to the input image, and $f_V$ and $f_T$ are the pre-trained vision and text encoders of CLIP respectively. In other words, this is the cross-entropy of the CLIP similarity between the image and the answer to the desired task compared to every other text option. This encourages higher likelihood for the solution to task $k$ compared to those for other tasks, thereby learning a prompt that leads to a representation better suited to that task.

Several prior works explore learning prompts for visual models. However, these works deal with prompting as a method to improve the performance of pre-trained models on a particular downstream dataset, such as ImageNet (Bahng et al., 2022; Salman et al., 2021; Jia et al., 2022b). Note that in our method, unlike previous work, the label set $y_1, \ldots, y_n$ used to compute the objective differs for each instance in our batch; what remains constant is the set of tasks labeled for each associated image.

We explore two main approaches to implement the prompt application function $f_\phi$. First, we aim to learn a prompt in pixel space, similar to Bahng et al. (2022); in this case, the learned parameters are simply added to a fixed border around the original image prior to tokenizing and image and passing it through the visual encoder. Secondly, we explore learning a prompt as a visual token, similarly to

| | | **Task** | Objects vs. Text | | Objects vs. Actions | | Actions vs. Text | |
|---|---|---|---|---|---|---|---|---|
| | | Direction: | Objects | Text | Objects | Actions | Text | Actions |
| **Method** | No Prompt | | 50.28 | 49.72 | 48.68 | 51.32 | 43.70 | 56.30 |
| | VP (PS=1) | | 92.04 | 82.94 | 67.55 | 56.25 | 80.87 | 98.04 |
| | VP (PS=5) | | 97.67 | 95.2 | 81.13 | 96.15 | 97.59 | 98.04 |
| | ViTP | | 99.46 | 99.15 | 100.00 | 100.00 | 100.00 | 100.00 |

Table 1: Accuracy for resolving task ambiguity on the test splits of each of our datasets using various prompting methods. Our results show that with just a few parameters (and, critically, without changing model parameters) both approaches to visual prompting successfully guide CLIP toward consistently solving one task over other.

Jia et al. (2022a), where the prompt parameters are prepended to the encoded visual tokens before they are passed through the vision transformer.

### 4.2 Results on Task Disambiguation

To evaluate the effectiveness of visual prompting for resolving task ambiguity, we construct random held-out splits from the dataset introduced above. (See Appendix A for details.) In contrast to previous sections, where we use task-specific prefixes in text label categories to allow the model to have the best chance for identifying the task associated with the word, we equalize the text prefixes for all experiments in this section. Every single option in the text retrieval set is prepended with “This is a photo of a” as is standard with CLIP usage. This prevents any visual prompt we learn from being optimized to simply pick the correct task-directed prompt prefix, instead of learning the task-level representation.

Our results on the held-out dataset can be found in Table 1. For each task and method, we report the total percentage of data points for which the model produces the intended task label. **No Prompt** indicates the values for the unmodified zero-shot CLIP encoder. In this row, the numbers across the two tasks for each dataset must add up to $100$, as the zero-shot model will choose either of the two tasks in the dataset. In subsequent rows, which show the task-optimized results, a successful task prompted model should have higher performance for the direction in which it was prompted. Note that the numbers in all rows except the first will not add up to 100, because separate models have been trained for each task inside a dataset, and the number shown is the prompted models preference for the task it was optimized to perform. **VP** indicates visual prompting as implemented by Bahng et al. (2022). Specifically, we use their edge prompting method, where prompt parameters replace the edges of the input image like padding, with the prompt size (PS) controlling the number of padding layers on each side. As such, the prompted model $PS = 5$ has more tuned parameters than the model with $PS = 1$. **ViTP** shows the results for vision-side prompting implemented as the typical method of prompting the inputs of transformer-style models, where parameters are added between the learned [CLS] token and input representation. We add a single embedding with the size of the model width.

We train each prompt for a *single* epoch, as we find we can obtain satisfactory performance without further training. Our results indicate that by optimizing only a few parameters that are shared across all the instances in our data, it is possible to direct the model to solve our intended problem. While optimizing even a one-pixel border of parameters in pixel space on the edges of the input image can provide a significant boost in directing the model to solve the task, we find that the token-based prompting yielded the best results.

### 4.3 Effects on Downstream Task Performance

We now consider how adding these task-directed prompts affects downstream performance on the associated task itself, rather than probing in comparison with other task solutions. If we have indeed led to visual representations better suited to a task, we should observe better performance on said task. We use the OpenImages object recognition categories to construct an object recognition task on our held-out dataset (see Appendix C for details); we embed all text options and evaluate with the standard zero-shot recognition procedure, choosing the category with the text embedding closest to the visual embedding in the CLIP latent space. We then compare performance, measured by classification accuracy, for this object recognition task with each of our learned prompts compared to baseline CLIP. Our results are summarized in Table 2. We find that prompting towards objects improves classification performance, substantially in the case of pixel-space prompting with a border size of 1. On first glance, it is surprising that a border size of 5 achieves a lower accuracy than a border size of 1, given that the latter achieves somewhat better performance at task disambiguation. However, the increased border size has the potential to pull the input image substantially further out of distribution than the border size of 1. We hypothesize this distribution mismatch tempers the improvement derived from a more task-focused representation.

### 4.4 Prompting and Self-Attention

Our goal is to modify the visual representation to direct it to capture the specific features in the image relevant to the intended task. In a transformer's encoder, this means that the token corresponding to the image embedding should pay more attention to the visual tokens of the patches most relevant for the task. Even though prompting is a light-weight, quick-to-tune modification to the input encoder, it has the ability to indirectly control the attention distribution for the rest of the image. This is because the image embedding will now also attend to the learnt prompt, changing the scale of attention to the rest of the image. We highlight the difference in attention between prompting for object recognition and prompting for action recognition for an example from the ImageNet dataset, which is out of distribution for our trained prompts, in Fig. 6. We see that the prompting changes the attention to put more weight on input features more relevant to the task associated with the prompt.

| | OpenImages Acc. |
|---|---|
| Unprompted | 20.467 |
| VP (PS = 1) | 37.254 |
| VP (PS = 5) | 20.799 |
| ViTP | 27.547 |

Table 2: Object Classification

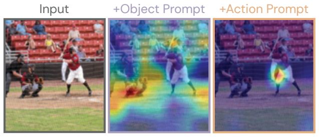

Figure 6: An example of task ambiguity from the ImageNet dataset. CLIP chooses the ImageNet category "swing", as its representation of the image is biased towards actions. By applying our prompts, we can redirect the attention to the features relevant for that task.

To test the impact of prompting on visual attention for a broader set of out-of-distribution images, we take images from Goh et al. (2021) and the dataset collected by Ilharco et al. (2022), both of which contain pictures of objects labelled with sticky notes of unrelated text. For each image, we compute two task-directed aggregated attention maps using the models that we prompt toward the respective tasks. Our method for calculating the attention maps is based on Abnar & Zuidema (2020); Gildenblat (2021). In order to isolate the features that are especially task relevant, we visualize the difference in the task-directed attention maps in the direction of the desired task. Our results can be seen in Figure 7. The attention maps in the text direction clearly show a focus on the sticky notes; those extracted from the model prompted towards the objects reveal a more object-centric focus.

## 5 Related Work

**Vision-Language Models (and Applications):** Internet pre-training with vision-and-language has become an established paradigm with the release of CLIP (Radford et al., 2021), and larger, stronger models have since followed (Jia et al., 2021a; Yuan et al., 2021; Alayrac et al., 2022; Singh et al., 2021). While designed by construction for retrieval only, the generality of the visual and text representations of CLIP has seen them find applications in video retrieval (Luo et al., 2021), robotic manipulation and navigation (Shridhar et al., 2021; Gadre et al., 2022; Khandelwal et al., 2021), OOD image detection (Mukhoti et al., 2020),and image generation, to steer generative models (Crowson et al., 2022; Gal et al., 2021), and much more, where parts of guiding signal for training are often based solely on the visual representation. Our work considers an inherent bias that these visual representations have toward certain tasks, and proposes a method to correct them. Having task bias-free visual representations is important not only for zero shot application of CLIP in retrieval settings but also for the described downstream tasks, where a representation biased toward a particular task may

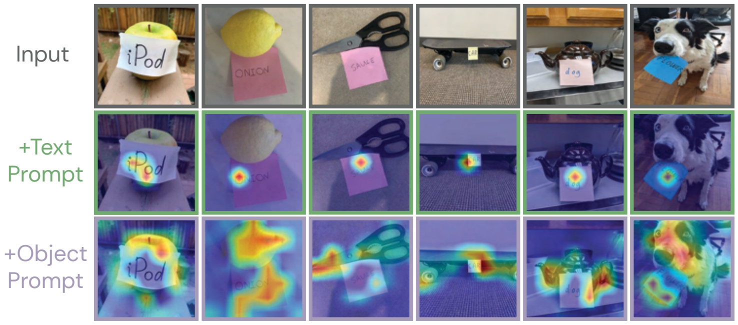

Figure 7: Difference between attention maps with scene text recognition and object recognition prompting on in-the-wild images from an out-of-distribution dataset Ilharco et al. (2022). Best viewed with zoom. The first image is the well-known “iPod vs Apple” from Goh et al. (2021). We see the text recognition prompt results in more attention on text and the object recognition prompt results in more attention on surrounding objects.

lead to incorrect targets for robotic systems, misguided image generations or faulty re-identifications when used as part of safety critical applications.

**Investigating CLIP’s representations:** While internet pretrained representations have seen impressive performances on several benchmarks, they are still susceptible to the most common problems in machine learning, especially deep learning in computer vision, with issues in interpretability, representation of uncertainty, social bias and robustness (Rudin et al., 2021; Nado et al., 2021). (Goh et al., 2021) studies some of these issues in great detail for CLIP and introduces the notion of a *typographic attack*, which shows that CLIP models are susceptible to “reading” instead of object detection. Subsequent works propose solutions for this specific problem: Materzynska et al. (2022) consider object and text representations as normal directions in CLIP’s visual representation space and learn projections on top of CLIP representations with an orthogonality constraint to isolate CLIP’s reading ability. Ilharco et al. (2022) introduce a method to improve accuracy on finetuned tasks while preserving zero shot performance on other tasks- they explore their method in the *typographic attack* setting and show that it is possible to steer CLIP away from typographic attacks with fine-tuning the entire model. Both Ilharco et al. (2022) and Materzynska et al. (2022) use synthetic data for adapting their models, which cannot be generated outside the specific object / scene text setting. Instead, we identify typographic attacks as a specific instantiation of the broader pose a method that can alter the visual representations for specific tasks by tuning only a few parameters.

**Learned Prompting** is a lightweight alternative to full fine-tuning used to quickly adapt pre-trained models to application domain data distributions. It has previously been used extensively in the natural language community (He et al., 2022; Schick & Schütze, 2021) with transformers. Learned prompts are essentially a small set of parameters that are optimized for a downstream task, and can be added anywhere inside a trained model. Our work builds on Bahng et al. (2022) who show that adding prompt parameters directly to the input image is an effective alternative to linear probing and fine-tuning for CLIP, with possible improvements to robustness. Our work also builds on Jia et al. (2022a) who show that adding prompts before the first self-attention layer but after encoding the inputs is also effective for vision transformers, as it is in NLP literature. Both these papers explore using vision side prompting for adapting pre-trained models to new downstream datasets; instead, we explore prompting for directing the model to perform specific tasks over others.

## 6 Conclusions

We explore the new problem of task bias in reference to the CLIP model for vision-language similarity, finding the representation for a given image is strongly biased towards the text solution for a particular task a priori. To this end, we introduce a dataset illustrating these issues, suitable for the community to further analyze the new phenomenon. Finally, we show effective methods for guiding representations towards a task of interest via visual prompting, and show that this substantially improves downstream recognition performance.

## 7 Ethics Statement

This work furthers our understanding of the inner workings of VLMs, and in particular their shortcomings. Bad actors equipped with better understanding of a model can better break it. In particular, we believe that potential adversarial attacks could be made exploiting task ambiguity. Our results may lead to the development of such attacks; however, we hope that they will also lead to work preventing them, as our exploration of visual prompting for task guidance could pose a strong defense.

## 8 Reproducibility Statement

We will release all models and data. We work principally with CLIP which has been open-sourced. We provide complete details for our dataset construction in Appendix A. We will release all code, including that associated with modifying the open source implementation of Radford et al. (2021) and Bahng et al. (2022) that we used. Any hyperparameters not mentioned in the paper are taken directly from Bahng et al. (2022).

## References

Samira Abnar and Willem Zuidema. Quantifying Attention Flow in Transformers. *arXiv:2005.00928 [cs]*, May 2020. URL http://arxiv.org/abs/2005.00928. arXiv: 2005.00928.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. URL https://arxiv.org/abs/2204.14198.

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring Visual Prompts for Adapting Large-Scale Models, June 2022. URL http://arxiv.org/abs/2203.17274. Number: arXiv:2203.17274 arXiv:2203.17274 [cs].

Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. *arXiv:1903.10830 [cs]*, April 2019. URL http://arxiv.org/abs/1903.10830. arXiv: 1903.10830.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*, July 2020. URL http://arxiv.org/abs/2005.14165. arXiv: 2005.14165.

Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance, 2022. URL https://arxiv.org/abs/2204.08583.

Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Clip on wheels: Zero-shot object navigation as object localization and exploration, 2022. URL https://arxiv.org/abs/2203.10421.

Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators. *arXiv:2108.00946 [cs]*, August 2021. URL http://arxiv.org/abs/2108.00946. arXiv: 2108.00946.

Jacob Gildenblat. Explainability for Vision Transformers (in PyTorch), November 2021. URL https://github.com/jacobgil/vit-explain. original-date: 2020-12-29T11:27:52Z.

Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal Neurons in Artificial Neural Networks. *Distill*, 6(3): e30, March 2021. ISSN 2476-0757. doi: 10.23915/distill.00030. URL https://distill.pub/2021/multimodal-neurons.

Huy Ha and Shuran Song. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models, 2022. URL https://arxiv.org/abs/2207.11514.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. URL http://arxiv.org/abs/1512.03385. arXiv: 1512.03385.

Yun He, Huaixiu Steven Zheng, Yi Tay, Jai Gupta, Yu Du, Vamsi Aribandi, Zhe Zhao, YaGuang Li, Zhao Chen, Donald Metzler, Heng-Tze Cheng, and Ed H. Chi. Hyperprompt: Prompt-based task-conditioning of transformers, 2022. URL https://arxiv.org/abs/2203.00759.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights, 2022. URL https://arxiv.org/abs/2208.05592.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *CoRR*, abs/2102.05918, 2021a. URL https://arxiv.org/abs/2102.05918.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. *arXiv:2102.05918 [cs]*, June 2021b. URL http://arxiv.org/abs/2102.05918. arXiv: 2102.05918.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning, 2022a. URL https://arxiv.org/abs/2203.12119.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual Prompt Tuning, July 2022b. URL http://arxiv.org/abs/2203.12119. Number: arXiv:2203.12119 arXiv:2203.12119 [cs].

Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: CLIP embeddings for embodied AI. *CoRR*, abs/2111.09888, 2021. URL https://arxiv.org/abs/2111.09888.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv: 1412.6980.

Ilya Krylov, Sergei Nosov, and Vladislav Sovrasov. Open Images V5 Text Annotation and Yet Another Mask Text Spotter. *arXiv:2106.12326 [cs]*, June 2021. URL http://arxiv.org/abs/2106.12326. arXiv: 2106.12326.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, July 2020. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-020-01316-z. URL http://arxiv.org/abs/1811.00982. arXiv: 1811.00982.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *arXiv:2107.13586 [cs]*, July 2021. URL http://arxiv.org/abs/2107.13586. arXiv: 2107.13586.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An Empirical Study of CLIP for End to End Video Clip Retrieval. *ArXiv*, 2021.

Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings, 2022. URL `https://arxiv.org/abs/2206.12403`.

Joanna Materzynska, Antonio Torralba, and David Bau. Disentangling visual and written concepts in clip, 2022. URL `https://arxiv.org/abs/2206.07835`.

B. E. Moore and J. J. Corso. Fiftyone. *GitHub. Note: https://github.com/voxel51/fiftyone*, 2020.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet K. Dokania. Calibrating Deep Neural Networks using Focal Loss. *arXiv:2002.09437 [cs, stat]*, October 2020. URL `http://arxiv.org/abs/2002.09437`. arXiv: 2002.09437.

Zachary Nado, Neil Band, Mark Collier, Josip Djolonga, Michael W. Dusenberry, Sebastian Farquhar, Angelos Filos, Marton Havasi, Rodolphe Jenatton, Ghassen Jerfel, Jeremiah Liu, Zelda Mariet, Jeremy Nixon, Shreyas Padhy, Jie Ren, Tim G. J. Rudner, Yeming Wen, Florian Wenzel, Kevin Murphy, D. Sculley, Balaji Lakshminarayanan, Jasper Snoek, Yarin Gal, and Dustin Tran. Uncertainty Baselines: Benchmarks for Uncertainty & Robustness in Deep Learning. *arXiv:2106.04015 [cs]*, June 2021. URL `http://arxiv.org/abs/2106.04015`. arXiv: 2106.04015.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *arXiv:2103.00020 [cs]*, February 2021. URL `http://arxiv.org/abs/2103.00020`. arXiv: 2103.00020.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. URL `https://arxiv.org/abs/2204.06125`.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges. *arXiv:2103.11251 [cs, stat]*, July 2021. URL `http://arxiv.org/abs/2103.11251`. arXiv: 2103.11251.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575 [cs]*, January 2015. URL `http://arxiv.org/abs/1409.0575`. arXiv: 1409.0575.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Sai Vemprala, A. Madry, and Ashish Kapoor. Unadversarial Examples: Designing Objects for Robust Vision. In *NeurIPS*, 2021.

Timo Schick and Hinrich Schütze. True few-shot learning with prompts - A real-world perspective. *CoRR*, abs/2111.13440, 2021. URL `https://arxiv.org/abs/2111.13440`.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. CLIPort: What and Where Pathways for Robotic Manipulation. *arXiv:2109.12098 [cs]*, September 2021. URL `http://arxiv.org/abs/2109.12098`. arXiv: 2109.12098.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. FLAVA: A foundational language and vision alignment model. *CoRR*, abs/2112.04482, 2021. URL `https://arxiv.org/abs/2112.04482`.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *CoRR*, abs/2111.11432, 2021. URL `https://arxiv.org/abs/2111.11432`.

Amir Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling Task Transfer Learning. *arXiv:1804.08328 [cs]*, April 2018. URL `http://arxiv.org/abs/1804.08328`. arXiv: 1804.08328.

# A Task Ambiguity - Dataset Construction

We constructed the datasets for the task ambiguity experiments by combining a subset of the OpenImages-V6 dataset, which contains information about objects, attributes and relationships with other objects, (Kuznetsova et al., 2020; Benenson et al., 2019) with independently annotated scene text labels associated with a subset of the OpenImages-V5 dataset (Krylov et al., 2021). In order to piece together this dataset, we installed the dataset using the recommended FiftyOne (Moore & Corso, 2020) library and combined images from all of the splits in their OpenImages installation. We reported results on four different comparisons: Object v. Scene Text, People v. Actions, Objects v. Actions and Scene Text v. Actions respectively. Below, we outline the pairwise dataset creation process for each one of these.

**Objects v. Scene Text**: We consider the intersection of images with scene text labels and images with object detection labels. Images often contain more than one object, but the goal of image classification for object recognition is to identify the most significant one, constraining the task to one label. Similarly to the ImageNet dataset Russakovsky et al. (2015), we want to consider the most salient object in the scene for this label. For every object with detection labels in an image, we calculate the area of the associated bounding box and choose the object label with the maximum area for a given image ID as the label for the object recognition task. (We consider detection labels instead of categorization labels for this reason; categorization labels do not give us any sense of how significant a given label is.) We replace all gendered object labels with the generic human label "Person". This leaves us with a set of 175335 images with both object and scene text labels.

**People v. Actions**: For this comparison, we consider all images with action annotations in OpenImages. (All such images must contain people by definition of the action recognition task, so we do not need to consider an intersection.) As of V6, OpenImages does have human action annotations (Kuznetsova et al., 2020); unfortunately, there is no field for 'actions' in the dataset.

Instead, actions are distributed through a couple of different fields. Images that contain actions form a subset of those labeled with 'relationships' in OpenImages. Every relationship instance is defined as a 3-tuple of the form (first label, relationship label, second label). The relationship label is a general term for the word that connects the first and second labels and can take various forms. For example, if 'is' is the relationship label, the second label can be an attribute of the first label. Alternatively, the relationship label could be an unconjugated verb, in which case the second label is another object. For all the relationships available, we first filter the entire dataset to only include those where the first label refers to a person-related class (In OpenImages, these are 'Boy', 'Girl', 'Man', 'Woman' and 'Person') to ensure action recognition is a valid task on the datapoints. We then isolate those pairs where the relationship label directly defines an action (eg. 'read', 'dance') and those where the relationship label is generic (eg. 'is') followed by a verb-like attribute (eg. 'Cry', 'Jump'). In the former case, we take the relationship label as the action label and in the latter case, we use the second label as the action. We conjugate all verbs in their present continuous form. The process leaves us with the first label (which is guaranteed to be a person) and an associated action label for a set of images. We then remove duplicates. We sample from each action to correct for label imbalance. Our final dataset contains 89626 distinct images.

**Objects v. Actions**: We obtain a set of images containing actions per the previous section. Among these, we consider images that have object detection labels to obtain images where people interact with another object through the action. We further discard any cases where the second label in the relationship 3-tuple is also a person, leaving us with 8611 images with paired action and inanimate object labels.

**Scene Text v. Actions**: We again consider the subset of images with valid action labels, this time taking the subset of images that also contain scene text. Our final dataset contains 56027 labelled images with both pairs.

When datasets are used for training prompts and evaluation, $90\%$ of the data is used for training and the remaining $10\%$ is used as a held out set.

| **Task** | **Prefix** |
|---|---|
| Scene Text | This is a photo of text which reads |
| Actions | This is a photo of someone who is |
| Objects, People | This is a photo of <article> |

Table 3: Tasks we investigated and the associated prefixes we attach to form our prompts.

## B Task Ambiguity - Task-Directed Prompting Details

This section provides details for the experiment from Section 3.3. The goal of the experiment is to determine whether a text prompt, added as a prefix to the text choices, can guide the zero-shot classification procedure to solve a desired task. In Table 3, we list the prompts used and the associated intended task. We arrived at these prompts from the original prompts shown to lead to an improvement in baseline performance in CLIP (Radford et al., 2021), reused by ALIGN (Jia et al., 2021b) and further models. The primary consideration for designing these prompts was that they must provide sufficient information for a human to understand which task is intended. We experimented with various prompts fulfilling this criterion, choosing the best among them. (This makes it especially surprising that for some experiments, the clarifying additional text information actually results in substantially *worse* performance.)

## C Downstream Task Evaluation - Details

For the results in Table 2, we use the model prompted toward the object task on the object vs. scene text dataset. All numbers are reported on our held-out set of this dataset containing 17000 images.

## D Task Ambiguity - Task Clarification Baselines

A preliminary question to whether we can resolve task bias is whether we can detect the direction of the task bias in the visual representation solely from the image embedding. In this section, we investigate the effective of using the full attention mask and the input image toward guessing the task bias in an input. We provide preliminary baselines for the difficult task of predicting which task a zero-shot model is solving for the Objects v. Scene Text and Actions v. Scene Text pairs.

**Dataset**: The results from our task ambiguity experiments give us the per label task preference for every image in our dataset. We use these pseudo-labels as indicators of CLIP's task bias on the particular image considered. Therefore, given an image that is part of a pairwise dataset as an input, we repurpose the index of CLIP's preferred task for that image as a label for training the classifier, ensuring that the final test set that we report results on is near-balanced.

**Methods and Results**: We train four different kinds of classifiers for the two paired datasets. These classifiers differ in their architecture and input space, but in each case the set of labels remains the same. Our results are summarized in 4 and 5, reporting the accuracy of the best trained classifier on the test set.

*Frequent* refers to the typical baseline in binary classification that always predicts the label which occurs more frequently inside the test set.

*Image* refers to using the 3-channel input RGB image directly as the input for the classifier. The model used is a ResNet-18 (He et al., 2015) without pre-training.

*Image+Attention Overlay* refers to overlaying the scaled self-attention map from CLIP's image encoder on the image and pre-processing this as a new image. We use the method from Abnar & Zuidema (2020); Gildenblat (2021), termed 'attention rollout', to calculate the self attention maps, modifying it for CLIP's architecture. We also experiment with other forms of self-attention maps, including using final layer only, and observe similar results for those. Once we have the attention map, we normalize it with its maximum value, scale it to the image shape, and colorize it using

| **Experiment** | **Test Accuracy (%)** |
|---|---|
| *Frequent* | 54.9 |
| *Image* | 61.4 |
| *Image+Attention* | 62.6 |
| *Embedding* | **71.7** |
| *Embedding+Image+Attention* | 71.4 |

Table 4: Results for various classifiers on Objects v. Scene Text task bias clarification task

| **Experiment** | **Test Accuracy (%)** |
|---|---|
| *Frequent* | 53.9 |
| *Image* | 59.9 |
| *Image+Attention* | 61.6 |
| *Embedding* | 72.0 |
| *Embedding+Image+Attention* | **72.8** |

Table 5: Results for various classifiers on Actions v. Scene Text task bias clarification task

OpenCV's JET colormap (Bradski, 2000) to turn it into an RGB image. Finally, we add this to the RGB image and use it as an input. The model used is also a ResNet-18 without pre-training.

*Embedding* refers to classifying directly on top of CLIP's 512-dimensional representation of the image. Not that this is a function of the self-attention and the input image, given the structure of transformers. The model used is a shallow 4-layer MLP, with layers sizes [256, 128, 64, 2] respectively.

*Embedding+Image+Attention* refers to classifying from both the input images and the embedding. Note that we do not necessarily expect this to work better as it gives redundant information to the classifier but in different forms. This is because the embedding itself is a function of the self-attention and the root image. For the model, we use a ResNet18 backbone with a single linear layer to produce a 256 dimensional representation. We further use a single linear layer to constrain the CLIP's image embedding to 256 dimensions. These are then fused and passed through the MLP used for *Embedding* only.

All models are trained end-to-end with the Adam optimizer (Kingma & Ba, 2017), with learning rate 0.0001 and other hyperparameters set to their default values.