More Context, Less Distraction: Improving Zero-Shot Inference of CLIP by Inferring and Describing Spurious Features

Bang An [*1] **Sicheng Zhu** [*1] **Michael-Andrei Panaitescu-Liess** [1]
Chaithanya Kumar Mummadi [2] **Furong Huang** [1]

Abstract

CLIP, as a foundational vision language model, is widely used in zero-shot image classification due to its ability to understand various visual concepts and natural language descriptions. However, how to fully utilize CLIP's unprecedented human-like understanding capabilities to achieve better zero-shot classification is still an open question. This paper draws inspiration from the human visual perception process, where a modern view is that when classifying an image of an object, humans will first infer its class-independent attributes such as background, orientation, and illumination, and then classify based on them. Similarly, we observe that providing CLIP with the object attributes improves classification, and that CLIP itself can reasonably infer the attributes from an image. Based on these, we propose `PerceptionCLIP`, a training-free zero-shot inference method. Given an image, it first infers the object attributes, and then does classification conditioning on them. Experiments show that `PerceptionCLIP` achieves better generalization, less dependence on spurious features, and better interpretability. For example, `PerceptionCLIP` improves average accuracy by 3.3% and worst-group accuracy by 24.8% on the Waterbirds dataset.

1. Introduction

The CLIP model (Contrastive Language-Image Pretraining, Radford et al. (2021)) is a foundational Visual Language Model (VLM) that bridges the gap between the fields of vision and natural language. By pretraining on 400 million image-caption pairs, CLIP can connect various visual concepts with their corresponding natural language descriptions, making it the foundation for numerous other visual language models (Zhu et al., 2023; Liu et al., 2023; Dai et al., 2023; Li et al., 2023b), diffusion models (Ramesh et al., 2022; Rombach et al., 2022), and semantic segmentation models (Kirillov et al., 2023). Such remarkable understanding capability of CLIP has an important application known as zero-shot inference (Larochelle et al., 2008) — open-ended image classification through natural language without access to a validation set. It enables many challenging tasks that suffer from little to no downstream data, such as model deployment in the wild (Li et al., 2023a), medical image classification (Wang et al., 2022) and satellite object recognition (Ramaswamy et al., 2023).

While CLIP exhibits strong potential for zero-shot inference, the corresponding methodology has not been systematically investigated, leading to sub-optimal generalization, reliance on spurious features (Yang et al., 2023), and lack of interpretability (Menon & Vondrick, 2022). Current methods treat the image classification as a text retrieval task, but lack systematic investigation into the text prompts used. For example, a basic method (Radford et al., 2021) uses a simple template "*a photo of a {class name}*" to find the most relevant class name, which however differs from the captions provided by annotators in the pretraining data. Another method uses an ad-hoc selection of 80 templates for ensemble, achieving better generalization, but it remains unclear whether these templates are optimal or why they are effective. These ad hoc zero-shot inference methods may risk squandering CLIP's understanding of both class-dependent visual concepts and class-independent attributes such as orientation and lighting (Figure 1).

Given the unprecedented human-like image and language understanding of CLIP, a natural idea is to draw inspiration from human visual perception for developing zero-shot inference methods. Indeed, the classic neuroscience textbook Kandel et al. (2013) offers a modern view of human visual perception, presenting a significant difference from current zero-shot inference methods:

[*]Equal contribution [1]University of Maryland, College Park [2]Bosch Center for Artificial Intelligence. Correspondence to: Bang An <bangan@umd.edu>, Sicheng Zhu <sczhu@umd.edu>.

Work presented at the ES-FoMo Workshop at ICML 2023, Honolulu, Hawaii, USA. PMLR 202, 2023.

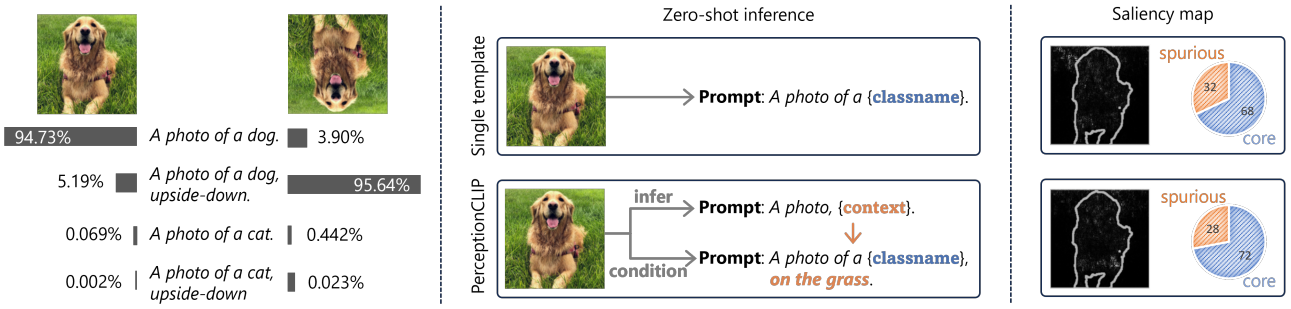

Figure 1. **(Left)**: CLIP understands natural language descriptions of object attributes (here orientation). **(Center)**: Compared to the basic zero-shot classification method, which uses a fixed template for class name retrieval, our method first infers object attributes (here the background), and then infers the class conditioning on the inferred attributes. **(Right)**: The corresponding saliency maps obtained by computing the gradient with respect to the classification loss showcase that our method focuses more on the core features (here the dog), and is less distracted by spurious features (here the background).

> *"The brain analyzes a visual scene at three levels: low, intermediate, and high. At the lowest level, visual attributes such as local contrast, orientation, color, and movement are discriminated. The intermediate level involves analysis of the layout of scenes and of surface properties, parsing the visual image into surfaces and global contours, and distinguishing foreground from background. The highest level involves object recognition. Once a scene has been parsed by the brain and objects recognized, the objects can be matched with memories of shapes and their associated meanings."*
>
> *"... the perceptual interpretation we make of any visual object depends not just on the properties of the stimulus but also on its context, on other features in the visual field."*

This perception process is hierarchical, cascaded, and context-dependent, distinguishing it from the current single-level zero-shot inference methods that overlook object attributes. An example that reflects this view is that when humans classify an object in an image, we almost always know its additional class-independent attributes such as orientation and illumination, as these pieces of information are byproducts of the perception process. Moreover, when presented with a rotated image, humans first infer that the image is rotated and then calibrate the classification accordingly.

In this work, we introduce `PerceptionCLIP`, a zero-shot inference method that emulates human visual perception by iteratively inferring and adjusting for the contextual features, resulting in improved generalization, reduced reliance on spurious features, and better interpretability. Our contributions are as follows:

Contributions. We formulate the contextual attributes of objects as generative factors in the data generation process, accompanied by textual descriptions understandable to CLIP (§4.1). We also use the CLIP similarity score to approximate the probability distributions needed for inference (§4.2). In doing so, we showcase that providing ground-truth object attributes helps inference for CLIP (§5.1), and that CLIP itself can infer object attributes reasonably (§5.2).

Based on the observations, we propose `PerceptionCLIP` for zero-shot inference (§6), which automatically infers generative factors and does class inference conditioned on them. Prompt ensemble can be viewed as a special (yet sub-optimal) case of PerceptionCLIP, thus explaining its effectiveness. We empirically evaluate `PerceptionCLIP` (§7) and show that it achieves better zero-shot generalization, interpretability, group robustness, and less reliance on spurious features.

2. Related Work

Descriptive prompts with external knowledge. Due to CLIP's ability to understand visual concepts at a finer granularity than just classes, such as body parts and components, some work leverages external knowledge to expand the visual concepts associated with class names and incorporates their descriptions into prompts to improve zero-shot classification. For example, Menon & Vondrick (2022); Pratt et al. (2022); Mao et al. (2022) use large language models (LLMs) such as GPT-3 to generate class-specific descriptions for each class and integrate them into prompts, such as "a photo of a hen, which has two legs". Novack et al. (2023) use class hierarchies (existing or by querying GPT-3) to generate sub-classes for each parent class and aggregate model predictions on all sub-classes to get a final prediction. In contrast, our method addresses class-independent attributes such as background and orientation, whose comprehension by CLIP is not well-known. These attributes are also combinatorial in nature, covering more aspects of

an image than just a few class-exclusive components and reducing distractions from spurious features.

Additionally, Roth et al. (2023) show that replacing the class-specific descriptions in the prior work with random words or even meaningless characters yields minimal impact on performance, resembling the effect of noise augmentation or randomized smoothing. Addressing this issue, we ablate our method and show that random attributes or meaningless characters yield approximately half the benefit compared to using correct or self-inferred attributes, thus cannot explain the effectiveness of our method. Roth et al. (2023) also show that appending high-level class-independent descriptions (e.g., Land Use for EuroSAT, Place for Places365) to prompts helps classification, which aligns with our findings.

Prompt tuning. Another line of work that modifies prompts to improve CLIP's classification is prompt tuning, which optimizes the prefix characters of the prompts. Typical prompt tuning methods require labeled (Zhou et al., 2022b;a; Zhu et al., 2022; Derakhshani et al., 2023) or unlabeled downstream data (Huang et al., 2022; Mirza et al., 2023; Menghini et al., 2023), making them fall outside our scope of zero-shot (data-free) classification. They are also prone to overfitting the training dataset, whereas our method relies on image attributes shared by common datasets. On the other hand, Shu et al. (2022) use test-time prompt tuning that applies to zero-shot classification. Specifically, they generate multiple views for each test image and optimize the prompt to minimize the entropy of the model's prediction on these views. This method introduces several hyperparameters (e.g., data augmentations, confidence threshold, optimization algorithm, learning rate) that require tuning on a labeled proxy validation set. In contrast, our method, depending on implementation, introduces either no additional hyperparameters or only one (temperature). Furthermore, our method is training-free and can work in the black-box setting.

Reasoning and chain-of-thoughts. The inference process of our method resembles the reasoning or chain-of-thoughts in prompting LLMs (Wei et al., 2022; Yao et al., 2023), where the model is prompted to give some intermediate step results and then conditioning on them to give final results. However, CLIP itself cannot do step-wise reasoning out of the box, so our method manually prompts it through the reasoning process.

3. Preliminaries

This section reviews the original method for the zero-shot inference of CLIP. We also review the captions in the pretraining data of CLIP to show its misalignment with the description templates used for the zero-shot inference.

Notations. We use uppercase letters to denote random variables, while the corresponding lowercase letters denote their realizations. For a random variable Z, we use $p_Z(z)$ to denote its probability mass or density function. For notation simplicity, we omit the subscript Z when the function's meaning can be inferred from the input notation z.

Captions in the pretraining data. CLIP is pretrained on a dataset of 400 million image-text pairs collected from the internet. For each image, the text caption typically describes the visual object in the image, including the object's class and some of its attributes such as color, style, and background (Radford et al., 2021). These captions are typically given by human annotators. For reference, we show some caption examples in Table 1, which are chosen from a comparable dataset LAION-400M (Schuhmann et al., 2021) since the original pretraining dataset of CLIP is not made public.

Table 1. Image caption examples from LAION-400M (comparable to CLIP's pretraining dataset).

Caption #1	*Men's Classics Round Bracelets Watch in Grey*
Caption #2	*stock photo of gremlins - 3 d cartoon cute green gremlin monster - JPG*
Caption #3	*Medium Size of Chair: Fabulous Mid Century Modern Chair Adalyn Accent In Red:*

Zero-shot inference of CLIP. The original CLIP work (Radford et al., 2021) uses the following method for zero-shot visual classification. First, they manually design a (prompt) template, represented by an annotation function $\alpha(y) =$ *"a photo of a {classname of y}"*, that takes the class index y as the input and outputs a text description. Then, we view the CLIP model as a score function $\texttt{CLIP}_1 : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$ via

$$\texttt{CLIP}_1(y; x) \triangleq \langle \phi_I(x), \quad \phi_T(\alpha(y)) \rangle, \tag{3.1}$$

which takes the label index y and the image x as inputs and outputs a scalar value, known as the score, that falls within $[-1, 1]$. The functions ϕ_I and ϕ_T represent the image encoder and the text encoder, respectively, which include the normalization operation before the final output. The symbol $\langle \cdot, \cdot \rangle$ denotes the inner product.

Lastly, with a set of candidate classes $\mathcal{Y}$ and an image x, the method predicts the class $\hat{y}$ as the one with the highest CLIP score:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \texttt{CLIP}_1(y; x). \tag{3.2}$$

Template ensemble. Moreover, Radford et al. (2021) propose to ensemble different templates to improve inference performance. Specifically, they manually design 80 different templates $\{\alpha_i\}_{i=1}^{80}$ and use the following new CLIP

score for inference:

$$\texttt{CLIP}_{80}(y;x) \triangleq \left\langle \phi_I(x), \frac{\frac{1}{80}\sum_{i=1}^{80}\phi_T(\alpha_i(y))}{\left\|\frac{1}{80}\sum_{i=1}^{80}\phi_T(\alpha_i(y))\right\|} \right\rangle. \tag{3.3}$$

Using such template ensemble improves zero-shot classification accuracy. For example, from 66.3% to 68.2% for ViT-B/32 on ImageNet.

4. Problem Formulation

This section formulates the factors in the data generation process with their corresponding text descriptions, which are omitted or not systematically addressed in previous prompt studies. Moreover, we approximate several probabilities using the CLIP score, providing interpretations. These forms the foundation of our step-by-step reasoning method introduced in subsequent chapters.

4.1. Structuring Generative Factors and Their Descriptions

When observing an image, humans recognize not only the class of an object but also its attributes, such as color, style, and background. To formalize this intuition, we consider the following data generation process.

Data generation process. Let Y denote the underlying class label (e.g., *dog* or *cat*) that takes values in the label set $\mathcal{Y}$. Let each Z_i, $1 \le i \le m$ be a certain generative factor (e.g., *illumination*, *orientation*) that describes a certain attribute of the object and takes values in the attribute set $\mathcal{Z}_i$ (e.g., *bright* and *dark* for *illumination*). Then, we consider an image X to be generated as

$$Y \to X \leftarrow \{Z_i\}_{i=1}^m,$$

We only consider discrete $\mathcal{Z}_i$'s since we observe that CLIP cannot effectively encode text descriptions of continuous values.

Text descriptions for abstract generative factor values. Each attribute set $\mathcal{Z}_i$ only contains abstract discrete values in the data generation process. To bridge these abstract values with the corresponding text descriptions in captions, we use an annotation function that reflects the linguistic preferences of human annotators when describing specific attributes of objects in natural language.

The **annotation function** $\alpha : \mathcal{Z} \to \mathcal{P}(\text{texts})$ maps an abstract discrete value in $\mathcal{Z}$ to a random variable with a distribution over all possible natural language text descriptions. We reuse the notation α for simplicity, whereas its previous appearance in Eq. 3.1 is a special case when the input is discrete values in $\mathcal{Y}$ and the output random variable $\alpha(y)$ only takes the value of one text description. Figure 2 illustrates some examples.

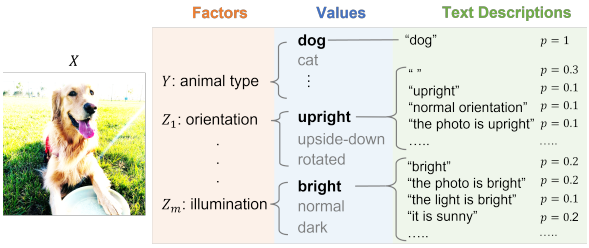

Figure 2. Illustration of some generative factors, their abstract values, and the corresponding distributions over text descriptions mapped by the underlying annotation function.

Note that the random text description may take the value of an empty string. This is more likely for common factor values in the dataset. For example, in the case of an upright image, human annotators often overlook descriptions of its upright orientation, whereas upside-down images usually have explicit descriptions of their inverted orientation.

With multiple factors describing different attributes of an object, we concatenate their descriptions together to form the final text description for an image. We use $\oplus$ to denote the **concatenation operation** that outputs a new random variable $\alpha(y) \oplus \alpha(z_1) \oplus \alpha(z_2) \oplus ...$ from multiple random variables $\alpha(y)$, $\alpha(z_1)$, $\alpha(z_2)$, ..., by concatenating their values separated by a comma. For example, when y represents "dog" and z represents "upright", the concatenation $\alpha(y) \oplus \alpha(z)$ can take the value of "*a photo of a dog, upright*".

4.2. Approximating Probability Distributions Using CLIP-Score

CLIP score with generative factors. Since previous CLIP scores do not explicitly consider the generative factors, we define a new CLIP score function that models them. Specifically, we define $\texttt{CLIP} : \mathcal{Y} \times \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_m \times \mathcal{X} \to \mathbb{R}$ via

$$\texttt{CLIP}(y, z_1, \ldots, z_m; x) \triangleq \tag{4.1}$$
$$\left\langle \phi_I(x), \frac{\mathbb{E}\ \phi_T(\alpha(y) \oplus \alpha(z_1) \oplus \cdots \oplus \alpha(z_m))}{\|\mathbb{E}\ \phi_T(\alpha(y) \oplus \alpha(z_1) \oplus \cdots \oplus \alpha(z_m))\|} \right\rangle.$$

CLIP score empirically captures generative factors. We observe that the CLIP score is high for correctly matched pairs of image and generative factors while low for incorrect ones. Specifically, it showcases the following property:

$$\texttt{CLIP}(y^*, z_i^*; x^*) \ge \texttt{CLIP}(y, z_i; x^*), \tag{4.2}$$
$$\forall\, y \in \mathcal{Y},\ \forall\, z_i \in \mathcal{Z}_i,\ \forall\, 1 \le i \le m$$

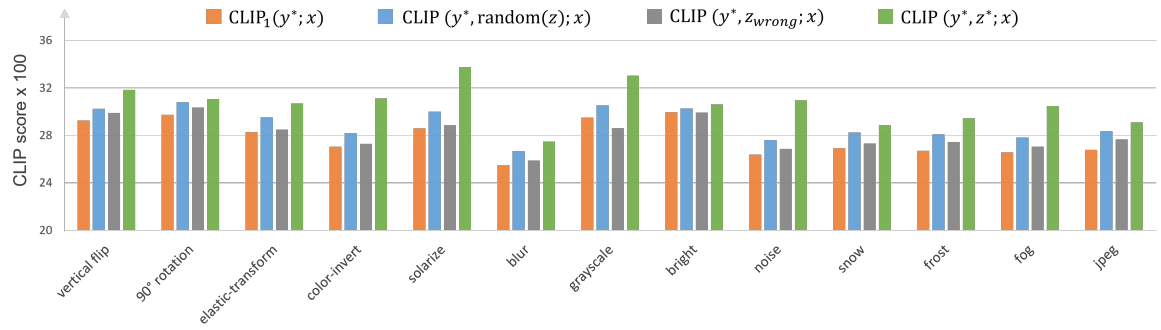

Figure 3. The averaged CLIP scores evaluated on the ImageNet test set. We apply different transformations to all the images to simulate different generative factors. y^* and z^* denote the ground-truth label and generative factor value. z_{wrong} denotes the wrong generative factor value. random(z) indicates that the randomized description is used in the text prompt following (Roth et al., 2023). CLIP scores indicate that CLIP has the knowledge of these generative factors and including the descriptions of correct generative factors in the text prompt yields better alignment with the corresponding image.

Table 2. Some (Bayesian) probabilities and their approximations using CLIP's similarity scores.

Probability	Approximation
$p(y, z\|x)$	$\frac{e^{\texttt{CLIP}(y,z;x)}}{\sum_y \sum_z e^{\texttt{CLIP}(y,z;x)}}$
$p(y\|x, z)$	$\frac{e^{\texttt{CLIP}(y,z;x)}}{\sum_y e^{\texttt{CLIP}(y,z;x)}}$
$p(z\|x)$	$\frac{\sum_y e^{\texttt{CLIP}(y,z;x)}}{\sum_z \sum_y e^{\texttt{CLIP}(y,z;x)}}$ or $\frac{e^{\texttt{CLIP}(z;x)}}{\sum_z e^{\texttt{CLIP}(z;x)}}$

where (y^*, z_i^*) are the ground-truth generative factors (consider y also as a generative factor) Figure 3 illustrates the empirical result.

Note that this property also aligns with the training objective of CLIP, since the contrastive training loss encourages high scores for correctly matched image-caption pairs while suppressing the score of incorrect ones.

Approximating probability distributions. Our subsequent analysis involves calculating the conditional probability $p(y, z_1, \ldots, z_m|x)$, $p(y|z_1, \ldots, z_m, x)$, and $p(y, z_1, \ldots, z_m|x)$. Since the CLIP score can take negative values and does not directly model any of them, we provide two ways to approximate them using the CLIP score. For notation simplicity, we use z to denote $(z_1, \ldots, z_m)$.

Table 2 shows the probabilities and our corresponding approximations. Observing the property in Eq. 4.2, our first approximation method is to view the CLIP score as an energy function that can model $p(y, z|x)$ by exponentiation and normalization. The rest two probability distributions can them be derived following the law of total probability.

5. CLIP Benefits from and Can Infer Generative Factors

This section presents two empirical observations: First, additionally conditioning on generative factors improves classification and mitigates spurious features. Second, CLIP itself has a certain capability for inferring generative factors from a given image. These two observations motivate our step-by-step reasoning method in the next section.

5.1. Telling CLIP Generative Factors Helps It Inferring Classes

To infer the class Y conditioning on the known generative factor z, we compute $\arg\max_y \; p(y|x, z)$ by resorting to the approximation in Table 2:

$$\begin{aligned} \arg\max_y \quad p(y|x, z) &= \arg\max_y \quad \frac{e^{\texttt{CLIP}(y,z;x)}}{\sum_y e^{\texttt{CLIP}(y,z;x)}} \\ &= \arg\max_y \quad e^{\texttt{CLIP}(y,z;x)} \\ &= \arg\max_y \quad \texttt{CLIP}(y, z; x), \end{aligned} \tag{5.1}$$

where the second equality holds because $\sum_y e^{\texttt{CLIP}(y,z;x)}$ is a constant of y.

Then, we evaluate if additionally conditioning on the ground-truth generative factors improves the inference accuracy of the class Y. Specifically, given an image x^*, its ground-truth generative factor z^*, and a randomly chosen generative factor z', we predict the class Y using the following three methods and compare the average accuracy over all test examples:

1. No generative factors considered:
   $\arg\max_y \texttt{CLIP}_1(y; x)$.
   Description example: "*a photo of a {classname}.*"

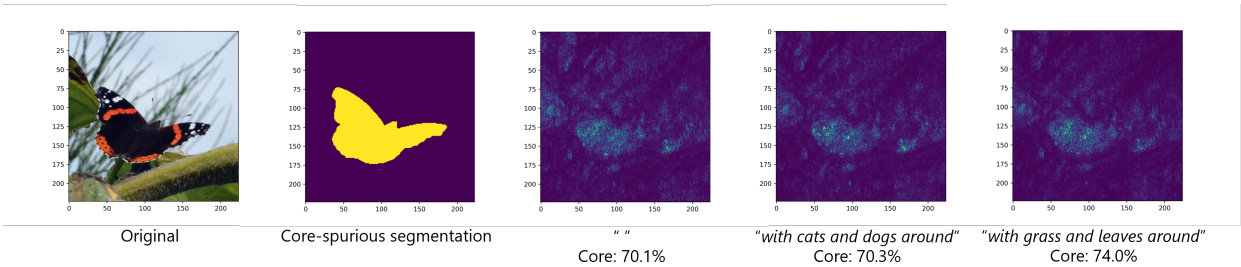

Figure 4. Visualization of the original image, the regions of core and spurious features, and three saliency maps corresponding to prompts with different contextual descriptions.

2. Conditioned on the random generative factor:
$\arg\max_y \texttt{CLIP}(y; x, z_{\text{random}})$.
Description example: "*a photo of a {classname}, iaYo5n0Dli7.*"

3. Conditioned on the wrong generative factor:
$\arg\max_y \texttt{CLIP}(y; x, z_{\text{wrong}})$.
Description example: "*a photo of a {classname}, upright.*"

4. Conditioned on the ground-truth generative factor:
$\arg\max_y \texttt{CLIP}(y; x, z^*)$.
Description example: "*a photo of a {classname}, upside-down.*"

Empirically verifying this hypothesis requires annotations of the latent factors for each image, which are not available. Therefore, we manually incorporate some latent factors into the image generation process. Specifically, we first confirm that most images in the ImageNet test set exhibit an upright orientation, natural illumination, and standard image quality. Then, we apply diverse transformations and corruptions controlled by the corresponding latent factors to alter the images. This incorporation of latent factors ensures that their underlying ground-truth values are known and become part of the generation process. Lastly, we use CLIP to do zero-shot classification, providing descriptions for correct or incorrect latent factors.

Result. Table 3 shows that conditioning on the correct latent factors indeed increases the prediction accuracy of the label when compared to conditioning on the incorrect ones, or intuitively, telling CLIP the correct latent factors helps its prediction. Furthermore, Figure 4 shows that conditioning on the correct generative factors reduces reliance on spurious features, which cannot be achieved by using random generative factors.

5.2. CLIP Can Infer Generative Factors

Our previous finding suggests improving the inference of CLIP by providing it with the correct latent factors. However, manually annotating latent factors for each image is impractical. Addressing this challenge, this section showcases the capability of CLIP to infer the correct latent factors from a given image reasonably.

To infer the generative factor Z from a given image x, we can compute $\arg\max_z \; p(z|x)$ by resorting to one of the two approximations in Table 2. The first approximation, which conditions on all possible ys and then aggregates them out, yields

$$\begin{aligned} \arg\max_z p(z|x) &= \arg\max_z \frac{\sum_y e^{\texttt{CLIP}(y,z;x)}}{\sum_y \sum_z e^{\texttt{CLIP}(y,z;x)}} \\ &= \arg\max_z \sum_y e^{\texttt{CLIP}(y,z;x)}, \end{aligned} \tag{5.2}$$

where the second equality holds because $\sum_y \sum_z e^{\texttt{CLIP}(y,z;x)}$ is a constant of z. An example of the corresponding description is "*a photo of a {classname}, {description of z}*".

Similarly, using the alternative simpler approximation yields

$$\begin{aligned} \arg\max_z p(z|x) &= \arg\max_z \frac{e^{\texttt{CLIP}(z;x)}}{\sum_z e^{\texttt{CLIP}(z;x)}} \\ &= \arg\max_z e^{\texttt{CLIP}(z;x)} \\ &= \arg\max_z \texttt{CLIP}(z; x). \end{aligned} \tag{5.3}$$

An example of the corresponding description is "*a photo of an object, {description of z}*".

To evaluate the inference of generative factors, similar to the setting in Section 5.1, we construct randomly altered images controlled by some manually incorporated generative factors with known ground-truth values. We apply each data transformation randomly to half of the images in the ImageNet test set while keeping the other half unchanged. In this case, inferring the factor values is a binary classification task with a random guessing accuracy of 50%. We report the average accuracy over five runs.

Result. Table 4 shows that CLIP can predict the correct generative variables with reasonable accuracy. This finding suggests that we may bootstrap CLIP's inference by conditioning on the generative factors inferred by itself.

Table 3. Classification accuracy (%) on ImageNet, conditioning on different generative factors.

Factor	**Acc**				
	w/o z	w/ random z	w/ wrong z	w/ correct z	w/ self-infer z
vertical flip	51.17	52.02 (↑0.85)	52.19 (↑1.02)	52.48 (↑1.31)	52.54 (↑**1.37**)
90° rotation	57.02	58.38 (↑1.36)	58.23 (↑1.21)	58.75 (↑**1.73**)	58.30 (↑1.28)
elastic-transform	48.66	48.45 (↓0.21)	48.75 (↑0.09)	48.89 (↑0.23)	49.00 (↑**0.34**)
color-invert	35.29	36.12 (↑0.83)	35.89 (↑0.60)	36.72 (↑1.43)	36.80 (↑**1.51**)
solarize	49.79	49.74 (↓0.05)	50.20 (↑0.41)	50.49 (↑0.70)	50.54 (↑**0.75**)
blur	38.86	39.65 (↑0.79)	39.21 (↑0.35)	39.92 (↑**1.06**)	39.80 (↑0.94)
grayscale	59.51	59.67 (↑0.16)	59.48 (↓0.03)	59.98 (↑0.47)	60.04 (↑**0.53**)
bright	60.81	62.04 (↑**1.23**)	60.94 (↑0.13)	61.41 (↑0.60)	61.28 (↑0.47)
noise	14.16	14.88 (↑0.72)	14.75 (↑0.59)	15.66 (↑1.50)	15.68 (↑**1.52**)
snow	33.09	32.94 (↓0.15)	33.56 (↑0.47)	34.50 (↑**1.41**)	34.33 (↑1.24)
frost	31.08	31.91 (↑0.83)	31.76 (↑0.68)	32.63 (↑1.55)	32.81 (↑**1.73**)
fog	37.61	38.40 (↑0.79)	38.00 (↑0.39)	39.31 (↑1.70)	39.34 (↑**1.73**)
jpeg	33.67	34.80 (↑1.13)	35.11 (↑1.45)	35.39 (↑1.72)	35.47 (↑1.80)
average		↑0.64	↑0.57	↑1.16	↑**1.17**

Table 4. The accuracy (%) of CLIP in predicting generating factors from images.

Factor	vflip	rotation	elastic	invert	solarize	blur	gray	bright	noise	snow	frost	fog	jpeg	Avg
W/ Y	76.30	68.65	72.03	78.67	74.67	62.91	84.67	56.98	66.00	86.56	82.39	89.11	66.66	74.28
W/o Y	75.77	61.58	66.37	80.79	82.11	73.99	70.19	62.17	79.68	86.75	81.19	95.28	67.49	75.64

6. `PerceptionCLIP`: Inference with Inferred Generative Factors

In this section, we propose `PerceptionCLIP`. Given an image, `PerceptionCLIP` first infers the generative factors of the image and refines the factors using human knowledge. Then, it infers the class of the image conditioning on the generative factors. We also show that the template ensemble can be viewed as a special case of `PerceptionCLIP` that does not include human intervention in the inferred generative factors.

Section 5 shows that inferring the class conditioning on the ground-truth generative factors via $\arg\max_y p(y|x,z)$ improves the prediction accuracy for the class Y. However, ground-truth generative factors are often unavailable in zero-shot inference. On the other hand, CLIP can infer the generative factors from a given image with a certain level of accuracy via $\arg\max_z p(z|x)$. Building on these observations, we propose to do zero-shot inference in two steps.

Step one: inferring generative factors. Since CLIP cannot perfectly infer the generative factors from a given image, we construct a distribution $\hat{p}(z|x)$ that models the uncertainty, instead of choosing the single most possible value. A natural choice for constructing $\hat{p}(z|x)$ is to directly use the approximation of $p(z|x)$ in Table 2. However, some existing results suggest that CLIP's estimation of long tail probabilities is not accurate. Therefore, we truncate the top-k probabilities of z in $p(z|x)$ and re-normalize them to construct $\hat{p}(z|x)$.

Step two: inferring the class. Due to the imperfect inference of the generative factors, instead of conditioning on a single inferred generative factor value, we use the constructed distribution $\hat{p}(z|x)$ and infer Y via

$$\arg\max_y \sum_z p(y|x,z)\hat{p}(z|x). \tag{6.1}$$

A simplified single-step version. We can also directly use the distribution $p(z|x)$ for constructing $\hat{p}(z|x)$, which yields a simplified implementation of `PerceptionCLIP` that essentially does inference and conditioning on generative factors in one step. Specifically, it follows that

$$\begin{aligned}\arg\max_y \sum_z p(y|x,z)p(z|x) &= \arg\max_y \sum_z p(y,z|x) \\ &= \arg\max_y \sum_z e^{\texttt{CLIP}(y,z;x)},\end{aligned} \tag{6.2}$$

where the second equality follows from the approximation in Table 2 and by omitting the denominator which is a constant of z and y.

The implementation includes: (1) for an input image, compute the CLIP score with multiple templates that describe all latent factors and class labels. (2) For each class label, sum over the latent factors to marginalize them out and get a score. (3) Choose the class label with the highest score.

Comparison. The one-step method, although simple in implementation, has two drawbacks. First, it does not allow

Table 5. Summary of generative factors and their descriptions. Here ϕ indicates an empty string.

Factor	Value Descriptions
orientation	ϕ, upside-down, rotated
background	ϕ, in water, in forest, in sky, at street, at outdoor, at home, in office
quality	ϕ, good, bad, low resolution, pixelated, jpeg corrupted, blurry, , clean, dirty
illumination	ϕ, bright, dark
quantity	ϕ, many, one, large, small
perspective	ϕ, close-up, cropped, hard to see
art	ϕ, sculpture, rendering, graffiti, tattoo, embroidery, drawing, doodle, , origami, sketch, art, cartoon
medium	ϕ, video game, plastic, toy, plushie
condition	ϕ, cool, nice, weird
color-scheme	ϕ, black and white
tool	ϕ, with pencil, with pen, digitally

Table 6. Zero-shot classification accuracy on five datasets using ViT-B/16. The best result in each column is highlighted in bold, while the next two highest values are underlined.

Factors		ImageNet	ImageNetV2	ImageNet-R	ImageNet-A	ImageNet-Sketch
single template		66.72%	60.85%	73.99%	47.80%	46.16%
80 templates		68.32%	61.93%	77.71%	49.95%	48.26%
single factor	background	67.70%	**61.91%**	75.71%	49.13%	47.21%
	illumination	66.91%	61.04%	74.60%	48.32%	46.08%
	orientation	67.21%	61.04%	74.31%	47.88%	46.54%
	quality	**68.11%**	61.78%	76.24%	**50.41%**	47.39%
	quantity	67.57%	61.39%	75.22%	50.08%	46.57%
	perspective	67.87%	61.36%	74.91%	49.55%	46.90%
	art	67.42%	60.94%	**77.08%**	49.59%	**47.95%**
	medium	67.22%	60.73%	76.30%	49.45%	46.78%
	condition	68.30%	61.64%	75.51%	49.25%	47.25%
	color-scheme	66.67%	60.70%	73.85%	48.07%	46.41%
	tool	66.70%	60.61%	75.32%	48.28%	47.22%
composition of top 2 factors		68.49%	61.95%	77.64%	50.85%	48.18%
composition of top 3 factors		**68.52%**	62.01%	77.92%	50.53%	48.39%
composition of top 4 factors		68.50%	**62.24%**	**77.95%**	**50.97%**	**48.79%**

human intervention during the process of inferring latent factors. Our validation reveals that CLIP does not always infer latent factors well, whereas human intervention can leverage prior knowledge to improve this. Second, the inference in the one-step approach prevents us from knowing the inferred latent factors, which could be utilized to improve the interpretability of the inference results.

Relationship to template (prompt) ensemble. Modulo the difference in templates, this implementation recovers the multi-template strategy adopted by the existing zero-shot inference method of CLIP, thus explaining its effectiveness. Nevertheless, the latter chooses the templates in an ad-hoc way, whereas our experiments indicate that the use of more diverse and systematic templates, describing all latent factors, further improves the inference of CLIP.

7. Experiments

In this section, we evaluate `PerceptionCLIP` in improving zero-shot generalization, providing interpretable prediction, improving group robustness, and mitigating spurious features. Since our method is training-free and deterministic, the quantitative results do not include error bars.

7.1. Improving Zero-Shot Generalization

We first evaluate the generalization of `PerceptionCLIP` by using a single generative factor to show the effects of different object attributes. Then, we extend the evaluation to multiple factors. Finally, we show how `PerceptionCLIP` can benefit from interventions on the inferred generative factors by leveraging prior knowledge. We test on the ImageNet dataset (Deng et al., 2009) and its out-of-distribution variants, including ImageNetV2 (Recht et al., 2019), ImageNet-R (Hendrycks et al., 2021a), ImageNet-A (Hendrycks et al.,

Table 7. Average accuracy and worst group accuracy on the Waterbirds dataset.

	ViT-B/32		ViT-B/16		RN50		ViT-L/14	
	Avg	**Worst**	**Avg**	**Worst**	**Avg**	**Worst**	**Avg**	**Worst**
CLIP	72.25	50.07	79.53	26.48	78.98	27.88	84.21	35.98
PerceptionCLIP ($\mathcal{Z}$={"*on land*", "*on water*"})	**81.14**	65.11	81.48	16.20	82.38	33.02	86.49	46.42
PerceptionCLIP ($\mathcal{Z}$={"*on land*", "*on water*", ...})	75.39	**68.25**	**82.97**	**36.60**	**83.10**	**38.16**	**87.63**	**60.75**

Table 8. Average accuracy and worst group accuracy on the CelebA dataset.

	ViT-B/32		ViT-B/16		RN50	
	Avg	**Worst**	**Avg**	**Worst**	**Avg**	**Worst**
CLIP	80.73	75.82	75.16	62.01	81.05	73.87
PerceptionCLIP ($\mathcal{Z}$={gender})	79.22	75.45	74.84	64.15	84.87	79.98
PerceptionCLIP ($\mathcal{Z}$={gender, age, race})	**81.08**	**75.82**	**78.45**	**70.85**	**87.37**	**86.24**

2021b), and ImageNet-Sketch (Wang et al., 2019).

Inference with single generative factors. We first utilized GPT-3 to compile a set of possible generative factors. This set systematically summarizes the generative factors involved in the 80 hand-crafted templates (Radford et al., 2021) and includes additional factors such as orientation, background, and drawing tools. Table 5 provides detailed descriptions of each factor and its potential values.

Table 6 presents the results of `PerceptionCLIP`. Here we use the two-step method. Compared to using the simple template "*a photo of {class name}*", considering almost any single generative factor improves the results. However, the impact of different generative factors varied. For example, considering only the *quality* factor (with 9 possible values) significantly improves accuracy, surpassing prompt ensemble with 80 templates (Radford et al., 2021) on ImageNet-A. Moreover, the most influential generative factors differed for different datasets, which may be due to their different data generation processes. For example, all images in ImageNet-Sketch are sketches, making *art* a crucial generative factor for image generation. This also indicates that `PerceptionCLIP` works the best when the considered generative factors cover the generation process of the downstream dataset.

Inference with multiple generative factors. The bottom section of Table 6 presents the results considering multiple generative factors. To this end, we simply concatenate the text descriptions of each generative factor using commas for details). As the number of factors considered increases, the classification accuracy gradually improves. By combining three factors, `PerceptionCLIP` outperforms prompt ensemble with 80 templates (Radford et al., 2021) on all datasets.

Intervening during inferring generative factors. Since CLIP's inference on generative factors is not always accurate, we leverage prior knowledge to intervene in this process and test if it helps. This intervention requires the two-step implementation of `PerceptionCLIP`. Here, we consider smoothing out CLIP's inference on generative factors to "acknowledge its uncertainty". To this end, we introduce a temperature hyperparameter t that is greater than 1 in $\hat{p}(z|x)$, or equivalently, replace $e^{\texttt{CLIP}(y,z;x)}$ with $e^{\texttt{CLIP}(y,z;x)/t}$ and replace $e^{\texttt{CLIP}(z;x)}$ with $e^{\texttt{CLIP}(z;x)/t}$ in the last row in Table 2. Table 9 (with $t = 5$) shows that this intervention achieves modest but consistent performance gains across different datasets.

Table 9. Results of intervening during inferring generative factors, using temperature$= 5$, ViT-B/16, and considering the combination of the top three factors.

	W/o intervention	W/ intervention	
		w/ y	w/o y
ImageNet	68.26%	**68.52%**	68.49%
ImageNetV2	61.89%	**62.01%**	**62.01%**
ImageNet-R	77.65%	77.92%	**77.96%**
ImageNet-A	50.42%	**50.53%**	50.50%
ImageNet-Sketch	48.41%	48.39%	**48.48%**

7.2. Improving Group Robustness and Mitigating Spurious Features

We evaluated the group robustness of `PerceptionCLIP` through bird type classification on the Waterbirds dataset (Sagawa* et al., 2020) and hair color classification on the CelebA (Liu et al., 2015) dataset. In both datasets, each image has an underlying group attribute unknown to the model. These group attributes are *background* in Waterbirds and *gender* in CelebA. They both spuriously correlate with the class labels but do not causally determine the labels, thus considered spurious features. When evaluating the worst group accuracy, we group the images based on their labels and group attributes, and evaluate the accuracy of each group.

Tables 7 and 8 show the results on the two datasets. When the text prompts only describe the labels, such as "*a photo of a {landbird/waterbird}.*" and "*a photo of a celebrity with {dark hair/blond hair}.*", the CLIP model exhibits biased accuracy, with a significant discrepancy between average accuracy and the accuracy of the worst-performing group. This bias arises because CLIP overly relies on spurious features, such as directly associating images with a water background to the water bird class, instead of focusing on the core features of the subject. By inferring and conditioning on the group attribute, `PerceptionCLIP` reduces reliance on spurious features and mitigates the bias.

8. Conclusion

Through systematic interpretation and structuring of the prompt, we showcase CLIP's abilities to understand and infer the factors involved in the data generation process. Based on this, we propose `PerceptionCLIP`, which achieves better generalization, less reliance on spurious features, and improved interpretability through self-inference and conditioning of the generative factors. Our work showcases CLIP, as a model with the unprecedented ability to communicate with humans through natural language, still holds enormous potential in zero-shot reasoning.

References

Dai, W., Li, J., Li, D., Tiong, A. M. H., Zhao, J., Wang, W., Li, B., Fung, P., and Hoi, S. Instructblip: Towards general-purpose vision-language models with instruction tuning. (arXiv:2305.06500), Jun 2023. URL http://arxiv.org/abs/2305.06500. arXiv:2305.06500 [cs].

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255. Ieee, 2009.

Derakhshani, M. M., Sanchez, E., Bulat, A., da Costa, V. G. T., Snoek, C. G. M., Tzimiropoulos, G., and Martinez, B. Bayesian prompt learning for image-language model generalization, 2023.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8340–8349, 2021a.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15262–15271, 2021b.

Huang, T., Chu, J., and Wei, F. Unsupervised prompt learning for vision-language models. arXiv preprint arXiv:2204.03649, 2022.

Kandel, E. R., Schwartz, J. H., Jessell, T. M., Siegelbaum, S., Hudspeth, A. J., Mack, S., et al. Principles of neural science, Fifth Edition, volume 4. McGraw-hill New York, 2013.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. Segment anything. (arXiv:2304.02643), Apr 2023. doi: 10.48550/arXiv.2304.02643. URL http://arxiv.org/abs/2304.02643. arXiv:2304.02643 [cs].

Larochelle, H., Erhan, D., and Bengio, Y. Zero-data learning of new tasks. In AAAI, volume 1, pp. 3, 2008.

Li, H., Niu, H., Zhu, Z., and Zhao, F. Cliper: A unified vision-language framework for in-the-wild facial expression recognition. ArXiv, abs/2303.00193, 2023a.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. (arXiv:2301.12597), Jun 2023b. URL http://arxiv.org/abs/2301.12597. arXiv:2301.12597 [cs].

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. (arXiv:2304.08485), Apr 2023. URL http://arxiv.org/abs/2304.08485. arXiv:2304.08485 [cs].

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In Proceedings of International Conference on Computer Vision (ICCV), December 2015.

Mao, C., Teotia, R., Sundar, A., Menon, S., Yang, J., Wang, X. E., and Vondrick, C. Doubly right object recognition: A why prompt for visual rationales. ArXiv, abs/2212.06202, 2022.

Menghini, C., Delworth, A., and Bach, S. H. Enhancing clip with clip: Exploring pseudolabeling for limited-label prompt tuning. (arXiv:2306.01669), Jun 2023. doi: 10.48550/arXiv.2306.01669. URL http://arxiv.org/abs/2306.01669. arXiv:2306.01669 [cs].

Menon, S. and Vondrick, C. Visual classification via description from large language models. arXiv preprint arXiv:2210.07183, 2022.

Mirza, M. J., Karlinsky, L., Lin, W., Kozinski, M., Possegger, H., Feris, R., and Bischof, H. Lafter: Label-free tuning of zero-shot classifier using language and unlabeled image collections. (arXiv:2305.18287),

May 2023. URL http://arxiv.org/abs/2305.18287. arXiv:2305.18287 [cs].

Novack, Z., Garg, S., McAuley, J., and Lipton, Z. C. Chils: Zero-shot image classification with hierarchical label sets. ArXiv, abs/2302.02551, 2023.

Pratt, S., Liu, R., and Farhadi, A. What does a platypus look like? generating customized prompts for zero-shot image classification. arXiv preprint arXiv:2209.03320, 2022.

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

Ramaswamy, V. V., Lin, S. Y., Zhao, D., Adcock, A. B., van der Maaten, L., Ghadiyaram, D., and Russakovsky, O. Geode: a geographically diverse evaluation dataset for object recognition. (arXiv:2301.02560), Apr 2023. doi: 10.48550/arXiv.2301.02560. URL http://arxiv.org/abs/2301.02560. arXiv:2301.02560 [cs].

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. (arXiv:2204.06125), Apr 2022. doi: 10.48550/arXiv.2204.06125. URL http://arxiv.org/abs/2204.06125. arXiv:2204.06125 [cs].

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In International conference on machine learning, pp. 5389–5400. PMLR, 2019.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. pp. 10684–10695, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Roth, K., Kim, J. M., Koepke, A. S., Vinyals, O., Schmid, C., and Akata, Z. Waffling around for performance: Visual classification with random words and broad concepts, 2023.

Sagawa*, S., Koh*, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114, 2021.

Shu, M., Nie, W., Huang, D.-A., Yu, Z., Goldstein, T., Anandkumar, A., and Xiao, C. Test-time prompt tuning for zero-shot generalization in vision-language models. arXiv preprint arXiv:2209.07511, 2022.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. Advances in Neural Information Processing Systems, 32, 2019.

Wang, Z., Wu, Z., Agarwal, D., and Sun, J. Medclip: Contrastive learning from unpaired medical images and text. ArXiv, abs/2210.10163, 2022.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., and Zhou, D. Chain of thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903, 2022.

Yang, Y., Nushi, B., Palangi, H., and Mirzasoleiman, B. Mitigating spurious correlations in multi-modal models during fine-tuning. ArXiv, abs/2304.03916, 2023.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. arXiv preprint arXiv:2305.10601, 2023.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Conditional prompt learning for vision-language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16816–16825, 2022a.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models. International Journal of Computer Vision, 130(9):2337–2348, 2022b.

Zhu, B., Niu, Y., Han, Y., Wu, Y., and Zhang, H. Prompt-aligned gradient for prompt tuning. arXiv preprint arXiv:2205.14865, 2022.

Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. (arXiv:2304.10592), Apr 2023. URL http://arxiv.org/abs/2304.10592. arXiv:2304.10592 [cs].