More Context, Less Distraction: Visual Classification by Inferring and Conditioning on Contextual Attributes

Bang An*
University of Maryland
bangan@umd.edu

Sicheng Zhu*
University of Maryland
sczhu@umd.edu

Michael-Andrei Panaitescu-Liess
University of Maryland
mpanaite@umd.edu

Chaithanya Kumar Mummadi
Bosch Center for Artificial Intelligence
chaithanyaKumar.mummadi@de.bosch.com

Furong Huang
University of Maryland
furongh@umd.edu

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 leverage 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: a modern neuroscience view suggests that in classifying an object, humans first infer its class-independent attributes (e.g., background and orientation) which help separate the foreground object from the background, and then make decisions based on this information. Inspired by this, we observe that providing CLIP with contextual attributes improves zero-shot classification and mitigates reliance on spurious features. We also observe that CLIP itself can reasonably infer the attributes from an image. With these observations, we propose a training-free, two-step zero-shot classification method named `PerceptionCLIP`. Given an image, it first infers contextual attributes (e.g., background) and then performs object classification conditioning on them. Our experiments show that `PerceptionCLIP` achieves better generalization, group robustness, and better interpretability. For example, `PerceptionCLIP` with ViT-L/14 improves the worst group accuracy by 16.5% on the Waterbirds dataset and by 3.5% on CelebA.

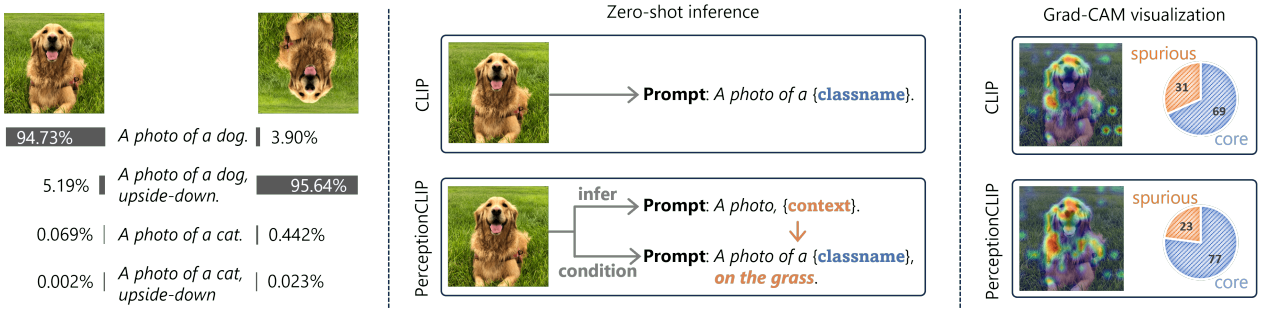

Figure 1: **(Left)**: CLIP understands natural language descriptions of contextual attributes (here, the orientation). **(Center)**: Compared to standard zero-shot classification method, which uses a fixed template for class name retrieval, our method first infers object attributes as context (here, the background), and then predicts the class conditioned on the inferred contextual attributes. **(Right)**: The corresponding visualization (Grad-CAM (Selvaraju et al., 2017)) showcases that our method focuses more on the core features (here, the dog), and it is less distracted by spurious features (here, the background) when doing the classification.

*Equal contribution

1 Introduction

CLIP (Contrastive Language-Image Pretraining, Radford et al. (2021)) is a foundational Visual Language Model (VLM) that bridges gap between the fields of vision and natural language. By pretraining on 400 million image-caption pairs, CLIP can associate various visual concepts with their corresponding natural language descriptions, making it the foundation for numerous other visual language models (Dai et al., 2023, Li et al., 2023b, Liu et al., 2023, Zhu et al., 2023), diffusion models (Ramesh et al., 2022, Rombach et al., 2022), and semantic segmentation models (Kirillov et al., 2023). Such remarkable understanding capability of CLIP finds its significance in an important application of zero-shot classification (Larochelle et al., 2008) — an open-ended image classification through natural language without the need for training or finetuning. 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 shown to exhibit strong potential for zero-shot classification, the corresponding methodology lacks systematic investigation, leading to sub-optimal generalization (Radford et al., 2021), reliance on spurious features (Yang et al., 2023), biased predictions (Agarwal et al., 2021, Chuang et al., 2023), and lack of interpretability (Menon and Vondrick, 2022, Zhou et al., 2022b). 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 in Radford et al. (2021) uses a simple template *"a photo of a {class name}"* to find the most relevant class (Figure 1) for a given image, which however differs from the image captions in the pretraining data (see examples in Table 9). Another method called prompt ensembling (Radford et al., 2021) manually crafts 80 unstructured templates and uses their average embeddings for class name retrieval, achieving better generalization. Nevertheless, it remains unclear whether these templates are optimal and why they are effective. By treating zero-shot classification simply as a class name retrieval problem, these methods potentially waste the capability of CLIP to understand both class-specific features and class-independent attributes (such as background and orientation, referred to as contextual attributes in this paper).

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

> *"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."*
>
> *"... 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, differing from current zero-shot classification methods which overlook contextual attributes. An example reflecting this view is when humans classify objects in images, as shown in Figure 1 (left), we unconsciously acquire its additional contextual attributes, such as the background (grass) and orientation (forward), since these pieces of information are byproducts of the perception process. Another example is when given a rotated image, humans first infer that the image is rotated and then calibrate the classification accordingly. Notably, modeling such a process, which involves iterative inferring and conditioning on contextual attributes, is challenging for traditional visual classification models.

Building on this insight, we propose a zero-shot classification method called `PerceptionCLIP`, which emulates a crucial part of human visual perception — inferring and conditioning on the contextual attributes — resulting in improved generalization (Table 4 and 5), reduced reliance on spurious features (Figure 1 right, Figure 4), reduced bias (Table 7 and 8), and better interpretability. Our contributions are as follows:

Structural analysis of CLIP's prompt and similarity score. CLIP is not readily capable of emulating the perception process, as it is essentially a function scoring the similarity between images and text, so we prepare CLIP for this purpose (§4). First, we define contextual attributes as the generative factors in the data generation process that are causally independent of the class. Emulating the perception process requires CLIP to understand these symbolic attributes, so we then design textual descriptions for these attributes following the linguistic preferences in the captions of the pretraining data (Figure 2), and structure CLIP's prompt as the composition of class and attributes, thus achieving CLIP's understanding of these attributes (Figure 3). Lastly, we use CLIP's similarity score to approximate some conditional probabilities needed for emulating perception (Table 1).

Revealing that contextual attributes are helpful and inferable. Although CLIP can now understand contextual attributes, it remains unclear whether these attributes help its classification or can be inferred by itself, so we showcase two proof-of-concept observations (§5). First, conditioning on ground-truth contextual attributes improves CLIP's zero-shot classification and mitigates reliance on spurious features, whereas providing incorrect or random attributes does not have this effect (Figure 3, Table 2, Figure 4). Second, CLIP can reasonably infer contextual attributes from a given image, with much higher accuracy than random guessing (Table 3).

A zero-shot classification algorithm emulating human perception. Based on the observations, we propose `PerceptionCLIP`, a training-free zero-shot classification method (§6). Given an image, it first employs CLIP to infer contextual attributes, with optional prior-knowledge-based intervention such as smoothing. Then, it uses CLIP to infer the class conditioned on the attributes by incorporating the descriptions of the inferred attributes into the prompt. This step-wise inference process resembles the concept of chain-of-thoughts in large language models. Notably, `PerceptionCLIP` subsumes prompt ensembling as a special (and sub-optimal) case which implicitly does the two perception steps, thus explaining the latter's effectiveness, guiding systematic prompt design, and suggesting that further weighting of the ensembled prompts is unnecessary.

Empirical evaluation on generalization, group robustness, and interpretability. We evaluate `PerceptionCLIP` on two zero-shot classification tasks: standard generalization and group robustness, where the latter measures the performance consistency across different data subgroups (Liu et al., 2021) (§7). For generalization, `PerceptionCLIP` consistently outperforms baselines that use simple templates and prompt ensembles (Table 4 and 5). Moreover, its inferred attributes naturally provide better interpretability, explaining how certain predictions are made. For group robustness, `PerceptionCLIP` with ViT-L/14 reduces the gap between average accuracy and worst group accuracy by 19.29% on the Waterbirds dataset (Table 7) and by 7.41% on the CelebA dataset (Table 8), showing less bias and mitigated reliance on spurious features (Figure 4).

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, Mao et al. (2022), Menon and Vondrick (2022), Pratt 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 (i.e., contextual attributes) such as background and orientation, whose comprehension by CLIP is not well-known. These attributes are also combinatorial, covering more aspects of an image than just a few class-exclusive components. Moreover, we can still leverage contextual attributes (e.g., gender, age) when class-exclusive components are hard to articulate, as in the hair-color classification tasks on CelebA (Liu et al., 2015). We also find that specifying contextual attributes can help reduce 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, indicating that our method's effectiveness stems from the proper use of contextual attributes instead of noise augmentation. Roth et al. (2023) also show that appending high-level class-independent descriptions (e.g., "food" for Food101 (Bossard et al., 2014), "place" for Places365 (Zhou et al., 2017)) 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 (Derakhshani et al., 2023, Zhou et al., 2022a,b, Zhu et al., 2022) or unlabeled downstream data (Huang et al., 2022, Menghini et al., 2023, Mirza 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 general image attributes (e.g, illumination) 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 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 basic and prompt ensembling methods for zero-shot classification of CLIP. Notably, we revisit the captions in CLIP's pretraining data, demonstrating their misalignment with the textual descriptions in the templates used for existing zero-shot classification methods.

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, and omit the subscript Z when the function's meaning can be inferred from the input notation z.

Pretraining of CLIP. CLIP is pretrained on a dataset of 400 million image-caption pairs collected from the internet. The training process involves learning to identify the correct pairing between a given image and a caption (textual description) from a set of possible pairings. This is achieved through a contrastive loss function, which encourages the model to produce similar representations for correctly matched image-caption pairs and dissimilar representations for mismatched pairs. The model's goal is to learn to generalize from these associations to new images and textual descriptions in a way that captures the shared semantic content.

Captions in the pretraining data. The human-written caption for each image typically describes the visual object, encompassing its class and a few contextual attributes like color, style and background (Radford et al., 2021). For reference, we show some caption examples in Table 9 in the appendix, which are chosen from a similar dataset LAION-400M (Schuhmann et al., 2021) since the original pretraining dataset of CLIP is not made public.

Zero-shot classification: basic method. After pretraining, Radford et al. (2021) provide a simple method for zero-shot visual classification. First, the authors use a universal (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 textual description. More abstractly, for any class index y in the class set $\mathcal{Y}$ and any **image** x in the image space $\mathcal{X}$, the CLIP model can be viewed as a score function $\texttt{CLIP}_1 : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$ via

$$\texttt{CLIP}_1(y;x) \triangleq \langle \phi_I(x),\, \phi_T(\alpha(y)) \rangle, \tag{1}$$

which takes y and x as inputs and outputs a scalar value, known as the similarity score, that falls within $[-1,1]$. The functions ϕ_I and ϕ_T represent the **image encoder** and the **text encoder**, respectively, which involve the normalization operation before the final output. Here, the symbol $\langle \cdot, \cdot \rangle$ denotes the inner product, and the subscript "1" indicates that the $\texttt{CLIP}_1$ score only uses one textual template. Then, given an image x, the method predicts the class $\hat{y} \in \mathcal{Y}$ as the one with the highest $\texttt{CLIP}_1$ score:

$$\hat{y} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \ \texttt{CLIP}_1(y;x). \tag{2}$$

Zero-shot classification: prompt (template) ensembling. In addition, Radford et al. (2021) propose to ensemble different templates to improve inference performance. Instead of using only one simple template α, the authors manually design 80 different templates $\{\alpha_i\}_{i=1}^{80}$, such as *"a bad photo of a {classname of y}"* and *"a sculpture of a {classname of y}"*. Then, they replace $\texttt{CLIP}_1$ with the following $\texttt{CLIP}_{80}$ score for inference, which improves zero-shot classification accuracy (e.g., from 66.7% to 68.3% for ViT-B/16 on ImageNet, see Table 4):

$$\texttt{CLIP}_{80}(y;x) \triangleq \left\langle \phi_I(x), \frac{\frac{1}{80}\sum_{i=1}^{80}\phi_T(\alpha_i(y))}{\left\| \frac{1}{80}\sum_{i=1}^{80}\phi_T(\alpha_i(y)) \right\|} \right\rangle. \tag{3}$$

4 Preparing CLIP for Perception

This work aims at emulating a key aspect of human visual perception using CLIP: inferring and conditioning on contextual attributes. However, CLIP cannot do this out of the box as it is essentially a function scoring the similarity between images and text. To prepare CLIP for this goal, this section structures the contextual attributes in the data generation process, describes them in text understandable to CLIP, and connects the conditional probabilities required for the modeling process with the CLIP score. These results also highlight the advantages of CLIP over traditional visual models in modeling this process.

4.1 Structuring and Describing Contextual Attributes

Prompt ensembling involves some contextual attributes in the templates but lacks a systematic analysis. Addressing this issue, we consider the data generation process to structure the contextual attributes in the templates, and discuss their corresponding text descriptions.

Contextual attributes as generative factors. We consider contextual attributes as generative factors that contribute to the data generation process. Specifically, let Y denote the underlying object class (e.g., *dog*) that takes values in the class set $\mathcal{Y}$. Let each Z_i, $1 \leq i \leq m$ denotes a certain **contextual attribute** of the object (e.g., *illumination*) that takes values in the contextual attribute set $\mathcal{Z}_i$ (e.g., {*bright*, *dark*}) and is causally independent (Pearl, 2009) of the object class Y. Then, we consider an image X to be generated as

$$Y \rightarrow X \leftarrow \{Z_i\}_{i=1}^{m}.$$

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

Textual descriptions for contextual attributes. The values of the generative factors in the data generation process are abstract, symbolized discrete values, but CLIP cannot directly accept these values as input; instead, it requires semantic text, thus creating a gap. When only considering the class, this gap is negligible because their textual descriptions rarely have ambiguities. For instance, we almost always use the term "*golden retriever*" to describe dogs of that breed. Therefore, their symbolic discrete values can be directly taken as their class names to input into CLIP. However, the textual description of the contextual attributes is more vague. For instance, for the upright value of an object's orientation attribute, people, especially those writing captions in the pretraining data, may use terms like "upright" or "upstanding," or they may not use any description at all because it is a common direction. To model this ambiguity and connect the discrete values of the contextual attributes with the text that CLIP can understand, we introduce the following annotation function.

The **annotation function** $\alpha : \mathcal{Z} \to \mathcal{P}(\text{text})$ maps a symbolic discrete value in $\mathcal{Z}$ to a distribution over natural language textual descriptions. Figure 2 illustrates some examples: for the value "upright", the annotation function maps it to several possible descriptions, with " " (the empty string) being the most likely. The simple template in Equation 1, which also uses the notation α, can be viewed as a special case where the output distribution is concentrated on a single textual description. We use the annotation function to model people's preference for descriptions when captioning pretraining images.

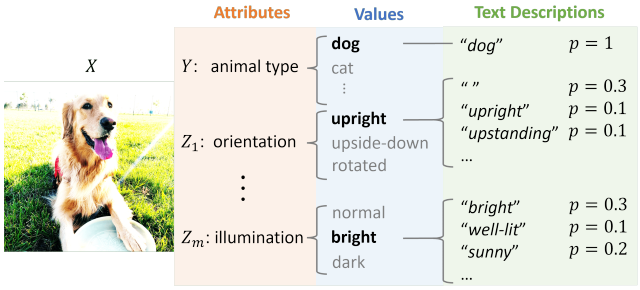

Figure 2: Illustration of contextual attributes, their symbolic discrete values, and the possible textual descriptions mapped by the annotation function.

With different contextual attributes depicting different aspects of the context, we concatenate their corresponding textual descriptions to form the final textual description of the image.

To this end, we consider the **concatenation operation**, denoted by $\oplus$, that takes multiple description distributions $\alpha(y)$, $\alpha(z_1)$, $\alpha(z_2)$, ... as input and outputs a new description distribution $\alpha(y) \oplus \alpha(z_1) \oplus \alpha(z_2) \oplus \ldots$ by concatenating their descriptions separated by a comma. For example, when y represents "*dog*", z_1 represents "*upright*", and z_2 represents "*bright*", the concatenation $\alpha(y) \oplus \alpha(z)$ can take the description of "*a photo of a dog, upright, bright*".

4.2 Connecting Conditional Probabilities with CLIP-Score

The human visual perception process involves inference conditioning on the contextual attributes. However, the corresponding conditional probabilities required to emulate this process cannot be directly modeled or approximated by existing CLIP scores. Therefore, we define a new attribute-aware CLIP score and approximate required conditional probabilities in terms of it.

Attribute-aware CLIP score. Existing CLIP scores such as $\texttt{CLIP}_1$ and $\texttt{CLIP}_{80}$ are agnostic of the contextual attributes and thus cannot be used to approximate conditional probabilities that are attribute-dependent. Therefore, we define a new score function $\texttt{CLIP} : \mathcal{Y} \times \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_m \times \mathcal{X} \to \mathbb{R}$ as follows:

$$\texttt{CLIP}(y, z_1, \ldots, z_m; x) \triangleq \left\langle \phi_I(x), \frac{\mathbb{E}\, \phi_T\big(\alpha(y) \oplus \alpha(z_1) \oplus \cdots \oplus \alpha(z_m)\big)}{\|\mathbb{E}\, \phi_T\big(\alpha(y) \oplus \alpha(z_1) \oplus \cdots \oplus \alpha(z_m)\big)\|} \right\rangle. \tag{4}$$

The $\texttt{CLIP}$ score takes contextual attributes as additional inputs, describes them internally alongside the class through the annotation function, and calculates the similarity with the image in the embedding space. The expectation is taken over the randomness of the textual descriptions of contextual attributes.

$\texttt{CLIP}$ score aligns with empirical observation. We observe that the defined $\texttt{CLIP}$ score captures the contextual attributes: it is high for correctly matched image-attribute pairs, while low for mismatched ones, thus behaving like an energy function (LeCun et al., 2006) that can be used to approximate probabilities. More formally, when $(y^*, z_1^*, \ldots, z_m^*)$ are the ground-truth class and attributes that generate x^* whereas $(y, z_1, \ldots, z_m)$ are some arbitrary class and attributes, the $\texttt{CLIP}$ score showcases the following property:

$$\texttt{CLIP}(y^*, z_1^*, \ldots, z_m^*) \geq \texttt{CLIP}(y, z_1, \ldots, z_m), \quad \forall y \in \mathcal{Y},\ \forall z_i \in \mathcal{Z}_i,\ \forall 1 \leq i \leq m \tag{5}$$

Figure 3 shows the empirical result considering a single attribute (more details and results appear in Appendix B.1). Given the pretraining process of CLIP, this observation may not be surprising, since the contrastive loss encourages high scores for correctly matched image-caption pairs, where the caption contains not only the class description but also descriptions of contextual attributes.

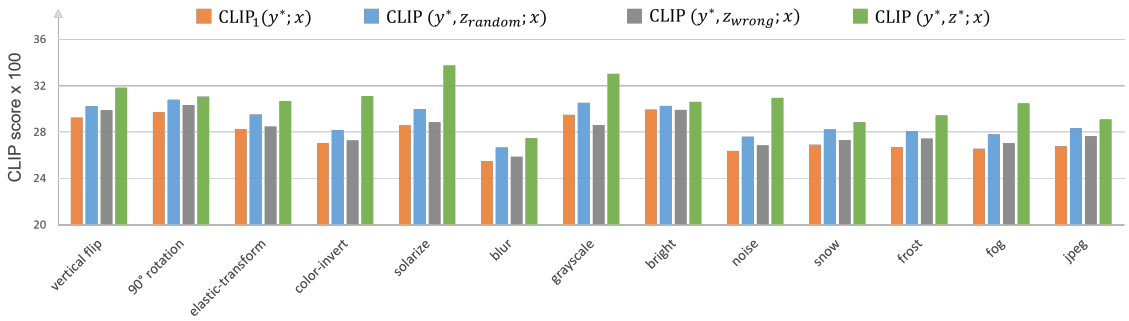

Figure 3: Evaluating CLIP scores on ImageNet with different image transformations altering the contextual attributes. The new attribute-aware CLIP score gives higher scores for correctly matched image-attribute pairs (green) while giving lower scores for mismatched pairs (grey) and random pairs (blue), indicating that CLIP understands our textual descriptions of contextual attributes and that the CLIP score measures the similarity between images and contextual attributes. In contrast, the original CLIP score (orange) is attribute-agnostic.

Approximating conditional probabilities. With the new energy-function-like CLIP score, we now approximate the conditional probabilities required for simulating human visual perception in subsequent sections. Specifically (in Table 1), we approximate 1) the joint conditional probability $p(y,z_1,...,z_m|x)$, which measures the likelihood of an object class and some contextual attributes occurring together given the image, and is only used to derive other probabilities; 2) the conditional probability $p(y|z_1,...,z_m,x)$, which measures the probability of an object class given both the image and the contextual attributes, and is our main inference objective; 3) the conditional joint probability $p(z_1,...,z_m|x)$, which measures the likelihood of some contextual attributes given the image, and is used for inferring the contextual attributes.

Table 1: Conditional probabilities required for emulating perception and the approximations using CLIP. Here, x, y, and z denote image, class, and contextual attributes, respectively.

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 $\frac{e^{\texttt{CLIP}(z;x)}}{\sum_z e^{\texttt{CLIP}(z;x)}}$

Given that the CLIP score (Equation 5) behaves like an energy function, we first use it to approximate $p(y,z_1,\ldots,z_m|x)$, requiring only exponentiation and normalization, and then derive the rest two using the law of total probability. Table 1 shows the approximations, where we use z to denote $(z_1,...,z_m)$ for notation simplicity. We provide two approximations for $p(z|x)$, referred to as "with Y" (left) and "without Y" (right). The textual description corresponding to $\texttt{CLIP}(y,z;x)$ in "with Y" is "*a photo of a {classname}, {description of z}*", while the textual description corresponding to $\texttt{CLIP}(z;x)$ in "without Y" is "*a photo of an object, {description of z}*".

5 Contextual Attributes are Helpful and Inferable

This section illustrates proof-of-concept experiments to show that emulating human visual perception by inference conditioned on contextual attributes improves zero-shot classification. Furthermore, such improvement does not require using ground-truth attributes, as CLIP itself can infer attributes and they already help.

5.1 Conditioning on Contextual Attributes Improves Zero-Shot Classification

We now evaluate if additional conditioning on the ground-truth contextual attributes improves the zero-shot classification accuracy for the object class. To this end, we find the most likely class Y given an image using

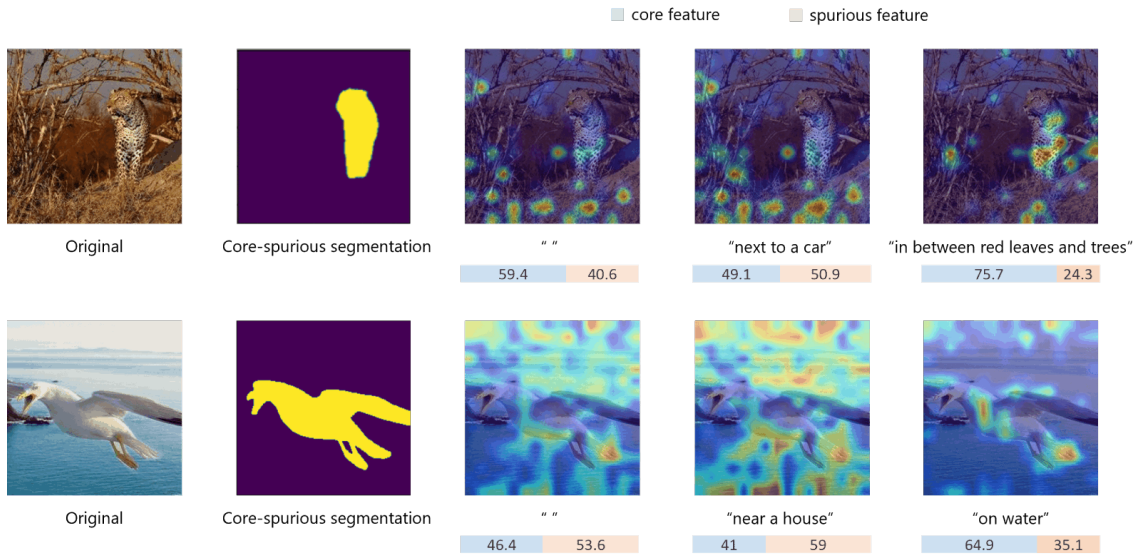

Figure 4: Visualization of the original image, the regions of core and spurious features, and the Grad-CAMs obtained using no, incorrect, and ground-truth contextual attributes (text below each image).

$\text{argmax}_y\, p(y|x,z)$ and its approximation in Table 1:

$$\underset{y}{\text{argmax}}\ p(y|x,z) = \underset{y}{\text{argmax}}\ \frac{e^{\texttt{CLIP}(y,z;x)}}{\sum_y e^{\texttt{CLIP}(y,z;x)}} = \underset{y}{\text{argmax}}\ e^{\texttt{CLIP}(y,z;x)} = \underset{y}{\text{argmax}}\ \texttt{CLIP}(y,z;x), \tag{6}$$

where the second equality holds because $\sum_y e^{\texttt{CLIP}(y,z;x)}$ is a constant of y. Intuitively, we classify an image using both possible classes and the ground-truth contextual attributes and find the class with the highest CLIP score.

Experimental setting. To show the benefit of conditioning on contextual attributes, we compare the following four methods in zero-shot classification, where the last two are for ablation study:

Conditioning on	Calculation	Prompt example
No contextual attributes	$\text{argmax}_y\ \texttt{CLIP}_1(y;x)$.	*a photo of a {classname}.*
Ground-truth contextual attributes	$\text{argmax}_y\ \texttt{CLIP}(y;x,z^*)$.	*a photo of a {classname}, upside-down.*
Wrong contextual attributes	$\text{argmax}_y\ \texttt{CLIP}(y;x,z_{\text{wrong}})$.	*a photo of a {classname}, upright.*
Random contextual attributes	$\text{argmax}_y\ \texttt{CLIP}(y;x,z_{\text{random}})$.	*a photo of a {classname}, iaYo5n0Dli7.*

We evaluate these methods on the ImageNet dataset. Due to the lack of annotated contextual attributes, we considered some easily observable and adjustable attributes, including image orientation, illumination, and image quality. We first examine and confirm that most ImageNet images share the same attribute values, including upright orientation, natural illumination, and standard image quality. However, these default values are too trivial, making their textual descriptions unlikely to appear in the captions of the pretraining data. Therefore, we then alter these attribute values through certain image transformations (e.g., vertical flipping), thus making the new attribute values have non-trivial descriptions. These new attribute values become part of the modified images' data generation process, for which we have ground-truth annotations.

We also do a proof-of-concept experiment to evaluate if conditioning on contextual attributes, especially those with spurious correlation to classes, could mitigate the model's reliance on them during classification. To this end, we consider some spatially separable spurious attributes (also known as spurious features), such as backgrounds, and annotate their corresponding regions with the help of *Segment Anything* (Kirillov et al., 2023). Then, we use Grad-CAM to compute the model's salient pixels, where the gradient computation uses the

Table 2: Classification accuracy (%) of five attribute-conditioned CLIP zero-shot classification methods on ImageNet (ViT-B/16). Different methods condition on different contextual attributes. We apply the left-side image transformations to alter the corresponding attribute values.

Contextual attribute	**Accuracy**				
	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**

scalar value corresponding to the correct class of the model's softmax output. Due to the softmax operation, computing this scalar value considers the combination of certain fixed context attributes with all possible classes. Intuitively, the regions where the salient pixels are located are the regions the model pays attention to when making predictions, and we hope that the model focuses as much as possible on regions of core features (i.e., features with causal correlation to classes). To quantify the model's reliance on spurious features, we aggregated the salient pixels within the spurious and core regions and calculated their ratios (see details in Appendix B.3).

Result. Table 2 shows that compared to not using or conditioning on incorrect contextual attributes, conditioning on correct contextual attributes improves classification accuracy. Moreover, Figure 4 illustrates that conditioning on correct contextual attributes reduces reliance on spurious features, suggesting a potential benefit of group robustness. As an ablation study, conditioning on randomly generated contextual attributes does not yield similar benefits in terms of accuracy and reduced spurious feature reliance. These results highlight the advantages of leveraging CLIP's understanding of contextual attributes to emulate the perception process. More results, including combined attributes and more visualized examples, appear in Appendix C. The last column of Table 2 indicates that using attributes inferred by CLIP itself leads to similar improvements as using ground-truth attributes, which will be discussed in subsequent sections.

5.2 Contextual Attributes are Inferable

While conditioning on ground-truth contextual attributes improves zero-shot classification accuracy and mitigates reliance on spurious features, manually annotating the attributes for each input image is impractical. Addressing this issue, we now showcase CLIP's ability to infer contextual attributes from a given image.

To infer the contextual attribute z from a given image x using CLIP, we calculate $\operatorname{argmax}_z p(z|x)$ using one of the two approximations in Table 1. The first approximation, which conditions on all possible ys and then marginalizes them out, yields

$$\operatorname*{argmax}_z p(z|x) = \operatorname*{argmax}_z \frac{\sum_y e^{\texttt{CLIP}(y,z;x)}}{\sum_y \sum_z e^{\texttt{CLIP}(y,z;x)}} = \operatorname*{argmax}_z \sum_y e^{\texttt{CLIP}(y,z;x)}, \tag{7}$$

where the second equality holds because $\sum_y \sum_z e^{\texttt{CLIP}(y,z;x)}$ is a constant of z. The corresponding textual description in $\texttt{CLIP}(y,z;x)$ is "*a photo of a {classname}, {description of z}*". Similarly, using the alternative approximation

Table 3: Inference accuracy (%) of two contextual attribute inference methods on ImageNet.

Attribute	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	77.31	66.01	60.00	80.61	88.79	59.26	74.26	58.94	67.16	86.56	78.23	93.95	68.71	73.83

yields $\text{argmax}_z\, p(z|x) = \text{argmax}_z\, \texttt{CLIP}(z;x)$, where the corresponding textual description in $\texttt{CLIP}(z;x)$ is "*a photo of an object, {description of z}*".

Experimental setting. We evaluate CLIP's ability to infer contextual attributes on ImageNet. Similar to the setting in Section 5.1, we apply certain transformations to randomly modify images and obtain the ground-truth attribute values for the modified images. For each image transformation, we randomly apply it to half of the images in ImageNet while keeping the other half unchanged. In this case, inferring each attribute is a binary classification task with a random guessing accuracy of 50%. We report the average accuracy over five runs.

Result. Table 3 shows the results of the two inference methods. The average accuracy of around 74% for both methods indicates that CLIP can reasonably infer the contextual attributes, with some attributes being easier to infer than others. This result suggests that we may bootstrap CLIP's inference by conditioning on the contextual attributes inferred by itself.

6 `PerceptionCLIP`: Emulating Perception with Contextual Attributes

Building on the observations in Section 5, we propose `PerceptionCLIP`, a two-step zero-shot classification method for CLIP. It emulates the human vision perception process by first inferring the contextual attributes and then inferring the class conditioning on the contextual attributes. We also show that these two steps can be simplified into one to recover the prompt ensembling method, but at the cost of disabling prior knowledge intervention and losing the benefit in interpretability. The pseudocode of `PerceptionCLIP` is outlined in Algorithm 1.

Step one: `PerceptionCLIP` estimates the distribution of contextual attributes, which is different from selecting the most probable single attribute value as discussed in Section 5. This is because CLIP can only infer contextual attributes in an imperfectly accurate manner, so we accommodate uncertainty by estimating the distribution.

In addition, we introduce a temperature hyperparameter τ to leverage our prior knowledge to intervene in the estimation of contextual attributes. A temperature τ greater than 1 smoothens CLIP's estimation of contextual attributes, implying that we do not fully trust CLIP's estimation and hope it to "acknowledge its uncertainty". The two-step nature also allows for other interventions, such as truncating top k predicted values for each contextual attribute (a.k.a., beam search), which we omit here for simplicity.

Algorithm 1: `PerceptionCLIP`

Require: class Y, contextual attributes $\{Z_1,...,Z_m\}$, `CLIP` score (with annotation function α), temperature hyperparameter τ

Input : image x

Output : predicted class $\hat{y}$

/* **Step 1:** infer contextual attributes */

$$\hat{p}(z_1,...,z_m|x) \leftarrow \frac{\sum_y e^{\texttt{CLIP}(y,z_1,...,z_m;x)/\tau}}{\sum_y \sum_{z_1,...,z_m} e^{\texttt{CLIP}(y,z_1,...,z_m;x)/\tau}}$$

/* **Step 2:** infer the class */

$$p(y|x,z_1,...,z_m) \leftarrow \frac{e^{\texttt{CLIP}(y,z_1,...,z_m;x)}}{\sum_y e^{\texttt{CLIP}(y,z_1,...,z_m;x)}}$$

$$\hat{y} \leftarrow \text{argmax}_y \sum_{z_1,...,z_m} p(y|x,z_1,...,z_m)\hat{p}(z|x).$$

Step two: `PerceptionCLIP` first approximates the class distribution conditioning on each combination of contextual attributes, and then uses the estimated distribution of contextual attributes to calculate the weighted sum of these class distributions. Finally, it selects the most probable y as the predicted output.

Computational complexity. With our current implementation, `PerceptionCLIP` has the same time complexity during inference as the basic single-template method (in terms of the number of encoder forward passes). Similar to the implementation of prompt ensembling, we pre-compute the embeddings of all class and contextual attributes combinations, and then use these pre-computed embeddings in each inference process. Since these computations are one-time, the time complexity during inference is unaffected by the number of contextual attributes. Compared to the basic method, which stores $O(|\mathcal{Y}|)$ embedding vectors, this implementation needs to store $O(|\mathcal{Y}| \times |\mathcal{Z}_1| \times \cdots \times |\mathcal{Z}_m|)$ embedding vectors. We will consider using beam search to reduce the space storage requirement in future work.

Simplifying into a single step. It can be seen from Algorithm 1 that setting the temperature to 1 and ignoring constant terms yields $\hat{y} \leftarrow \text{argmax}_y \sum_{z_1,\ldots,z_m} e^{\texttt{CLIP}(y,z_1,\ldots,z_m;x)}$, essentially simplifying the two-step algorithm into a single step. Intuitively, for each possible class, it sums the exponentiated `CLIP` scores calculated over each contextual attribute combination, resulting in an aggregated score for the class. Then, it selects the class with the highest aggregated score.

Single-step vs. prompt ensembling. This single-step method coincides with the prompt ensembling method if we aggregate over some randomly selected attributes instead of all contextual attribute combinations and modulo some implementation differences that have minimal impact on model performance. This coincidence potentially accounts for the effectiveness of prompt ensembling - it undergoes an implicit perception process. It also suggests that further reweighting these prompts for prompt ensembling is unnecessary, as it has already implicitly reweighted the prompts according to the inferred distributions of attributes. Nevertheless, our experimental results indicate that constructing diverse and systematic templates using our contextual attribute combinations is superior to ad-hoc template selections in prompt ensembling, especially when we can leverage prior knowledge to design specific candidate attributes tailored to the dataset.

Two-step vs. single-step. Compared to the two-step method, the one-step method is simpler to implement but has two drawbacks. First, it does not allow for human intervention in inferring contextual attributes. Our experiments indicate that CLIP does not always infer contextual attributes, whereas reasonable human intervention can leverage prior knowledge to further improve performance. Second, the one-step method prevents us from knowing the inferred contextual attributes, which could have improved the interpretability of the inference results.

Constructing candidate contextual attributes. At the core of `PerceptionCLIP` is the set of possible contextual attributes. To construct such a set, we propose two approaches: 1) We manually construct basic attributes that may cause spurious correlations. This is particularly effective when we have knowledge about the test dataset, as it allows for tailored construction. For instance, for satellite images, we choose attributes like geographical features and image source. 2) We leverage the in-context learning capability of large language models (LLMs) for semi-automated construction. To do this, we first use keywords to retrieve related captions in the LAION-400M dataset, then design instructions and provide in-context examples for GPT-4 (OpenAI, 2023) to summarize attributes from these captions. Appendix C.3 provides more details about this LLM-based approach.

7 Experiments

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

7.1 Zero-shot Generalization

We first evaluate the generalization of `PerceptionCLIP` using a single contextual attribute to show the effects of different attributes. Then, we extend the evaluation to multiple attributes. Finally, we show how `PerceptionCLIP` can benefit from interventions on the inferred attributes by leveraging prior knowledge.

Table 4: 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.

Attributes		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 attribute	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 attributes		68.49%	61.95%	77.64%	50.85%	48.18%
composition of top 3 attributes		**68.52%**	62.01%	77.92%	50.53%	48.39%
composition of top 4 attributes		68.50%	**62.24%**	**77.95%**	**50.97%**	**48.79%**

Table 5: Zero-shot classification accuracy of ViT-B/16 on different data domains with `PerceptionCLIP`.

	CUB200	EuroSAT	Places365	Flowers102	Food101	Oxford Pets
simple template	56.07	51.44	38.93	67.73	88.24	88.25
domain template	56.32	54.94	38.93	70.99	88.72	89.04
+ $\mathcal{Z}$	**57.08**	**59.23**	**40.27**	**72.84**	**89.14**	**90.37**

Settings. We first use GPT-4 to compile a set of possible contextual attributes, which systematically abstract the contextual attributes underlying the 80 hand-crafted templates (Radford et al., 2021) and include additional attributes such as orientation, background, and drawing tools. Detailed descriptions of each attribute and its potential values appear in Table 12, and implementation details appear in Appendix B.4. 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., 2021b), and ImageNet-Sketch (Wang et al., 2019). We also test on different data domains, including CUB200 (Wah et al., 2011), EuroSAT (Helber et al., 2019), Places365 (Zhou et al., 2017), Flowers102 (Nilsback and Zisserman, 2008), Food101 (Bossard et al., 2014), and Oxford Pets (Parkhi et al., 2012). We use ViT-B/16 as CLIP's image encoder unless otherwise stated.

Using a single attribute. Table 4 shows the result of `PerceptionCLIP` (two-step). The result of the one-step version is similar and deferred to Table 16 in the appendix. Compared to using the simple template "*a photo of a {class name}*", considering almost any single contextual attribute improves the results. However, the impact of different attributes varied. For example, considering only the *quality* attribute (with 9 possible values) significantly improves accuracy, surpassing prompt ensemble with 80 templates (Radford et al., 2021) on ImageNet-A. Moreover, the most influential contextual attributes vary 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 contextual attribute for image generation. This also indicates that `PerceptionCLIP` works the best when the considered contextual attributes cover the generation process of the dataset.

Using multiple attributes. The bottom section of Table 4 presents the results considering multiple contextual attributes. To this end, we simply concatenate the text descriptions of each attribute using commas (see Appendix B.4 for details). As the number of attributes considered increases, the classification accuracy gradually improves.

Table 6: Intervening in inferring contextual attributes using non-trivial temperature further improves zero-shot classification. Here we set temperature $\tau = 5$ and use the combination of three attributes.

	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%**

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

	RN50			ViT-B/32			ViT-B/16			ViT-L/14		
	Avg ↑	**Worst** ↑	**Gap** ↓	**Avg** ↑	**Worst** ↑	**Gap** ↓	**Avg** ↑	**Worst** ↑	**Gap** ↓	**Avg** ↑	**Worst** ↑	**Gap** ↓
CLIP	90.47	16.07	74.40	87.34	47.28	40.06	87.34	26.79	60.56	90.55	44.64	45.91
PerceptionCLIP ($\mathcal{Z}$={"*on land*", "*on water*"})	88.78	16.07	72.71	89.80	66.07	23.73	82.98	16.07	66.91	86.44	44.94	41.51
PerceptionCLIP ($\mathcal{Z}$={"*on land*", "*on water*", ...})	90.01	32.14	**57.87**	78.60	60.33	**18.28**	85.80	41.07	**44.73**	87.74	61.12	**26.62**

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

	RN50			ViT-B/32			ViT-B/16			ViT-L/14		
	Avg ↑	**Worst** ↑	**Gap** ↓	**Avg** ↑	**Worst** ↑	**Gap** ↓	**Avg** ↑	**Worst** ↑	**Gap** ↓	**Avg** ↑	**Worst** ↑	**Gap** ↓
CLIP	81.05	73.87	7.19	80.73	75.82	4.91	75.16	62.01	13.16	86.98	77.36	9.61
PerceptionCLIP ($\mathcal{Z}$={gender})	85.10	80.44	4.65	79.89	76.70	**3.19**	75.27	65.13	10.14	80.30	74.31	5.99
PerceptionCLIP ($\mathcal{Z}$={gender, age})	87.71	84.98	**2.74**	82.82	78.06	4.76	75.81	65.52	10.29	82.26	79.06	3.21
PerceptionCLIP ($\mathcal{Z}$={gender, age, race})	85.55	82.51	3.05	82.02	75.94	6.09	77.17	69.18	**7.99**	83.04	80.84	**2.20**

By combining three attributes, `PerceptionCLIP` outperforms prompt ensemble with 80 templates (Radford et al., 2021) on all datasets. We also test our method on different domains of data in Table 5. We first use domain templates provided in (Radford et al., 2021), which incorporate descriptions of the domain into the text prompt (e.g., *"a centered satellite photo of {classname}"*). The domain can be seen as a special contextual attribute, and specifying it improves the accuracy. We then take more contextual attributes into consideration, for instance, the cuisine and condition for Food101 (see details in Table 13). With `PerceptionCLIP`, we further improve the zero-shot classification accuracy.

Intervening in attributes inference. Table 6 (with temperature $\tau = 5$) shows that intervening in inferring contextual attributes using non-trivial temperature achieves modest but consistent performance gains across different datasets. We leave exploring other interventions such as top-k truncation (beam search) to future work.

7.2 Group Robustness and Interpretability

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 but do not causally determine the class, thus considered spurious features. When evaluating the worst group accuracy, we group the images based on their classes and group attributes, and evaluate the accuracy of each group as in (Sagawa* et al., 2020).

Tables 7 and 8 show the results on the two datasets. When the text prompts only describe the class, such as "*a photo of a {landbird/waterbird}.*" and "*a photo of a celebrity with {dark hair/blond hair}.*", CLIP 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 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

In this paper, we propose PerceptionCLIP, a zero-shot classification method for CLIP that emulates the human visual perception process. By doing class inference conditioned on self-inferred contextual attributes, it achieves improved generalization, less reliance on spurious features, and improved interpretability. Along the path of proposing PerceptionCLIP, we also systematically analyze the structure of CLIP prompts, and showcase CLIP's understanding of object attributes beyond common category features. Our work indicates that CLIP, as a model capable of communicating with humans via natural language, can achieve things that traditional models find challenging (such as conditional inference). Hence, it still has great potential in zero-shot classification and even broader tasks. Furthermore, this capability complements the study of neuroscience, enabling a better transition of the latter's research findings into practical use.

Limitations. One limitation of PerceptionCLIP is its sensitivity to text description perturbations: using different synonyms to describe the same attribute sometimes has non-trivial effects on the results. Although using more descriptions to describe an attribute value (Figure 2) alleviates this sensitivity, this issue is more intrinsic to CLIP and still persists. Future work may overcome this limitation by replacing CLIP with other vision-language models or improving CLIP's sensitivity to textual perturbations (e.g., through training-time text augmentation (Fan et al., 2023)). Another limitation of PerceptionCLIP is the need to design a set of contextual attributes. While this process provides a way to integrate human prior knowledge, it also requires additional effort, especially when we aim to cover many attributes. Currently, we use caption retrieval from the LAION-400M dataset and the in-context learning ability of large language models to semi-automate the construction process. In the future, our goal is to fully automate this process.

Acknowledgments

An, Zhu, and Huang are supported by National Science Foundation NSF-IIS-FAI program, DOD-ONR-Office of Naval Research, DOD Air Force Office of Scientific Research, DOD-DARPA-Defense Advanced Research Projects Agency Guaranteeing AI Robustness against Deception (GARD), Adobe, Capital One and JP Morgan faculty fellowships.

References

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

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13, pages 446–461. Springer, 2014.

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

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 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].

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor Guilherme Turrisi da Costa, Cees G. M. Snoek, Georgios Tzimiropoulos, and Brais Martinez. Bayesian prompt learning for image-language model generalization, 2023.

Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. arXiv preprint arXiv:2305.20088, 2023.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 12(7):2217–2226, 2019.

Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. arXiv preprint arXiv:1807.01697, 2018.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8340–8349, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15262–15271, 2021b.

Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. arXiv preprint arXiv:2204.03649, 2022.

Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, A James Hudspeth, Sarah Mack, et al. Principles of neural science, Fifth Edition, volume 4. McGraw-hill New York, 2013.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. (arXiv:2304.02643), Apr 2023. doi: 10.48550/arXiv.2304.02643. URL http://arxiv.org/abs/2304.02643. arXiv:2304.02643 [cs].

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In AAAI, volume 1, page 3, 2008.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. A tutorial on energy-based learning. Predicting structured data, 1(0), 2006.

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

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

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In International Conference on Machine Learning, pages 6781–6792. PMLR, 2021.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. (arXiv:2304.08485), Apr 2023. URL http://arxiv.org/abs/2304.08485. arXiv:2304.08485 [cs].

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In Proceedings of International Conference on Computer Vision (ICCV), December 2015.

Chengzhi Mao, Revant Teotia, Amrutha Sundar, Sachit Menon, Junfeng Yang, Xin Eric Wang, and Carl Vondrick. Doubly right object recognition: A why prompt for visual rationales. ArXiv, abs/2212.06202, 2022.

Cristina Menghini, Andrew Delworth, and Stephen H. Bach. 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].

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

M. Jehanzeb Mirza, Leonid Karlinsky, Wei Lin, Mateusz Kozinski, Horst Possegger, Rogerio Feris, and Horst Bischof. 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].

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In 2008 Sixth Indian conference on computer vision, graphics & image processing, pages 722–729. IEEE, 2008.

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

OpenAI. Gpt-4 technical report, 2023.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In 2012 IEEE conference on computer vision and pattern recognition, pages 3498–3505. IEEE, 2012.

Judea Pearl. Causal inference in statistics: An overview. 2009.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pages 8748–8763. PMLR, 2021.

Vikram V. Ramaswamy, Sing Yu Lin, Dora Zhao, Aaron B. Adcock, Laurens van der Maaten, Deepti Ghadiyaram, and Olga Russakovsky. 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].

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 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].

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In International conference on machine learning, pages 5389–5400. PMLR, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. page 10684–10695, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Karsten Roth, Jae Myung Kim, A. Sophia Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. Waffling around for performance: Visual classification with random words and broad concepts, 2023.

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114, 2021.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision, pages 618–626, 2017.

Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. arXiv preprint arXiv:2209.07511, 2022.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. Advances in Neural Information Processing Systems, 32, 2019.

Zifeng Wang, Zhenbang Wu, Dinesh Agarwal, and Jimeng Sun. Medclip: Contrastive learning from unpaired medical images and text. ArXiv, abs/2210.10163, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903, 2022.

Yu Yang, Besmira Nushi, Hamid Palangi, and Baharan Mirzasoleiman. Mitigating spurious correlations in multi-modal models during fine-tuning. ArXiv, abs/2304.03916, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. arXiv preprint arXiv:2305.10601, 2023.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=KRLUvxh8uaX.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence, 40(6):1452–1464, 2017.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16816–16825, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. International Journal of Computer Vision, 130(9):2337–2348, 2022b.

Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. arXiv preprint arXiv:2205.14865, 2022.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 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].

Appendix

A Image Caption Examples

Table 9 shows caption examples from dataset LAION-400M (Schuhmann et al., 2021). We can see that those captions not only describe class but also contextual attributes like color, style, and background.

Table 9: 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:*

B Experimental Details

B.1 Details on the Evaluation in Figure 3 and Table 2

Contextual attributes and their descriptions. In Figure 3, we show that by adding text descriptions of the ground-truth contextual attributes, the text embedding aligns better with the image embedding, resulting in higher similarity scores. By applying transformation functions (e.g., vertical flip) to all the ImageNet test images, we know the ground-truth contextual attribute of the images. Note that the last five attributes are tested directly on the ImageNet-C dataset (Hendrycks and Dietterich, 2018), which contains the same images as in the ImageNet test set while images are corrupted with certain transformations. We use the transformed image's image embedding to calculate the CLIP score. Table 10 shows the descriptions we used in this evaluation. When we ignore the contextual attribute, we use a simple template, "*a photo of a {classname}.*". When considering the contextual attribute, we test the cases using the correct attribute value (e.g., upside-down) and wrong attribute value (e.g., upright), respectively. We use three descriptions and average their embeddings as the text embedding to calculate the CLIP score.

Randomized descriptions. To validate the effectiveness of contextual attributes, we also compare with the cases where we use random descriptions. We replace every word in $\alpha(z^*)$ with random characters while keeping the word length unchanged. For example, $\alpha(y) \oplus \alpha(z_{random})$ of vertical flip contains three descriptions: "*a photo of a {y}.*", "*a photo of a {y}, iaYo5n0Dli7.*", "*a photo of a {y}, 8g2, Me5tx, q1, 6Ud2ole94Ir.*"

Additional results on the composition of contextual attributes. We perform the same evaluation while considering more than one contextual attribute. Table 11 shows the results. We draw the same conclusion that correct contextual attribute values lead to better image-text alignment and higher classification accuracy.

B.2 Details on the Evaluation in Table 3

In Table 3, we test whether CLIP can infer underlying contextual attributes by itself. In this experiment, we only apply transformations to half of the images from the ImageNet test set and use descriptions shown in Table 10. The task is to predict the correct contextual attribute's value, which is a binary classification task. For example, half images are upright, and half images are upside-down, and the goal is to classify the orientation of the images by CLIP. We evaluate two methods with two approximations of $p(z|x)$ in Table 1. Note that we do not intervene in the inference in this experiment.

Table 10: Summary of descriptions for different attributes used in Figure 3, Table 2 and Table 3. z^* denotes the correct value of the contextual attribute, and z_{wrong} denotes the wrong value of the contextual attribute. Ideally, each attribute has a distribution of text descriptions. Here, we use three descriptions and use the averaged text embeddings of them to calculate the CLIP score.

Attribute	$\alpha(y) \oplus \alpha(z^*)$	$\alpha(y) \oplus \alpha(z_{wrong})$
vertical flip	"a photo of a {y}." "a photo of a {y}, upside-down." "a photo of a {y}, the photo is upside-down."	"a photo of a {y}." "a photo of a {y}, upright." "a photo of a {y}, the photo is upright."
90° rotation	"a photo of a {y}." "a photo of a {y}, rotated." "a photo of a {y}, the photo is rotated."	"a photo of a {y}." "a photo of a {y}, upright." "a photo of a {y}, the photo is upright."
elastic-transform	"a photo of a {y}." "a photo of a {y}, with distortion." "a photo of a {y}, the photo is distorted."	"a photo of a {y}." "a photo of a {y}, normal." "a photo of a {y}, the photo is normal."
color-invert	"a photo of a {y}." "a photo of a {y}, color-inverted." "a photo of a {y}, the photo is color-inverted."	"a photo of a {y}." "a photo of a {y}, normal." "a photo of a {y}, the photo is normal."
solarize	"a photo of a {y}." "a photo of a {y}, solarized." "a photo of a {y}, the photo is solarized."	"a photo of a {y}." "a photo of a {y}, normal." "a photo of a {y}, the photo is normal."
blur	"a photo of a {y}." "a photo of a {y}, blurred." "a photo of a {y}, the photo is blurred."	"a photo of a {y}." "a photo of a {y}, clear." "a photo of a {y}, the photo is clear."
grayscale	"a photo of a {y}." "a photo of a {y}, grayscale." "a photo of a {y}, the photo is in black and white."	"a photo of a {y}." "a photo of a {y}, colorful." "a photo of a {y}, the photo is colorful."
bright	"a photo of a {y}." "a photo of a {y}, bright." "a photo of a {y}, the photo is bright."	"a photo of a {y}." "a photo of a {y}, dark." "a photo of a {y}, the photo is dark."
noise	"a photo of a {y}." "a photo of a {y}, with noise." "a photo of a {y}, the photo has noise."	"a photo of a {y}." "a photo of a {y}, clear." "a photo of a {y}, the photo is clear."
snow	"a photo of a {y}." "a photo of a {y}, in the snow." "a photo of a {y}, the photo is in the snow."	"a photo of a {y}." "a photo of a {y}, clear." "a photo of a {y}, the photo is clear."
frost	"a photo of a {y}." "a photo of a {y}, in the frost." "a photo of a {y}, the photo is in the frost."	"a photo of a {y}." "a photo of a {y}, clear." "a photo of a {y}, the photo is clear."
fog	"a photo of a {y}." "a photo of a {y}, in the fog." "a photo of a {y}, the photo is in the fog."	"a photo of a {y}." "a photo of a {y}, clear." "a photo of a {y}, the photo is clear."
jpeg	"a photo of a {y}." "a photo of a {y}, in jpeg format." "a photo of a {y}, the photo is in jpeg format."	"a photo of a {y}." "a photo of a {y}, in high resolution." "a photo of a {y}, the photo is in high resolution."

B.3 Details on the Visualization

To generate the Grad-CAM for an input sample, first, we use a function that computes CLIP's similarity score for each class and applies softmax on top of these values. Then, we consider the restriction of the function to the output class that we want to visualize and compute the gradient for a specific layer. All the Grad-CAMs from Figure 1, 4, 5 and 6 are computed using the layer before the final attention block of the ViT-L/14 model as suggested in a popular explainability library.[1]

[1] https://github.com/jacobgil/pytorch-grad-cam

Table 11: Similarity score and classification accuracy on ImageNet test set. We apply a composition of two transformation functions on images, and use the composition of attributes' descriptions for text.

Attributes	**Similarity (CLIP score $\times$ 100)**			
	w/o z	w/ random z	w/ wrong z	w/ correct z
vertical flip + color-invert	25.39	28.23 (↑2.84)	26.28 (↑0.88)	30.26 (↑**4.86**)
grayscale + elastic-transform	26.66	30.55 (↑3.89)	26.48 (↓0.19)	32.15 (↑**5.49**)
Attributes	**Accuracy (%)**			
	w/o z	w/ random z	w/ wrong z	w/ correct z
vertical flip + color-invert	19.44	20.88 (↑1.44)	20.01 (↑0.57)	21.32 (↑**1.88**)
grayscale + elastic-transform	29.79	30.49 (↑0.70)	30.14 (↑0.35)	30.59 (↑**0.80**)

Table 12: Summary of contextual attributes and their value descriptions used in ImageNet-related datasets.

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

To compute the ratio between the usage of core and spurious regions in prediction, we *(1)* run Segment Anything (Kirillov et al., 2023) and select a segmentation mask for the core region of each image (e.g., the bird or the leopard), then consider the rest (non-core) regions as spurious; *(2)* use the masks to identify the core and spurious pixel values from the Grad-CAMs and compute the mean pixel value for both of these regions; *(3)* normalize the numbers and show them as two percentages in a bar plot for each Grad-CAMs.

B.4 Details on the Experiments in Section 7

In Table 4, we test `PerceptionCLIP` on ImageNet and its OOD datasets. We first use GPT-4 to summarize the contextual attributes involved in the 80 hand-crafted templates (Radford et al., 2021), then add three contextual attributes (orientation, background, tool) to the testing bed. Table 12 shows the text descriptions for every contextual attribute. When considering a single attribute, we use a main template, "*a photo of a {class name}*" and concatenate it with each attribute value. For example, for *illumination*, we will have three templates: "*a photo of a {class name}.*", "*a photo of a {class name}, bright.*", "*a photo of a {class name}, dark.*". For simplicity, we only use one description for every attribute value instead of having a distribution of descriptions in all the experiments in Section 7. When considering the composition of attributes, we generate combinations from the values across all attributes. For example, the composition of *orientation* and *illumination* results in 3x3=9 templates and one of them is "*a photo of a {class name}, upside-down, bright.*" We concatenate descriptions of each attribute without taking care of the order. Such simple concatenation of descriptions works well, probably because the pre-trained CLIP model behaves like a bag-of-words (Yuksekgonul et al., 2023). Future works could explore better ways of composing text prompts.

Table 13: Datasets, domain templates and contextual attributes used in Table 5

Dataset	Domain Template	Attributes
CUB200	"a photo of a {classname}, a type of bird"	size, background, condition
EuroSAT	"a centered satellite photo of {classname}"	condition, source
Places365	"a photo of a {classname}"	background, quality, condition
Flowers102	"a photo of a {classname}, a type of flower"	background, illumination, quality, condition
Food101	"a photo of a {classname}, a type of food"	cuisines, condition
Oxford Pets	"a photo of a {classname}, a type of pet"	species, background, pose, interaction

Table 14: Domain templates, contextual attributes and their descriptions used in Table 7 and Table 8

Dataset	Domain Template	Attributes	Value Descriptions
Waterbirds	"a photo of a {classname}"	simple background	"on land", "on water"
		background	+ "in forest", "in sky", "on street", "on grass", "on tree", "with flowers", "on beach", "with human", "on a branch"
CelebA	"a photo of a celebrity with {classname}"	gender	"female", "male"
		age	"young", "old"
		race	"white skin", "dark skin", "asian"

Table 13 and 14 list the contextual attributes used in Table 5, 7 and 8. Attributes and their descriptions are manually designed based on our priors of datasets. For `PerceptionCLIP`, we use the two-step inference method and use Eq. 7 to estimate $p(z|x)$. Experiments on Waterbirds and CelebA are conducted on their training set without intervention.

C Additional Results

C.1 More Results

Table 16 shows the results of using one-step inference. The conclusion is the same as using two-step inference. Incorporating contextual attributes improves zero-shot classification accuracy and composing more attributes yields greater improvements.

Table 15 compares the two-step inference (i.e., w/ intervention) with the one-step inference (i.e., w/o intervention) when using 80 templates. The intervention is conducted by adjusting the temperature. With intervention, we improve the performance for free since we do nothing about the templates or data.

Table 15: Comparing two-step inference (inference w/ intervention) with one-step inference (inference w/o intervention). All the results are tested with ViT-B/16 CLIP model using 80 templates.

	W/o intervention	W/ intervention	
		w/ y	w/o y
ImageNet	68.32%	68.50%	**68.56%**
ImageNetV2	61.93%	**62.24%**	**62.24%**
ImageNet-R	77.71%	**77.83%**	77.53%
ImageNet-A	49.95%	**50.37%**	50.35%
ImageNet-Sketch	**48.26%**	48.18%	48.22%

Table 16: Zero-shot classification accuracy on five datasets. We use ViT-B/16 CLIP model and the one-step inference method. The first part shows the results used in (Radford et al., 2021). The second part shows the result of considering one contextual attribute. The highest performance is in bold for each dataset, and the next two highest values are underlined. The third part shows the result of using compositions of contextual attributes. The best performance is highlighted in bold. Results show that considering only one attribute at inference time can achieve comparable performance as using 80 templates and composing factors further improves performance.

Attributes		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	67.48%	**61.60%**	75.75%	48.99%	47.12%
	illumination	66.94%	60.90%	75.01%	48.16%	46.23%
	orientation	67.32%	60.99%	74.50%	48.28%	46.84%
	quality	67.99%	61.45%	76.34%	**50.11%**	47.50%
	quantity	67.71%	61.50%	75.38%	50.01%	46.73%
	perspective	67.60%	61.20%	74.92%	49.21%	46.91%
	art	67.33%	61.04%	**77.15%**	49.48%	**48.05%**
	medium	67.19%	60.59%	76.38%	49.17%	46.75%
	condition	**68.20%**	61.37%	75.50%	49.16%	47.22%
	color-scheme	66.50%	60.49%	73.78%	47.83%	46.61%
	tool	66.68%	60.58%	75.40%	48.37%	47.38%
composition of top 2 factors		**68.35%**	61.67%	77.88%	**50.83%**	48.11%
composition of top 3 factors		68.26%	61.89%	77.65%	50.42%	48.41%
composition of top 4 factors		68.22%	**61.98%**	**78.04%**	50.49%	**48.72%**

C.2 More Visualizations

We show more visualizations in Figure 5 and 6. Figure 5 shows images from the ImageNet dataset with the ground-truth class *leopard*. Figure 6 shows images from the Waterbirds dataset with the ground-truth class *waterbird*. Grad-CAMs show that CLIP relies more on core features when conditioned on the correct contextual attributes (e.g., background) for classification. The reliance on core features also improves model interpretability.

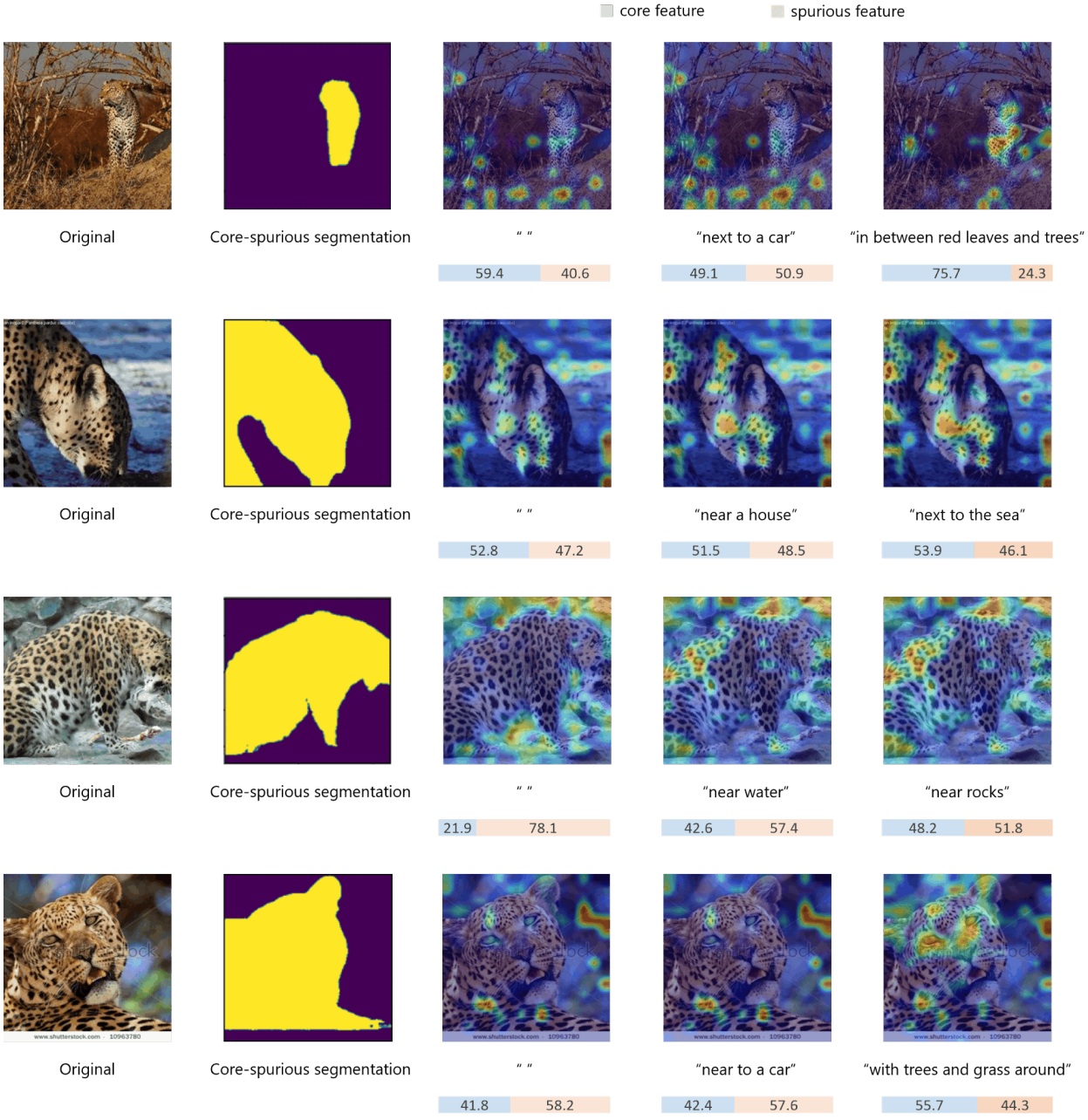

Figure 5: Leopard images from ImageNet dataset. Visualization of the original image, the regions of core and spurious features, and the Grad-CAMs obtained using no, incorrect, and ground-truth contextual attributes.

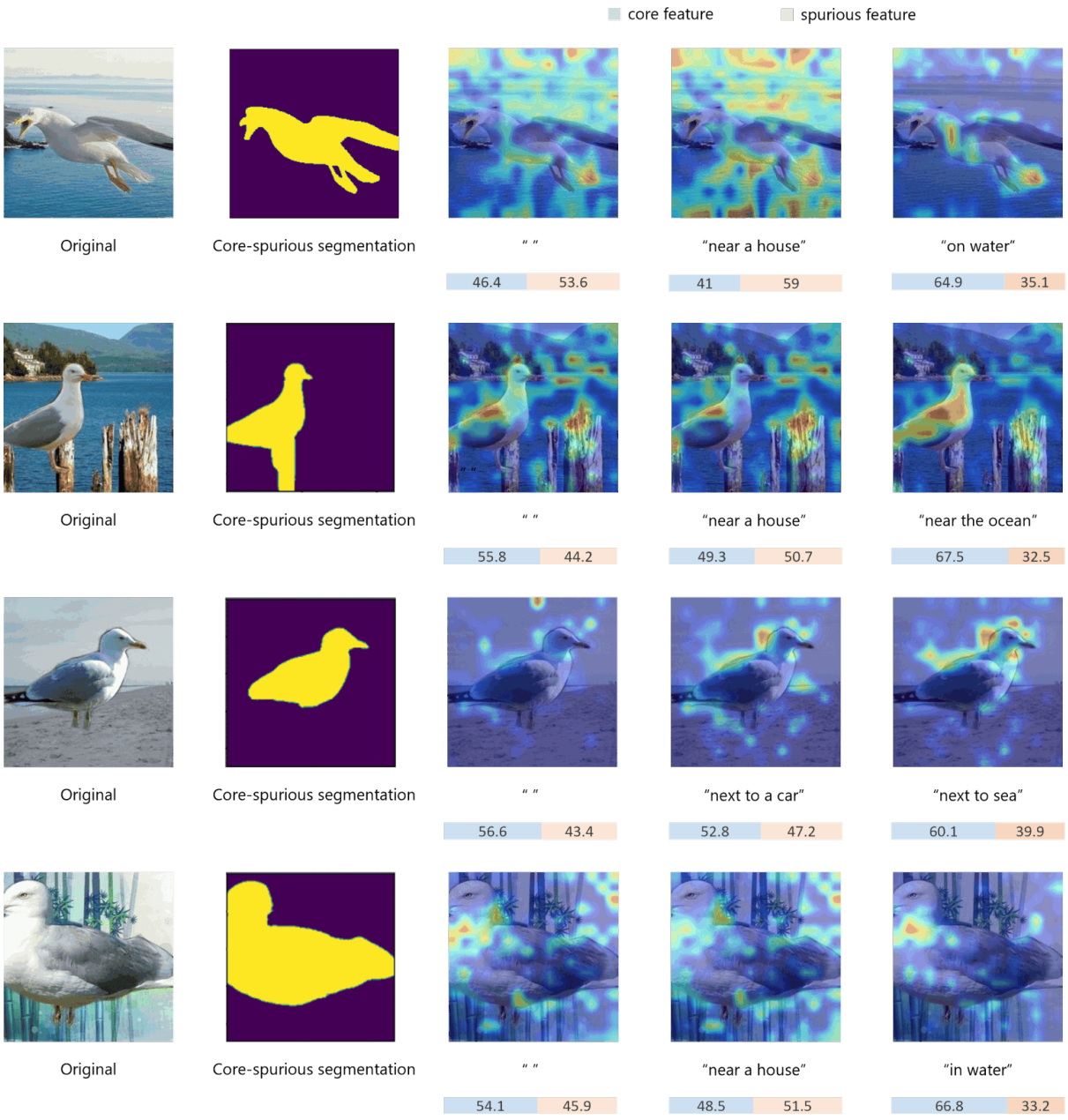

Figure 6: Waterbird images from Waterbirds dataset. Visualization of the original image, the regions of core and spurious features, and the Grad-CAMs obtained using no, incorrect, and ground-truth contextual attributes.

C.3 Discovering Contextual Attributes by LLMs

In this section, we provide an example of how to use GPT-4 (OpenAI, 2023) to generate contextual attributes and their descriptions automatically. We take EuroSAT dataset as an example. There are three steps:

1. Given a dataset (e.g., EuroSAT) with a specific domain, we retrieve similar images (e.g., satellite images) from a large image+text dataset LAION-400M [2].
2. We crawl the captions and randomly sample a limited number of these captions (e.g., 200).
3. We provide GPT-4 with the captions and the information of the dataset, and ask it to extract contextual attributes using Prompt 1.

Table 17 shows the contextual attributes discovered by GPT from captions. Adding those attributes to the domain template, we improve the accuracy from 51.44% to 59.20% (with intervention $\tau = 5$), which is comparable to manually designed ones. However, we found that the attributes identified by GPT are not always appropriate, possibly because of the gap between the retrieved images and our dataset. Future work could involve using image-based searches to find more similar images rather than relying on language-based searches.

Prompt 1: An example prompt for discovering contextual attributes and their descriptions from example captions for EuroSAT dataset.

```
You are a helpful assistant who helps me summarize my text captions. I have a dataset of
     image-caption pairs, where each caption briefly describes the image. I need to
    extract some attributes from these textual descriptions that contribute to the data
    generation process of the image.

For example, from the three descriptions ["A black dog on the green grass", "A red car
    on the road in a bright environment", "A white refrigerator"], you can summarize the
     "background", "color", "illumination" as three attributes, with possible values ["
    grass field", "road", " "], ["black", "green", "red", "white", " "], ["bright", "
    dark", " "] respectively. Note that they each have an empty value because the human
    annotators may choose not to mention them in the captions.

Note:
1. The number of potential values for each factor should not exceed 3, so you should
    extract the most representative values.
2. You should summarize at most 5 attributes, covering the most representative
    attributes in the provided captions.
3. I have a list of labels for these images, namely ['annual crop land', 'forest', '
    brushland or shrubland', 'highway or road', 'industrial buildings or commercial
    buildings', 'pasture land', 'permanent crop land', 'residential buildings or homes
    or apartments', 'river', 'lake or sea',]. The attributes you summarized should not
    overlap with the concepts of these labels, and the values you summarized should not
    include any of these labels. For example, since "river" is in my label set, your
    summarized values should not include "river" for any attributes.
4. The set of all values for all attributes you summarized should not overlap.

I need your summary to have the following format:

summerized_factors = {
    "background": [
        "",
        "grass",
        "road",
    ],

    "color": [
```

[2] https://github.com/rom1504/clip-retrieval

```
        "black",
        "green",
        "red",
        "white",
    ],
    "illumination": [
        "bright",
        "dark",
        ""
    ]
}

Here are the captions:

//200 captions
```

Table 17: Contextual attributes and their value descriptions for EuroSAT generated by GPT-4.

Attributes	**Value Descriptions**
source	"", "Yandex satellite", "NASA", "Google Maps"
geographical feature	"", "island", "ul.", "street"
image type	"", "satellite", "aerial", "map"
natural phenomenon	"", "hurricane", "earthquake", "deforestation"
structure type	"", "residential", "commercial", "fortress"