Visual-Text Cross Alignment: Refining the Similarity Score in Vision-Language Models

Jinhao Li [1] **Haopeng Li** [1] **Sarah M. Erfani** [1] **Lei Feng** [2] **James Bailey** [1] **Feng Liu** [1]

Abstract

It has recently been discovered that using a pre-trained *vision-language model* (VLM), e.g., CLIP, to align a whole query image with several finer text descriptions generated by a large language model can significantly enhance zero-shot performance. However, in this paper, we empirically find that the finer descriptions tend to align more effectively with *local areas of the query image* rather than the whole image, and then we theoretically validate this finding. Thus, we present a method called *weighted visual-text cross alignment* (WCA). This method begins with a *localized visual prompting* technique, designed to identify local visual areas within the query image. The local visual areas are then *cross-aligned* with the finer descriptions by creating a similarity matrix using the pre-trained VLM. To determine how well a query image aligns with each category, we develop a score function based on the weighted similarities in this matrix. Extensive experiments demonstrate that our method significantly improves zero-shot performance across various datasets, achieving results that are even comparable to few-shot learning methods. The code is available at github.com/tmlr-group/WCA.

1. Introduction

Following the significant advancements of large-scale pre-training in natural language processing (Devlin et al., 2018; Radford et al., 2018; 2019; Brown et al., 2020), the CLIP model (Radford et al., 2021) scales up its pre-training data through *aligning* images and the corresponding natural language captions in the shared latent space, which achieves remarkable performance in zero-shot classification. Despite its achievements, CLIP's performance exhibits notable sensitivity to the prompts used during the inference stage (Radford et al., 2021; Zhou et al., 2022b). For example, Zhou et al. (2022b) has highlighted that changing the prompt from "a photo of [CLASS]" to "a photo of a [CLASS]" can lead to a performance boost of 6%. Crafting effective prompts is crucial but it requires significant time, effort, and domain-specific knowledge (Zhou et al., 2022b), making it challenging to deploy such models in practical applications.

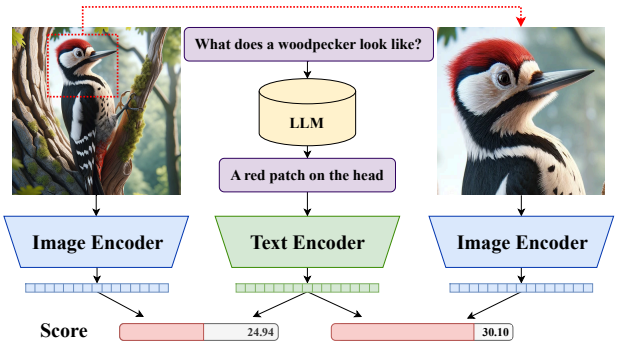


Figure 1. **Aligning an entire image with a detailed text description results in lower scaled cosine similarity**, as shown on the left. Aligning the description with a specific image part, such as the detailed red patch (on the right), increases the score.

To address the above issue, a promising solution is to use *large language models* (LLMs) to generate several finer text descriptions of each category (Menon & Vondrick, 2022; Pratt et al., 2023). This strategy helps reduce the manual effort in creating prompts and, more importantly, does not necessitate additional turning, thereby more easily preserving models' generalization abilities. The ability to generalize is a crucial issue in prompt-learning methods (Li et al., 2022b; Wang et al., 2022; Wu et al., 2023; Tanwisuth et al., 2023) as these methods tend to overfit training data. LLM-based visual-text alignment emphasizes *global matching*, namely, text descriptions are aligned with the whole image.

However, in this paper, we find:

Finer-grained text descriptions may align more accurately with the specific area of an image but not necessarily with the image as a whole.

[1]School of Computing and Information Systems, University of Melbourne, Australia. [2]Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore. Correspondence to: Feng Liu <fengliu.ml@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

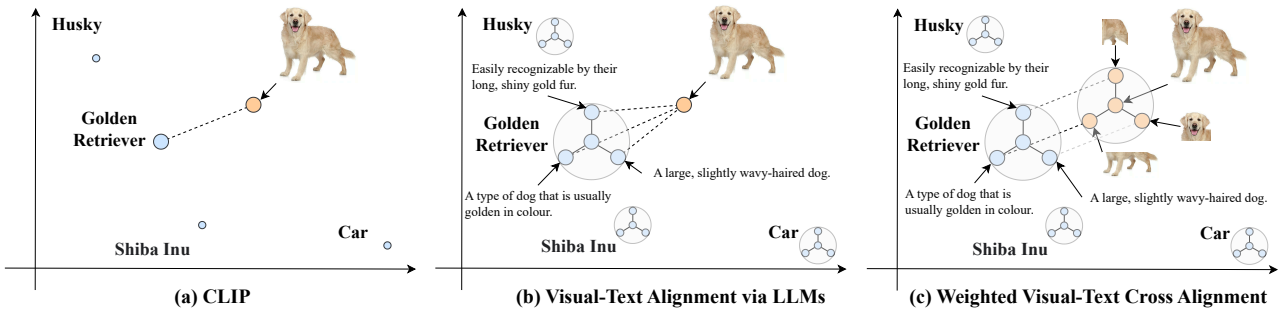


Figure 2. **We show different zero-shot visual-text alignment methods**: (a) CLIP, (b) Visual-Text Alignment via LLMs (Menon & Vondrick, 2022; Pratt et al., 2023), and (c) Weighted Visual-Text Cross Alignment (ours). Unlike (a) and (b), (c) utilizes a localized visual prompting technique to enhance alignment by ensuring that detailed descriptions match precisely with specific areas of the visual content.

This is because finer descriptions often contain detailed and finer-grained visual concepts, such as "a woodpecker has a straight and pointed bill", may not align precisely with an entire image, as demonstrated in Figure 1. In expanding this, it becomes clear that the complexity of images often contains a myriad of details that a description might capture only in part. These finer elements, while crucial, might lead to a misalignment when the objective is to correlate the entire image with its description. Additionally, we provide a theoretical analysis in Section 4 to gain a deeper understanding of the issue. This discrepancy suggests a need for more nuanced approaches in visual-text alignment, where the focus is not just on the whole query image (i.e., the global alignment) but also on recognizing and aligning different areas within the query image.

To this end, we propose a method called ***w**eighted visual-text **c**ross **a**lignment* (WCA). This method concentrates on local and regional visual areas within the query image that can align better with the fine-grained text descriptions of each category. This can be achieved through *localized visual prompting*, where the image is prompted to focus on localized visual elements, such as via cropping. These visual elements are then *cross-aligned* with their corresponding detailed text descriptions, leading to a similarity matrix. Furthermore, we introduce a score function based on the *weighted similarities* in the matrix and use the score to see how well a query image is aligned with each category.

A key feature of WCA is its consideration of the varying importance of localized visual elements and text descriptions within the similarity matrix, ensuring that each similarity value in this matrix contributes differently to the overall similarity aggregation. The importance of a specific localized visual element is quantified by its cosine similarity with the query image, where a high score suggests that the element captures the main semantic content of the query image. Similarly, the relevance of a text description can be estimated by its similarity to the category label it corresponds to. Therefore, our method achieves accurate visual-text alignment scores efficiently without additional models or training, making it a highly efficient approach. We demonstrate the distinction between WCA and the current methods of visual-text alignment in Figure 2. Our empirical results show that WCA significantly enhances zero-shot performance across various datasets, even comparable with few-shot methods.

To summarize, our main contributions are outlined as follows: (i) We have identified and conducted a theoretical analysis of the issue where aligning an entire image with finer text descriptions results in suboptimal performance. (ii) We introduce a method, WCA, that performs *weighted cross alignment* between the finer descriptions and local visual areas using localized visual prompting. (iii) Our extensive experiments validate our theoretical hypothesis and demonstrate the efficacy of our method, significantly surpassing the state-of-the-art methods without the need for extra models or data. (iv) We offer insights into the key factors that contribute to the effectiveness of our method.

2. Related Work

Vision-language models. Vision-language models (VLMs) pre-trained on large-scale data have shown efficacy in enhancing representation learning capabilities (Cho et al., 2021; Kim et al., 2021; Xue et al., 2021; Li et al., 2021; Wang et al., 2021; Li et al., 2023). CLIP (Radford et al., 2021) underwent training on a corpus of 400 million paired images and texts, exhibiting robust transferable ability and exceptional zero-shot performance. In a similar vein, the introduction of ALIGN (Jia et al., 2021) demonstrates that despite being pre-trained on datasets containing image-text pairs with considerable noise, the scale of the training corpus can compensate for this noise and is capable of learning superior representations. Subsequent works, including FLAVA (Singh et al., 2022), Florence (Yuan et al., 2021), BLIP (Li et al., 2022a), and so on, have continued to advance this paradigm, contributing further to the field.

Textual prompting in vision-language models. Despite

CLIP exhibiting superior zero-shot capabilities, the effectiveness of its application in downstream tasks is significantly influenced by the choice of prompts, as noted by Radford et al. (2021) and Zhou et al. (2022b). Zhou et al. (2022b) highlight that selecting the optimal prompt is complex and time-intensive, often requiring prompt tuning. To address this, Menon & Vondrick (2022) and Pratt et al. (2023) leverage the knowledge embedded in LLMs, such as GPT-3 (Brown et al., 2020) for the automatic generation of class-specific descriptions. These descriptions, particularly focusing on the discriminating features of image categories, are then aligned with the query image. This method has been shown to be effective as it enriches the textual representation by incorporating LLMs. Roth et al. (2023) examine this phenomenon and introduce the WaffleCLIP framework, which replaces LLM-generated descriptions with random character and word descriptions, eliminating the need to query LLMs and offering a cost-effective alternative. However, our work diverges from these approaches as they do not engage in visual prompting techniques.

Visual prompting in vision-language models. In contrast to text-based prompting in VLMs, visual prompting aims to process the visual input accordingly. Yao et al. (2024) color image regions and utilize a captioning model to identify objects based on color predictions. Bahng et al. (2022) experiment with learning the image perturbation, keeping the model parameters unchanged. These approaches, along with the studies (Jia et al., 2022; Tu et al., 2023) require at least a few samples from downstream tasks. The RedCircle introduced by Shtedritski et al. (2023) suggests that highlighting an object with a red circle can direct the model's attention to that region, but it requires manual annotation. Yang et al. (2024) propose FGVP, which uses an extra model SAM (Kirillov et al., 2023), to identify objects first and then employ *Blur Reverse Masks* to enhance the semantic localization capability of areas around the objects, reducing the need for manual annotation but adding complexity. Our method differs from these as we do not need downstream data, manual annotation, or additional models.

Test time prompt tuning in vision-language models. While fine-tuning prompts can adapt pre-trained VLMs to specific downstream tasks, this approach requires labeled training data, which can be costly and unavailable for zero-shot tasks. Test-time prompt tuning (TPT), as introduced by Shu et al. (2022), addresses this issue by learning adaptive prompts for individual test samples through the generation of multiple randomly augmented views. The goal is to optimize text prompts in an unsupervised manner. However, naive augmentation methods may lead to overly simplistic variations in test data. To address this, Feng et al. (2023) proposed DiffTPT, which uses diffusion models to augment test samples with richer visual appearance variations. In contrast, our method, while also applied during testing, does not involve the same tuning processes as TPT and DiffTPT. Instead, it leverages the strengths of pre-trained VLMs in a different manner, potentially offering a more efficient approach. Our method avoids the need for extensive data augmentation and fine-tuning procedures, which are typically required by TPT and DiffTPT to enhance their performance. By directly utilizing the inherent capabilities of pre-trained VLMs, our approach simplifies the alignment process.

3. Problem Setting and Preliminaries

In this section, we introduce the problem setting and the preliminaries considered in this paper.

Problem setting. Let $\mathcal{X}$ be an image space and $\mathcal{Y}$ be a label space, where $\mathcal{Y}$ is a set of words or phrase, e.g., $\mathcal{Y} = \{\text{car}, \dots, \text{bicycle}\}$. Considering a pre-trained VLM, let $f\colon \mathcal{X} \to \mathbb{R}^d$ be its image encoder and $g\colon \mathcal{Y} \to \mathbb{R}^d$ be its text encoder. These encoders are designed to transform input images and texts into a shared embedding space of dimension d. In this paper, $\boldsymbol{x}$ represents an arbitrary image from $\mathcal{X}$, and $\boldsymbol{y}$ denotes an arbitrary label from $\mathcal{Y}$.

The aim in *zero-shot visual classification* is to label images into predefined classes based on their visual content, without updating the parameters of the pre-trained model. We will introduce two representative methodologies to address the zero-shot visual classification problem in the following.

CLIP zero-shot transfer (Radford et al., 2021). The core idea is to devise a scoring function $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that can assess the semantic matching between the given image and a set of corresponding labels. $s(\cdot)$ is computed based on the cosine similarity of their hidden representations. The scoring function is mathematically expressed as:

$$s(\boldsymbol{x}, \boldsymbol{y}|f, g) = \cos(f(\boldsymbol{x}), g(\boldsymbol{y})), \tag{1}$$

where a higher score implies a closer semantic match between $\boldsymbol{x}$ and $\boldsymbol{y}$. Therefore, the predicted label for image $\boldsymbol{x}$ is the label $\boldsymbol{y}^*$, which has the highest cosine similarity score with $\boldsymbol{x}$ among all possible labels from $\mathcal{Y}$.

Enhancing zero-shot transfer using LLMs (Pratt et al., 2023; Menon & Vondrick, 2022). Given a label $\boldsymbol{y} \in \mathcal{Y}$, an LLM model $h(\cdot)$ can be utilized to generate rich and descriptive text that encapsulates the characteristics and details of the category $\boldsymbol{y}$. The descriptions are as follows:

$$h(\boldsymbol{y}) = \{\boldsymbol{y}_j\}_{j=1}^{M}, \tag{2}$$

where M represents the total number of generated descriptions. In this case, the scoring function s is calculated as the average of similarity scores between $\boldsymbol{x}$ and each text description $\boldsymbol{y}_j$. This can be mathematically represented as:

$$s_{\text{LLM}}(\boldsymbol{x}, \boldsymbol{y}|f, g) = \frac{1}{M} \sum\nolimits_{j=1}^{M} s(\boldsymbol{x}, \boldsymbol{y}_j|f, g). \tag{3}$$

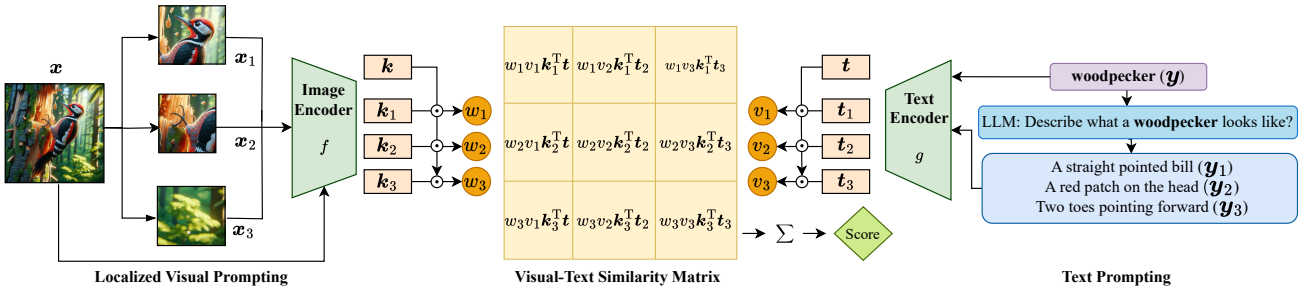


Figure 3. **Overview of weighted visual-text cross alignment (WCA).** The process begins with *localized visual prompting*, where the input image $\boldsymbol{x}$ is divided into localized patches, such as $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$. These patches are encoded by an image encoder to produce visual features. The *text prompting* stage utilizes a large language model to generate detailed textual descriptions $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3\}$ for a given class label $\boldsymbol{y}$ (e.g., "woodpecker"). The WCA calculates alignment scores between visual features and textual features, using patch weights $\{w_1, w_2, w_3\}$ and text weights $\{v_1, v_2, v_3\}$. The final score is computed by summing the visual-text similarity matrix.

4. Motivation from Theoretical Justification

In Section 1, we have empirically shown that aligning a whole image with finer text descriptions might cause a lower similarity compared to aligning an area of the image with the finer description (see Figure 1). In this section, we gain a deeper understanding of the issue and provide a theoretical analysis regarding the issue mentioned in Figure 1.

For simplicity, we assume that the image encoder f is linear functional and satisfies the condition[1] $\boldsymbol{x} \neq \mathbf{0} \Rightarrow f(\boldsymbol{x}) \neq \mathbf{0}$. We focus on cosine similarity as we investigate CLIP-like models. Specifically, we have the following theorem (For the complete proof, please refer to Appendix A).

Theorem 4.1. *Let $\boldsymbol{x}$ represent an image along with its corresponding ground truth label $\boldsymbol{y}$. $\boldsymbol{x}$ can be partitioned into two components $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, where $\boldsymbol{x} = \boldsymbol{x}_1 + \boldsymbol{x}_2$. Assume $\boldsymbol{x}_1$ is a discriminative region that is perfectly correlated with $\boldsymbol{y}$ as $\cos(f(\boldsymbol{x}_1), g(\boldsymbol{y})) = 1$, and non-discriminative region $\boldsymbol{x}_2$ has an imperfect correlation to $\boldsymbol{y}$ denoted as $\cos(f(\boldsymbol{x}_2), g(\boldsymbol{y})) < 1$. If $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ satisfy linear independence[2], then we have $\cos(f(\boldsymbol{x}), g(\boldsymbol{y})) < 1$.*

This highlights a possible limitation in the current methodology of visual-text alignment, where encoding the entire image content might lead to a less-than-ideal performance. Therefore, it becomes essential to accurately retrieve $\boldsymbol{x}_1$, ensuring its semantic content is perfectly correlated with $\boldsymbol{y}$. A simple method to tackle this issue is to choose the highest cosine similarity score, expressed as $\max(\cos(f(\boldsymbol{x}_1), g(\boldsymbol{y})), \cos(f(\boldsymbol{x}_2), g(\boldsymbol{y})))$, to determine the most accurate alignment. However, this approach might not always be feasible, particularly if $\boldsymbol{x}_1$ shows a perfect similarity score to an incorrectly matched $\boldsymbol{y}$, leading to potential errors, which is validated in Table 7. Motivated by this, we propose WCA to address this issue.

[1]This means that when $\boldsymbol{x}$ represents a non-black image, its representation is generally not a zero vector, which is a relatively weak assumption.

[2]It means that there is not a constant c such that $\boldsymbol{x}_1 = c\boldsymbol{x}_2$. Intuitively, for an image of a cat in a garden, where $\boldsymbol{x}_1$ represents the cat and $\boldsymbol{x}_2$ denotes the garden. The information about the cat (like its shape, color, etc.) is exclusive to $\boldsymbol{x}_1$, and the information about the garden (like plants, sky, etc.) is exclusive to $\boldsymbol{x}_2$. $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are linearly independent as neither can be represented by the other. Therefore, this assumption is relatively weak.

5. Visual-text Cross Alignment

In this section, we formally introduce our proposed method WCA, where the overall pipeline is shown in Figure 3. Specifically, we start by describing localized visual prompting and then discuss how to perform the weighted cross alignment. Finally, we show the overall algorithm.

Localized visual prompting. As described previously, matching an entire image with finer text descriptions could result in a lower similarity score compared to aligning a specific area of the image with the finer description. To tackle this issue, we propose *localized visual prompting*, which seeks to segment an image into multiple areas, each holding critical semantic content. The objective of this method is to enhance the extraction of semantic information from images by focusing on specific regions rather than the entire image.

This can be achieved through a localized visual prompting function $p(\cdot)$. Given an image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times 3}$, where H and W are its height and width respectively, the function $p(\cdot)$ can be described as follows:

$$p(\boldsymbol{x}) = \{\boldsymbol{x}_i = \phi(\boldsymbol{x}, \gamma_i \min(W, H)) \mid i = 1, ..., N\}, \quad (4)$$

where γ_i is a random variable sampled from a uniform distribution $U(\alpha, \beta)$. α and β are predefined parameters that set the lower and upper bound. The function $\phi(\cdot)$ crops the image at a random location, with the second argument specifying the size of the output. The random nature of γ_i ensures that the cropping is varied, covering different parts of the image, thus retrieving the different semantic information from the various regions. These localized image

patches are then *cross-aligned* with finer text descriptions.

Cross alignment. Upon obtaining the set of localized image patches $p(\boldsymbol{x})$ and the set of text descriptions $h(\boldsymbol{y})$, it is essential to evaluate the similarities between them, a process referred to as *cross alignment*. This process results in a matrix defined as:

$$\begin{bmatrix} s(\boldsymbol{x}_1, \boldsymbol{y}_1) & \cdots & s(\boldsymbol{x}_1, \boldsymbol{y}_M) \\ \vdots & \ddots & \vdots \\ s(\boldsymbol{x}_N, \boldsymbol{y}_1) & \cdots & s(\boldsymbol{x}_N, \boldsymbol{y}_M) \end{bmatrix}, \tag{5}$$

where each column's entries represent the similarity scores between the j-th text description and every image patch, while the entries in each row denote the scores of the i-th image patch and all text descriptions. A naive approach to aggregate this matrix is by averaging all scores as follows,

$$s_{\text{AVG}}(\boldsymbol{x}, \boldsymbol{y}|f, g) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} s(\boldsymbol{x}_i, \boldsymbol{y}_j|f, g). \tag{6}$$

The issue with Eq. (6) lies in treating each image patch or description *equally* in the calculation of the final similarity score. However, Menon & Vondrick (2022) demonstrated that with such scoring LLMs can result in suboptimal outcomes. For instance (as shown in Figure 4), labels like "jackfruit" receive text descriptions related to taste and smell, irrelevant to visual cues, which is less useful in this case. Similarly, the way we prompt images has the same issue, where $p(\cdot)$ is likely to generate unexpected output, such as the areas containing only background information or task-unrelated objects as demonstrated in Figure 7. This motivates us to develop a method to select reliable localized image patches and text descriptions. So this leaves us with another challenge: *how to select reliable $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ in the set of localized image patches $p(\boldsymbol{x})$ and the set of text descriptions $h(\boldsymbol{y})$, respectively.*

Weighted aggregation for cross alignment. Considering how the semantic relevance between an image and a text is measured using cosine similarity, the question arises: *could this approach also be applied to image-to-image or text-to-text pairs to assess their relevance?* Our empirical studies in Section 6 supports the idea. Consequently, we introduce the set of weights for image patches, denoted as $\mathcal{W} = \{w_i\}_{i=1}^{N}$, and for text descriptions, referred to as $\mathcal{V} = \{v_j\}_{j=1}^{M}$. These weights adjust the contribution of each entry in the similar matrix to the similarity score aggregation as follows:

$$w_i = \frac{\exp\left(s(\boldsymbol{x}, \boldsymbol{x}_i|f, g)\right)}{\sum_{l=1}^{N} \exp\left(s(\boldsymbol{x}, \boldsymbol{x}_l|f, g)\right)}, \tag{7}$$

$$v_j = \frac{\exp\left(s(\boldsymbol{y}, \boldsymbol{y}_j|f, g)\right)}{\sum_{l=1}^{M} \exp\left(s(\boldsymbol{y}, \boldsymbol{y}_l|f, g)\right)}. \tag{8}$$

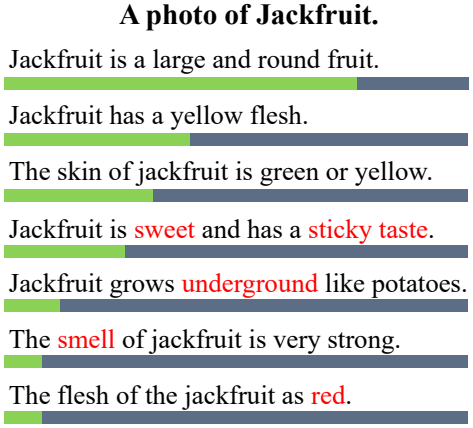


Figure 4. **Overview of text description weights for Jackfruit**. This figure illustrates various text description weights based on their relevance to the visual cue "A photo of Jackfruit." The green lines represent the relative similarity scores, indicating how closely each description aligns with the visual cue. Longer green lines denote higher relevance, while shorter lines indicate lower relevance. Descriptions deemed irrelevant or incorrect are highlighted in red.

Then, the score function for the WCA is defined as:

$$s_{\text{WCA}}(\boldsymbol{x}, \boldsymbol{y}|f, g) = \sum_{i=1}^{N} \sum_{j=1}^{M} w_i v_j s(\boldsymbol{x}_i, \boldsymbol{y}_j|f, g). \tag{9}$$

The underlying idea here is twofold: (i) a high value of w_i indicates that $\boldsymbol{x}_i$ is crucial in representing the primary semantic content of image $\boldsymbol{x}$, and (ii) a high value of v_j suggests a strong correlation between $\boldsymbol{y}_j$ and the label $\boldsymbol{y}$. Essentially, higher weights are indicative of the relative importance of specific image patches or text descriptions within their respective contexts, which is visually demonstrated in Section 6. Finally, s_{WCA} can aggregate a more accurate and reliable score.

Overall algorithm. Algorithm 1 demonstrates how to predict the best match label from $\mathcal{Y}$ for a query image $\boldsymbol{x}$. The algorithm first prompts the image $\boldsymbol{x}$ into multiple localized regions $\{\boldsymbol{x}_i\}_{i=1}^{N}$, and assigns a weight w_i to each $\boldsymbol{x}_i$. Similarly, for each label $\boldsymbol{y}^k \in \mathcal{Y}$, $h(\cdot)$ is used to generate $\boldsymbol{y}^k$-related descriptions $\{\boldsymbol{y}_j^k\}_{j=1}^{M}$, and v_j is assigned to each $\boldsymbol{y}_j^k$. The core of the algorithm is the calculation of a cross-alignment score s_{WCA}^k. Finally, the algorithm selects the label $\boldsymbol{y}^{k^*}$ that maximizes this cross-alignment score, indicating it as the most suitable label for the image $\boldsymbol{x}$.

6. Experiments

In this section, we evaluate the performance of our method by a series of experiments and various ablation studies. A detailed insight into our method is also provided.

Datasets. First, we evaluate our method on zero-shot vi-

Table 1. **Comparison of zero-shot visual classification performance (accuracy in %) across different image classification benchmarks using three different CLIP models (B/32, B/16, L/14)**. The standard deviation (σ) of WCA 's performance is listed, along with the improvement (Δ) highlighted in green over the top-performing baseline, which is shown as underlined.

Method	ImageNet			CUB			Oxford Pets			DTD			Food101			Place365		
	B/32	B/16	L/14	B/32	B/16	L/14	B/32	B/16	L/14	B/32	B/16	L/14	B/32	B/16	L/14	B/32	B/16	L/14
CLIP	62.05	66.74	73.48	51.21	56.01	62.12	85.04	88.14	93.24	42.93	42.98	52.61	82.60	88.40	92.55	38.51	39.27	39.63
CLIP-E	63.37	68.37	75.52	52.74	56.16	62.53	87.38	89.10	93.62	43.83	45.27	55.43	83.93	88.83	93.07	39.28	40.30	40.55
CLIP-D	63.01	68.04	75.03	52.69	57.08	63.26	84.46	87.52	93.30	44.20	46.17	55.05	84.12	88.85	93.03	39.90	40.34	40.55
Waffle	63.30	68.12	75.31	52.04	56.89	62.27	85.50	86.51	91.55	42.98	44.68	54.31	83.98	89.06	93.33	39.47	40.76	40.89
CuPL	64.37	69.61	76.62	49.76	56.42	62.15	87.03	91.14	94.33	47.50	50.53	60.59	84.20	88.98	93.37	39.08	39.83	40.77
WCA	**66.84**	**71.08**	**77.32**	**56.91**	**59.78**	**65.24**	**89.89**	**92.23**	**94.66**	**49.39**	**52.79**	**61.78**	**86.40**	**90.01**	**93.96**	**40.66**	**41.43**	**42.23**
σ	0.07	0.05	0.03	0.17	0.15	0.12	0.09	0.10	0.09	0.16	0.17	0.16	0.05	0.04	0.04	0.05	0.05	0.03
Δ	**+2.47**	**+1.47**	**+0.70**	**+4.17**	**+2.70**	**+1.98**	**+2.51**	**+1.09**	**+0.33**	**+1.89**	**+2.26**	**+1.19**	**+2.20**	**+0.95**	**+0.59**	**+0.76**	**+0.67**	**+1.34**

Table 2. **Comparison on natural distribution shifts with accuracy (%) reported**. TP, VP, TTP, and LLM represent textual prompting, visual prompting, test-time promoting, and large language models, respectively. The term "Tuned" refers to whether the model is fine-tuned on ImageNet. "Source" refers to in-distribution performance, while "Target" represents out-of-distribution performance.

Method	Prompts	Tuned?	Source	Target				Average
			ImageNet	ImageNet-V2	ImageNet-R	ImageNet-S	ImageNet-A	
CoOp (Zhou et al., 2022b)	TP	✓	71.51	64.20	75.21	47.99	49.71	61.72
CoCoOp (Zhou et al., 2022a)	VP+TP	✓	71.02	64.07	76.18	48.75	50.63	62.13
UPT (Zang et al., 2022)	VP+TP	✓	**72.63**	64.35	76.24	48.66	50.66	62.51
ProGrad (Zhu et al., 2023)	TP	✓	72.24	64.73	74.58	47.99	49.39	61.79
KgCoOp (Yao et al., 2023)	TP	✓	71.20	64.10	76.70	48.97	50.69	62.33
TPT (Shu et al., 2022)	TTP	✓	69.70	64.30	73.90	46.40	53.67	61.59
DiffTPT (Feng et al., 2023)	TTP	✓	70.30	**65.10**	75.00	46.80	55.68	62.58
CLIP (Radford et al., 2021)	Hand-crafted	✗	66.74	60.83	73.96	46.15	47.77	59.09
CLIP-E (Radford et al., 2021)	Hand-crafted	✗	68.37	61.90	77.40	47.87	49.00	60.91
CuPL (Pratt et al., 2023)	LLM-TP	✗	69.61	63.27	77.10	48.80	50.77	61.91
WCA (Ours)	LLM-TP+VP	✗	71.08	64.71	**78.06**	**50.18**	**56.13**	**64.03**

sual classification benchmarks outlined in (Menon & Vondrick, 2022): (i) ImageNet (Deng et al., 2009) for recognizing everyday objects; (ii) CUB for fine-grained classification of birds (Welinder et al., 2010); (iii) Oxford Pets (Parkhi et al., 2012) for common animals; (iv) DTD (Cimpoi et al., 2014) for in-the-wild patterns; (v) Food101 (Bossard et al., 2014) specifically designed for food classification; and (vi) Place365 (Zhou et al., 2017) for scene recognition.

Then we evaluate our method on domain generalization benchmarks in (Radford et al., 2021), including: (i) ImageNet-V2 (Recht et al., 2019) to evaluate distribution shift from ImageNet; (ii) ImageNet-Sketch (Wang et al., 2019) consisting of black and white sketch images; (iii) ImageNet-A (Hendrycks et al., 2021b) for naturally occurring images that are adversarial examples; and (iv) ImageNet-R (Hendrycks et al., 2021a) for focusing on art, cartoons, graffiti, and other renditions. Each dataset represents a unique distribution shift from ImageNet. This benchmark evaluates the model's robustness in natural distribution shifts.

Baselines. In the context of zero-shot visual classification, our evaluation includes a comparison with the following baselines: (i) CLIP (Radford et al., 2021), an approach utilizing a manually created template: "A photo of {class}; (ii) An ensemble version of CLIP (CLIP-E) (Radford et al., 2021) employing a variety of manually crafted templates; (iii) CLIP-D (Menon & Vondrick, 2022) leveraging LLMs for the description generation; (iv) CuPL (Pratt et al., 2023) known for generating higher quality LLM descriptions in comparison to CLIP-D; and (v) Waffle (Roth et al., 2023), a unique approach that replaces LLM-generated descriptions with randomly generated character and word descriptions.

Furthermore, we employ the following methods for comparison in domain generalization benchmarks: CoOp (Zhou et al., 2022b), CoCoOp (Zhou et al., 2022a), UPT (Zang et al., 2022), ProGrad (Zhu et al., 2023), KgCoOp (Yao et al., 2023) and MaPLe (Khattak et al., 2023). Also, we compare our method to test-time prompting methods, such as TPT (Shu et al., 2022) and DiffTPT (Feng et al., 2023). Notably, these methods require model fine-tuning, whereas our method operates without any tuning.

Implementation details. We employ a range of VLMs, including CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021), GroupViT (Xu et al., 2022) and AltCLIP (Chen et al., 2022). Unless specified otherwise, our experiments are conducted using CLIP[3] with a backbone of ViT-B/32. All experiments are performed on an NVIDIA A100 GPU. Our

[3] https://github.com/openai/CLIP

Algorithm 1 Weighted Visual-Text Cross Alignment

input A query image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times 3}$; a label set $\mathcal{Y} = \{\boldsymbol{y}^k\}_{k=1}^K$; an LLM model $h(\cdot)$; the number of crops N; the number of text prompts M; the lower and upper bound α, β; the crop function ϕ.

for $i = 1$ to N **do**
  Sample $\gamma_i \sim U(\alpha, \beta)$
  Let $n_i = \gamma_i \times \min(W, H)$
  Obtain $\boldsymbol{x}_i = \phi(\boldsymbol{x}, n_i)$
  Compute w_i according to Eq. (7)
end for
for $k = 1$ to K **do**
  Prompt $h(\boldsymbol{y}^k) = \{\boldsymbol{y}_j^k\}_{j=1}^M$
  for $j = 1$ to M **do**
    Compute v_j according to Eq. (8)
  end for
  Obtain s_{WCA}^k according to Eq. (9)
end for
$k^* = \text{argmax}_{k\in[1..K]}\, s_{\text{WCA}}^k$

output $\boldsymbol{y}^{k^*}$

method incorporates two key parameters: the crop lower and upper bound (α, β) and the number of crops (N). We evaluated various β values and observed that larger β yields better results as demonstrated in Table 8 (in Appendix). Thus we set $\beta = 0.9$. In addition, other parameters are generally set to $\alpha = 0.5$, $N = 60$ and $M = 50$ across all experiments. These values were chosen by our empirical analysis in Figure 6. To optimize computational efficiency, we adopt a strategy where the embedding of an image is pre-computed and stored. This embedding is derived from a weighted average of the embeddings of its localized image patches, a method detailed in Appendices C.1 and C.2. This approach guarantees that computational costs do not increase over time, as the embedding is only computed once, which is similar to the technique used in CLIP (Radford et al., 2021) for managing the expenses with prompt ensembling.

The descriptions used in our study are derived from prior works (Menon & Vondrick, 2022; Pratt et al., 2023), which have made progress in automating description generation with minimal human involvement. These works guide LLMs to produce descriptions efficiently by using carefully designed prompts. For example, a prompt from (Menon & Vondrick, 2022) asks the model to identify useful features for distinguishing a specific category in a photo. The models then output the visual features associated with the specified category, and these outputs are stored for later use. Additionally, Menon & Vondrick (2022); Pratt et al. (2023) have made certain files containing pre-generated outputs available as open-source resources, serving as the default approach when implementation details are unspecified. Examples using the PaLM model are in Appendix E.

Zero-shot visual classification results. Table 1 showcases the zero-shot visual classification performance comparison

Table 3. **Ablation study on ImageNet**. Top-1 accuracy (%) is reported here. The **bold** value indicates the highest accuracy in each column. The first row serves as the baseline. Δ shows the mean improvement on top-1 accuracy compared the baseline.

$p(\cdot)$	$h(\cdot)$	$\mathcal{W}$	$\mathcal{V}$	ImageNet			Δ
				B/32	B/16	L/14	
Baseline				63.35	68.36	75.52	–
	✓			64.36	69.61	76.63	+1.12
	✓		✓	64.77	70.09	76.68	+1.44
✓				64.76	68.76	75.53	+0.61
✓		✓		65.44	69.50	76.23	+1.31
✓	✓			65.51	69.72	76.34	+1.45
✓	✓	✓	✓	**66.66**	**71.03**	**77.33**	**+2.60**

across different image classification benchmarks and different model sizes. The results underscore the consistent superiority of WCA over established baselines. Furthermore, the results in the table highlight an intriguing trend: the smaller-sized models exhibit more significant performance enhancements compared to their larger counterparts. This phenomenon suggests that while larger models like CLIP are known for their robustness, the relatively smaller models inherently possess more room for improvement, allowing WCA to yield substantial gains in accuracy. Moreover, our method excels notably in tasks where CLIP models struggle, indicating that these particularly challenging tasks offer substantial potential for improvement. This observation reemphasizes the fact that our method excels in addressing complex tasks where there's ample room for advancement, potentially leading to substantial performance gains in domains that pose greater challenges for existing models. Additionally, we show a case where WCA makes a correct prediction and its decision is explained by its descriptions in Figure 5 and more examples can be found in Appendix F.

Domain generalization results. Table 2 presents a comparison of various methods on their performance across different natural distribution shifts of the ImageNet dataset. Our method surpasses others in in-distribution (Imagenet) performance, except for UPT, and does so without needing fine-tuning data. For out-of-distribution datasets, it excels except on ImageNet-V2, achieving the highest average score, notably on ImageNet-S and ImageNet-A. This demonstrates that our method is comparable to state-of-the-art in-distribution performance and significantly surpasses them in out-of-distribution scenarios.

Ablation study. The ablation study presented in Table 3 systematically shows how various elements impact the top-1 accuracy across three model sizes (B/32, B/16, L/14) for ImageNet classification. The inclusion of each element, either alone or combined, demonstrates differing levels of influence on the model's performance. Notably, when all components are combined, there is a significant improvement in accuracy, indicated by bold values, which show

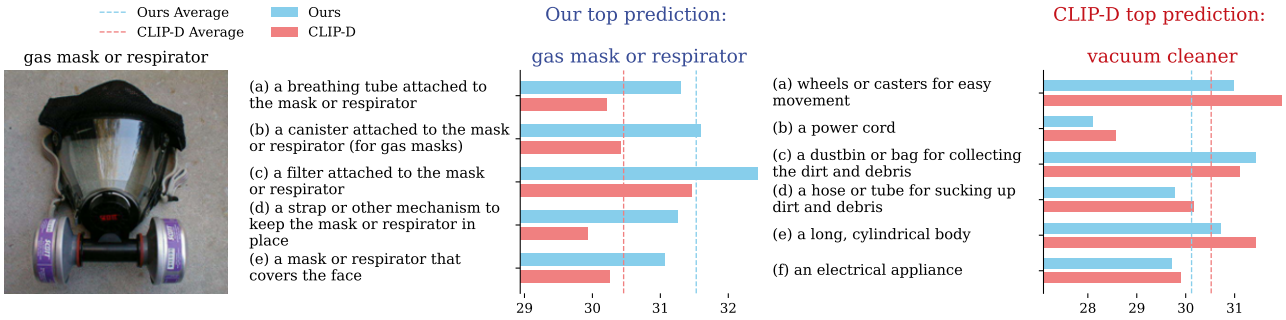


Figure 5. **We demonstrate the prediction and explanation of our methods and CLIP-D** (Menon & Vondrick, 2022), in identifying and explaining a given image of a gas mask or respirator. The image is analyzed to predict its category, with the scaled cosine similarity scores between the image and various descriptions plotted for each method.

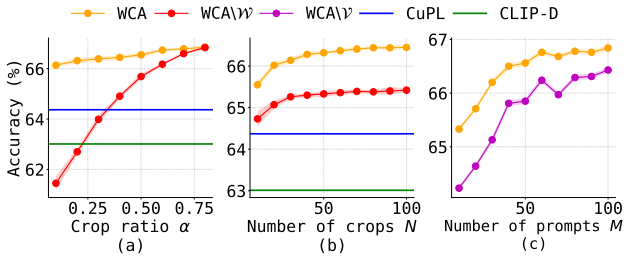


Figure 6. **Sensitivity analysis of crop ratio α, number of crops N and number of prompts M**. The shading around curves represents the standard deviation. Note that WCA$\setminus\mathcal{W}$ and WCA$\setminus\mathcal{V}$ represent the WCA without patch weights $\mathcal{W}$ and text weights $\mathcal{V}$, respectively. CuPL (Pratt et al., 2023) and CLIP-D (Menon & Vondrick, 2022) represent comparative baselines.

the highest accuracy across all model sizes. Specifically, including only $p(\cdot)$ can enhance the performance solely for the B/32 model or potentially worsen it compared to the baseline for B/16 and L/14. This is because the randomness in generating $p(\cdot)$ could still negatively impact the model's performance. However, integrating reweighting parameters $\mathcal{W}$ improves performance consistently by ensuring the selection of only reliable patches. Moreover, the inclusion of $h(\cdot)$ demonstrates an average improvement of 1.12% with the support of LLMs, explaining the observed enhancement. In contrast, weighted cropping, involving both $h(\cdot)$ and $\mathcal{W}$, shows a larger improvement of 1.45% compared to the inclusion of $h(\cdot)$ alone. This emphasizes the efficiency of our method, even without relying on other models or data.

Sensitivity analysis. Figure 6.(a) shows how accuracy correlates with the lower bound parameter α. It indicates that increasing α generally leads to a higher accuracy. The line labeled WCA$\setminus\mathcal{W}$ suggests that a small α might result in accuracy falling below the baseline. However, the line WCA demonstrates that α has a minimal impact on performance, suggesting that the factor $\mathcal{W}$ reduces the sensitivity of α. Figure 6.(b) illustrates the effect of the number of image crops N on accuracy. It shows that increasing the number of crops has a positive impact on accuracy. As can be seen, the performance reaches a plateau at $N = 60$. Our observations from Figure 6.(c) suggest that increasing the number of descriptions generally leads to improved performance, up to a certain threshold. However, when incorporating text description weights $\mathcal{V}$ in WCA, we notice that the performance tends to converge faster compared to the scenario without $\mathcal{V}$. Additionally, for a small number of prompts, not using weights results in lower performance compared to the baseline. This underscores the significance of incorporating text description weights, as they play a crucial role in enhancing the overall performance.

Revise failure in text descriptions via weighting. Previous studies (Menon & Vondrick, 2022) have pointed out several limitations in how LLMs generate descriptions. These models occasionally produce descriptions that are non-visual features. For example, as shown in Figure 4, when GPT-3 (Brown et al., 2020) describes a jackfruit, it mentions descriptions associated with taste and smell, which are not part of the visual features. While these descriptions are accurate, they present difficulties for VLMs, which are primarily designed to process and align visual elements. Non-visual descriptions are less useful in this context as they cannot be visually recognized, especially in the scenario where these models come across categories they have never seen before.

Recent studies (Chen et al., 2023; Bielawski et al., 2022; Zhang et al., 2022) have shown that the CLIP model outperforms text-only trained models, such as Bert, in terms of visual understanding. CLIP's advanced visual perception enables it to associate text with corresponding visuals in a way that mirrors human perception. Consequently, we leverage VLMs' visual perception strengths to overcome the shortcomings of LLMs in generating descriptions. As illustrated in Figure 4 (with additional examples in Figure 11), our method successfully identifies these non-visual features and faulty examples, as evidenced by their low similarity scores. This demonstrates the effectiveness of our method

Table 4. **Comparison of time costs between CLIP and WCA methods for different numbers of patches (N) in seconds**. The table includes the time for cropping and preprocessing, encoding, and the total time for both methods. It highlights the additional time required by WCA compared to CLIP as the number of patches increases.

Process Step	CLIP	N									
		10	20	30	40	50	60	70	80	90	100
Crop+Preprocess	0.0032	0.0195	0.0394	0.0615	0.0861	0.1096	0.1314	0.1572	0.1811	0.2036	0.2032
Encoding	0.0049	0.0049	0.0050	0.0052	0.0061	0.0068	0.0080	0.0084	0.0083	0.0085	0.0086
Total	0.0081	0.0244	0.0444	0.0666	0.0923	0.1164	0.1394	0.1656	0.1894	0.2121	0.2117

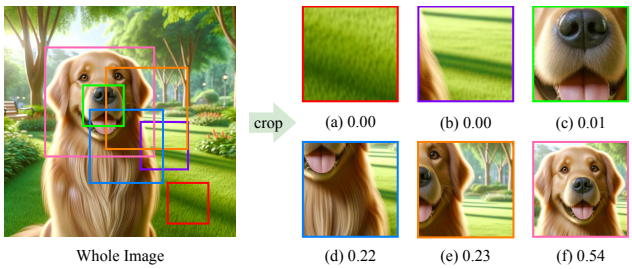


Figure 7. **Similarity scores between the query image and its localized image patches**. We show that image-patch cosine similarity can filter those patches with less semantic information.

in filtering out non-visual or incorrect descriptions, thereby enhancing the accuracy and reliability of the description generation process in alignment with visual content.

Explanation of image patch weights. Here we explore the efficacy of using image weights. Our chosen technique for visual prompting is random cropping. While this method is straightforward, it inherently carries the risk of randomness, which could negatively impact performance. The goal is to ensure that the random cropping process captures discriminative regions of the image that can then be effectively cross-aligned with textual descriptions.

To achieve this, we employ a specific method, as outlined in Eq. (7). This equation is designed to evaluate the information degree of different regions within an image. A high score in this context signifies that a particular region of the image is rich in informative content. In Figure 7, we present a series of examples to illustrate this concept. Images (a), (b), and (c) in the figure, for instance, receive scores close to zero. This indicates that the cropped patches from these images contain regions that are not particularly important or informative. In contrast, images (d), (e), and (f) demonstrate a different scenario. These images successfully highlight various characteristic parts of a golden retriever, such as the face, head, and body. The patch weights in these cases are significantly higher, reflecting the discriminative value of these regions. As a result, these images are identified as more informative and thus more suitable for effective cross-alignment with textual descriptions. This approach showcases the potential of image patch weights in enhancing the selection and accuracy of image processing, particularly in aligning images with relevant textual information.

Time cost. We break down the time cost of cropping N patches from an image and obtaining their feature embeddings compared to using CLIP. Our experimental dataset consists of 1 000 images selected from ImageNet, with the results averaged across these images. As shown in Figure 4, for CLIP, encoding a single image with the CLIP image encoder takes approximately 0.0081 seconds, with 0.0032 seconds for preprocessing and 0.0049 seconds for encoding. Our method details the time required to crop one image into N patches and encode these patches using the CLIP image encoder through batch processing. The majority of the computational overhead compared to CLIP is attributed to "Cropping + Preprocessing." In contrast, "Encoding" time is mitigated by batch processing the patch images. To optimize computational efficiency and address increased inference time, we pre-compute and store the embeddings of each processed image. Once these embeddings are computed and stored, the inference time is unaffected by "Cropping + Preprocessing" and "Encoding." We then only need to perform a dot product between image embeddings and text embeddings, similar to CLIP, which is very fast and takes less than 10 seconds for 50 000 images.

Additional experiments and analysis. Additional experiments are detailed in Appendix B, including various visual prompting techniques, aggregating strategies, applying WCA with various VLMs, and WCA with ResNet backbone. Further analysis can be found in Appendix C, such as visualization of prompt image embedding. Our discussion on limitations is presented in Appendix D.

7. Conclusion

We introduce a method WCA, which capitalizes on the precise alignment between localized image areas and finer textual descriptions generated by LLMs, using pre-trained VLMs. By empirically and theoretically demonstrating that finer descriptions align more closely with local image regions, we significantly enhance zero-shot classification performance. Our comprehensive experiments show that WCA not only surpasses traditional zero-shot benchmarks but also competes closely with few-shot learning techniques.

Acknowledgements

Jinhao Li and Haopeng Li are supported by the Melbourne Research Scholarship. Feng Liu receives support from the Australian Research Council with grant numbers DP230101540 and DE240101089, and the NSF&CSIRO Responsible AI program with grant number 2303037. Sarah Erfani is in part supported by Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA) DE220100680.

We extend our gratitude to the anonymous reviewers and our colleagues for their insightful comments and suggestions, which greatly improved the quality of this paper. Furthermore, we acknowledge the support provided by the University of Melbourne's Research Computing Services and the Petascale Campus Initiative.

Impact Statement

This paper presents work to advance the field of VLMs by enhancing zero-shot performance. Our findings reveal that finer text descriptions align more effectively with localized areas of an image rather than the entire image. This discovery has significant implications for applying VLMs in various real-world scenarios. By improving the zero-shot performance, our method can benefit industries relying on large-scale image data analysis, such as enhancing visual search engines, improving automated image tagging systems, and advancing medical imaging diagnostics.

Societal impacts of our work include democratizing access to powerful AI tools, as our method can deliver high performance even with limited or no labeled data, making advanced VLMs more accessible and usable in resource-constrained environments. Additionally, our approach contributes to the ethical practice of AI by reducing the need for extensive data labeling and minimizing the computational resources required for model training and adaptation, aligning with goals of sustainability and efficiency.

References

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Bahng, H., Jahanian, A., Sankaranarayanan, S., and Isola, P. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.

Bielawski, R., Devillers, B., Van de Cruys, T., and VanRullen, R. When does clip generalize better than unimodal models? when judging human-centric concepts. In *ACL*, 2022.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101–mining discriminative components with random forests. In *ECCV*, 2014.

Boyd, S. and Vandenberghe, L. *Introduction to applied linear algebra: vectors, matrices, and least squares*. Cambridge university press, 2018.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *NeurIPS*, 2020.

Chen, Z., Liu, G., Zhang, B.-W., Ye, F., Yang, Q., and Wu, L. Altclip: Altering the language encoder in clip for extended language capabilities. In *ACL (Findings)*, 2022.

Chen, Z., Chen, G. H., Diao, S., Wan, X., and Wang, B. On the difference of bert-style and clip-style text encoders. In *ACL (Findings)*, 2023.

Cho, J., Lei, J., Tan, H., and Bansal, M. Unifying vision-and-language tasks via text generation. In *ICML*, 2021.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *CVPR*, 2014.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Feng, C.-M., Yu, K., Liu, Y., Khan, S., and Zuo, W. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *ICCV*, 2023.

Helber, P., Bischke, B., Dengel, A., and Borth, D. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS*. IEEE, 2018.

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 *ICCV*, 2021a.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. In *CVPR*, 2021b.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. Visual prompt tuning. In *ECCV*, 2022.

Khattak, M. U., Rasheed, H., Maaz, M., Khan, S., and Khan, F. S. Maple: Multi-modal prompt learning. In *CVPR*, 2023.

Kim, W., Son, B., and Kim, I. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. In *ICCV*, 2023.

Li, H., Ke, Q., Gong, M., and Drummond, T. Progressive video summarization via multimodal self-supervised learning. In *WACV*, 2023.

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. C. H. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022a.

Li, W., Huang, X., Zhu, Z., Tang, Y., Li, X., Zhou, J., and Lu, J. Ordinalclip: Learning rank prompts for language-guided ordinal regression. In *NeurIPS*, 2022b.

Menon, S. and Vondrick, C. Visual classification via description from large language models. In *ICLR*, 2022.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. In *CVPR*, 2012.

Pratt, S., Covert, I., Liu, R., and Farhadi, A. What does a platypus look like? generating customized prompts for zero-shot image classification. In *ICCV*, 2023.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. OpenAI, 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

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 *ICML*, 2021.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

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

Shtedritski, A., Rupprecht, C., and Vedaldi, A. What does clip know about a red circle? visual prompt engineering for vlms. In *ICCV*, 2023.

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. In *NeurIPS*, 2022.

Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., and Kiela, D. Flava: A foundational language and vision alignment model. In *CVPR*, 2022.

Tanwisuth, K., Zhang, S., Zheng, H., He, P., and Zhou, M. Pouf: Prompt-oriented unsupervised fine-tuning for large pre-trained models. In *ICML*, 2023.

Tu, C.-H., Mai, Z., and Chao, W.-L. Visual query tuning: Towards effective usage of intermediate representations for parameter and memory efficient transfer learning. In *CVPR*, 2023.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *JMLR*, 2008.

Wang, F., Li, M., Lin, X., Lv, H., Schwing, A., and Ji, H. Learning to decompose visual features with latent textual prompts. In *ICLR*, 2022.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019.

Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., and Cao, Y. Simvlm: Simple visual language model pretraining with weak supervision. In *ICLR*, 2021.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Wu, C., Wang, T., Ge, Y., Lu, Z., Zhou, R., Shan, Y., and Luo, P. π-tuning: Transferring multimodal foundation models with optimal multi-task interpolation. In *ICML*, 2023.

Wu, H.-H. and Wu, S. Various proofs of the cauchy-schwarz inequality. *Octogon mathematical magazine*, 2009.

Xu, J., De Mello, S., Liu, S., Byeon, W., Breuel, T., Kautz, J., and Wang, X. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 2022.

Xue, H., Huang, Y., Liu, B., Peng, H., Fu, J., Li, H., and Luo, J. Probing inter-modality: Visual parsing with self-attention for vision-and-language pre-training. In *NeurIPS*, 2021.

Yang, L., Wang, Y., Li, X., Wang, X., and Yang, J. Fine-grained visual prompting. In *NeurIPS*, 2024.

Yao, H., Zhang, R., and Xu, C. Visual-language prompt tuning with knowledge-guided context optimization. In *CVPR*, 2023.

Yao, Y., Zhang, A., Zhang, Z., Liu, Z., Chua, T.-S., and Sun, M. Cpt: Colorful prompt tuning for pre-trained vision-language models. *AI Open*, 5, 2024.

Yuan, L., Chen, D., Chen, Y.-L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

Zang, Y., Li, W., Zhou, K., Huang, C., and Loy, C. C. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022.

Zhang, C., Van Durme, B., Li, Z., and Stengel-Eskin, E. Visual commonsense in pretrained unimodal and multimodal models. In *NAACL*, 2022.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE PAMI*, 2017.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Conditional prompt learning for vision-language models. In *CVPR*, 2022a.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models. *IJCV*, 2022b.

Zhu, B., Niu, Y., Han, Y., Wu, Y., and Zhang, H. Prompt-aligned gradient for prompt tuning. In *ICCV*, 2023.

A. Proof of Theorem 1

This section outlines the proof of Theorem 4.1.

Definition A.1. (Linear independence (Boyd & Vandenberghe, 2018)). A collection of vectors $\{\boldsymbol{a}_1, \cdots, \boldsymbol{a}_k\}$ (with $k \geq 1$) is called *linearly independent* if it is not linearly dependent, which means that $\beta_1 \boldsymbol{a}_1 + \cdots + \beta_k \boldsymbol{a}_k = \mathbf{0} \Leftrightarrow \beta_1 = \cdots = \beta_k = 0$.

Theorem A.2. *(Cauchy–Schwarz inequality (Wu & Wu, 2009)). Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be arbitrary vectors in an inner product space over the scalar field $\mathbb{R}$. Then $|\langle \boldsymbol{u}, \boldsymbol{v} \rangle| = \|\boldsymbol{u}\|\|\boldsymbol{v}\| \Leftrightarrow \exists \lambda \in \mathbb{R} : \boldsymbol{u} = \lambda \cdot \boldsymbol{v}$.*

Let $f : \mathbb{R}^n \to \mathbb{R}^d$ and $g : \mathbb{R}^m \to \mathbb{R}^d$ denote two functions. For vectors $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{y} \in \mathbb{R}^m$ with the *same semantic context*, $\cos(f(\boldsymbol{x}), g(\boldsymbol{y}))$ is supposed to be 1.

Assumption A.3. Let f be linear functional and satisfies the condition

$$\boldsymbol{x} \neq \mathbf{0} \Rightarrow f(\boldsymbol{x}) \neq \mathbf{0}. \tag{10}$$

Assumption A.4. Let $\boldsymbol{x}$ can be decomposed into two vectors as follows

$$\boldsymbol{x} = \boldsymbol{x}_1 + \boldsymbol{x}_2, \tag{11}$$

where

$$\cos(f(\boldsymbol{x}_1), g(\boldsymbol{y})) = 1, \tag{12}$$
$$\cos(f(\boldsymbol{x}_2), g(\boldsymbol{y})) < 1. \tag{13}$$

Now we begin the proof of Theorem 4.1.

Proof. (Contradiction) Suppose

$$\cos(f(\boldsymbol{x}_1 + \boldsymbol{x}_2), g(\boldsymbol{y})) = 1, \tag{14}$$

when $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ satisfy *linear independence* (See Definition A.1).

Theorem A.2 shows that Eq. (14) implies $\exists \lambda_1 (\lambda_1 \neq 0)$ such that

$$f(\boldsymbol{x}_1 + \boldsymbol{x}_2) = \lambda_1 g(\boldsymbol{y}), \tag{15}$$

and Eq. (12) implies $\exists \lambda_2 (\lambda_2 \neq 0)$ such that

$$f(\boldsymbol{x}_1) = \lambda_2 g(\boldsymbol{y}). \tag{16}$$

Combining Eq. (15) and Eq. (16), then

$$f(\boldsymbol{x}_1 + \boldsymbol{x}_2) = \lambda f(\boldsymbol{x}_1), \tag{17}$$

where $\lambda := \frac{\lambda_1}{\lambda_2}$. According to f is linear functional, we have

$$f(\boldsymbol{x}_1) + f(\boldsymbol{x}_2) = \lambda f(\boldsymbol{x}_1), \tag{18}$$

which can also be expressed as

$$(1 - \lambda) f(\boldsymbol{x}_1) + f(\boldsymbol{x}_2) = \mathbf{0}. \tag{19}$$

Then as f is linear functional, we have

$$f((1 - \lambda)\boldsymbol{x}_1 + \boldsymbol{x}_2) = \mathbf{0}. \tag{20}$$

Note that the contraposition of the statement in Eq. (10) is

$$f(\boldsymbol{x}) = \mathbf{0} \Rightarrow \boldsymbol{x} = \mathbf{0}. \tag{21}$$

Combining Eq. (20) and Eq. (21) we have,

$$(1 - \lambda)\boldsymbol{x}_1 + \boldsymbol{x}_2 = \mathbf{0}, \tag{22}$$

where we have $\beta_1 = 1 - \lambda$ and $\beta_2 = 1$. This violates the condition that $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ satisfy linear independence. This concludes that the assumption Eq. (14) is *false*.

Table 5. Comparison of WCA with existing approaches on cross-dataset evaluation with fine-tuned methods using 16-shot training data per category (CoOp, CoCoOp, MaPLe) and test-time prompting (TPT, DiffTPT) methods requiring extra parameter turning. We report the top-1 classification accuracy (%) on each dataset.

Method	Tuned?	Source	Target				Average
		ImageNet	DTD	Pets	Food101	Flowers102	
CoOp (Zhou et al., 2022b)	✓	**71.51**	41.92	89.00	85.30	68.71	71.29
CoCoOp (Zhou et al., 2022a)	✓	71.02	45.73	88.71	86.06	71.88	72.68
MaPLe (Khattak et al., 2023)	✓	70.72	46.49	90.49	86.20	72.23	73.23
TPT (Shu et al., 2022)	✓	69.70	46.23	86.49	86.93	69.31	71.73
DiffTPT (Feng et al., 2023)	✓	70.30	47.00	88.22	87.23	70.1	72.57
CLIP-E (Radford et al., 2021)	✗	68.37	45.27	89.1	88.83	71.48	72.61
CuPL (Pratt et al., 2023)	✗	69.91	50.53	91.14	88.98	73.39	74.79
WCA (Ours)	✗	71.08	**54.02**	**91.96**	**89.98**	**73.66**	**76.14**

As the range of cos is $[-1, 1]$, therefore we have

$$\cos(f(\boldsymbol{x}_1 + \boldsymbol{x}_2), g(\boldsymbol{y})) < 1, \tag{23}$$

then according to Eq. (11) we have

$$\cos(f(\boldsymbol{x}), g(\boldsymbol{y})) < 1, \tag{24}$$

when $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is *linear independent*. □

B. Further Experiments

B.1. Cross-dataset Generalization Results

Table 5 demonstrates the performance of various state-of-the-art fine-tuned methods utilizing 16-shot training data per category and test-time prompting (TPT) methods which require at least one test sample. Notably in terms of in-distribution generalization ability (ImageNet), ours stands out by achieving competitive results without the need for any data-driven parameter tuning. A crucial highlight for the performance in ImageNet is that our method attains comparable results, being just marginally less accurate than CoOp, while we significantly surpass other methods. However, the distinction lies in our approach's superior generalization ability, which is notably stronger than that of other methods. This observation signifies that while our method does not necessitate model fine-tuning, it assures the preservation of the foundation model's generalization capacity. In essence, our method achieves remarkable performance without compromising the model's inherent ability to generalize to new, unseen data within the image distribution.

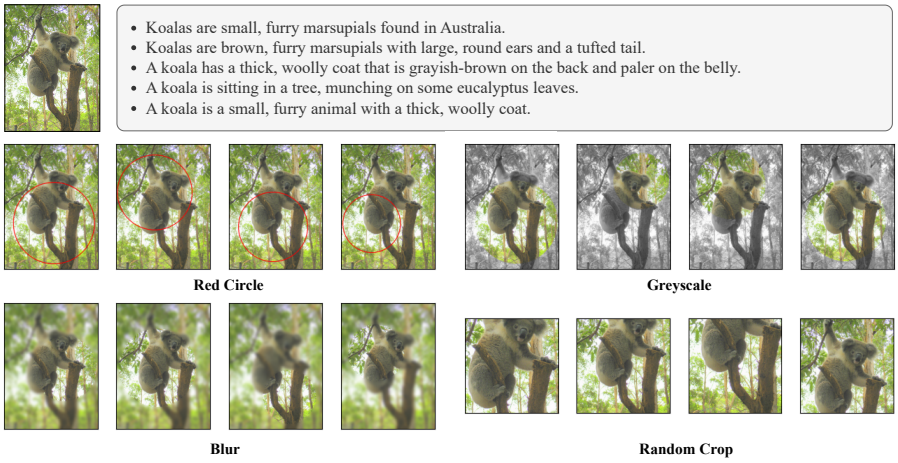
B.2. Experiment with Various Visual Prompting Methods

The Table 6 is presented to evaluate the top-1 accuracy of image classification in ImageNet under different prompting methods. CLIP serves as the baseline accuracy of 64.37% without any visual prompting methods. The subsequent columns represent different visual prompting techniques: "Red Circle", "Blur", "Greyscale", and "Random Crop", which results in a varying impact on the model's performance. Notably, "Red Circle" and "Blur" show a slight decrease in accuracy (61.83% and 61.85%, respectively) compared to the baseline. "Greyscale" results in a minor drop in accuracy (62.36%), indicating that color information might be somewhat relevant to the model's performance. Our method, "Random Crop" increases the accuracy to 66.84%, surpassing the baseline and other methods. This indicates that this particular prompting method might be introducing some beneficial variance or focusing the model's attention on more relevant features of the images.

B.3. Experiment with Various Aggregating Methods

The Table 7 illustrates the zero-shot visual classification performance using various aggregation techniques. It is evident from the table that both Max and Mean aggregation methods are not only less effective compared to our method but also fall behind the baseline in some cases.

Table 6. Top-1 accuracy (%) of various visual prompting methods in ImageNet. The figure on the left demonstrates the examples of each visual prompting method.



Method	Accuracy
Baseline (CLIP)	64.37
Red Circle	61.83 ± 0.06
Blur	61.85 ± 0.07
Greyscale	62.36 ± 0.07
Random Crop (Ours)	**66.84** ± 0.07

Table 7. Accuracy (%) of various aggregating methods on ImageNet with CLIP ViT-B/32, ViT-B/16 and ViT-L/14, where the baseline is selected from the top-performing method from Table 1.

Backbone	Baseline	Aggregation Method		
		Max	Mean	WCA (Ours)
ViT-B/32	64.37	57.53	65.51	**66.84**
ViT-B/16	69.61	59.94	69.72	**71.08**
ViT-L/14	76.62	67.99	76.34	**77.32**

B.4. Experiment with Residual Network Backbone

We further conduct the experiments with the Residual Network (ResNet) backbone in ImageNet as shown in Table 9. We can see our method consistently outperforms other baselines.

B.5. Exploring Different VLM Architectures

This section is dedicated to an ablation study involving various VLMs characterized by distinct model architectures and pre-training datasets. Specifically, we examine models such as ALIGN (Jia et al., 2021), AltCLIP (Chen et al., 2022), and GroupVit (Xu et al., 2022). The outcomes of this exploration are detailed in Table 10. A critical observation from this study is that our method consistently outperforms others even under different VLMs. This consistency underscores the adaptability and effectiveness of our methodology when applied to a diverse range of VLM architectures.

B.6. Incorporating the Entire Image Features.

We have explored the possibility of incorporating the entire image into our method. We devised a new scoring function for the form: $\lambda \cdot \text{Sim}(x, h(y)) + (1 - \lambda) \cdot \text{Sim}(p(x), h(y))$, where x represents the entire image, $p(x)$ denotes the patch images, $h(y)$ signifies the LLM-generated text descriptions and λ serves as a hyperparameter controlling the balance between scores related to the entire image, and patches. However, after thorough experimentation as shown in the table below, we found that the performance improvement achieved by this modification was marginal (0.06% when $\lambda = 0.1$). Moreover, integrating the entire image into the methodology introduced an additional parameter (λ) and complexities without significant enhancement in performance.

B.7. Experiment with Google PaLM

Despite observing a performance improvement using GPT-3 (Brown et al., 2020), we further explored the capabilities of other large-scale language model experiments, such as Google PaLM API (Anil et al., 2023). We fetched the descriptions for

Table 8. Experiment results of various β values. α is set to 0.5. This experiment is evaluated in ImageNet for CLIP ViT-B/32.

$\alpha = 0.5$	β				
	0.6	0.7	0.8	0.9	1
Top-1 Accuracy (%)	61.79	63.2	64.45	66.84	66.06

Table 9. Zero-shot visual classification top-1 accuracy (%) of various ResNet backbone CLIP. σ represents the standard deviation. Δ stands for the performance gain achieved by our method over the best baseline, which is denoted as underlined. Each backbone's top score is shown in **bold**.

Backbone	CLIP	CLIP-E	CLIP-D	Waffle	CuPL	WCA (Ours)	σ	Δ
ResNet-50	58.19	59.70	59.50	60.40	61.32	**62.87**	0.05	+1.55
ResNet-101	61.22	62.33	61.92	62.97	64.09	**64.98**	0.04	+0.89
ResNet-50x4	65.52	66.54	66.09	67.11	68.01	**68.37**	0.04	+0.36

ImageNet classes with the same prompt used in CuPL (Pratt et al., 2023). The outcome shows that the PaLM-based CLIP model achieved a 62.16% accuracy, marginally exceeding the original CLIP's performance by 0.05%. By incorporating description weighting as Eq. 8, we managed to improve the performance to 64.03%. This improvement highlights the effectiveness of the weighted approach regarding the situation for some reasons LLMs may provide less useful descriptions.

C. Further Analysis and Discussion

C.1. Visualisation of Prompt Image Embedding

WCA is defined as in Eq. (9), and here we look at another view of our method:

$$
\begin{aligned}
s_{\text{WCA}}(\boldsymbol{x}, \boldsymbol{y}|f, g) &= \sum_{i=1}^{N}\sum_{j=1}^{M} w_i v_j \frac{f(\boldsymbol{x}_i)^{\mathrm{T}} g(\boldsymbol{y}_j)}{\|f(\boldsymbol{x}_i)\|\|g(\boldsymbol{y}_j)\|} \\
&= \left(\sum_{i=1}^{N} w_i \frac{f(\boldsymbol{x}_i)}{\|f(\boldsymbol{x}_i)\|}\right)^{\mathrm{T}} \left(\sum_{j=1}^{M} v_j \frac{g(\boldsymbol{y}_j)}{\|g(\boldsymbol{y}_j)\|}\right) \\
&= \underbrace{\left(\sum_{i=1}^{N} p_i f(\boldsymbol{x}_i)\right)^{\mathrm{T}}}_{\text{Augmented Image Embedding}} \underbrace{\left(\sum_{j=1}^{M} q_j g(\boldsymbol{y}_j)\right)}_{\text{Augmented Text Embedding}} = \boldsymbol{k}^{\mathrm{T}}\boldsymbol{t}
\end{aligned}
$$

where $p_i := \frac{w_i}{\|f(\boldsymbol{x}_i)\|}, q_j := \frac{v_j}{\|g(\boldsymbol{y}_j)\|} \in \mathbb{R}$, and $\boldsymbol{k} := \sum_{i=1}^{N} p_i f(\boldsymbol{x}_i), \boldsymbol{t} := \sum_{j=1}^{M} q_j g(\boldsymbol{y}_j) \in \mathbb{R}^d$. Based on the derivation, our visual-text alignment score is equivalent to the inner product of the augmented visual embedding $\boldsymbol{k}$ and text embedding $\boldsymbol{t}$. The augmented embeddings are computed as the weighted sum of the embeddings of the image patches/descriptions. $\boldsymbol{k}$ and $\boldsymbol{t}$ can be pre-computed to be stored for quick access in later use.

We employ t-SNE (Van der Maaten & Hinton, 2008) to compare the image embedding of CLIP and our method, using a dataset comprising 500 samples from 10 selected ImageNet classes. As illustrated in Figure 8, our method demonstrates more distinct class boundaries, particularly for classes 3 (dowitcher) and 7 (tusker), and a clear separation between classes 1 (indri) and 10 (three-toed sloth). Focusing on classes 1 and 10, which are visually similar as both species climb trees, we observe in Figure 9 that CLIP tends to misclassify an indri as a sloth due to its emphasis on general semantic information. Our method, however, utilizes localized visual prompting to effectively discount tree-related features, enhancing accuracy. Additionally, we explore an outlier purple point near class 7, shown in Figure 10, which, despite being an anomaly, aligns closely with the Tusker group images.

C.2. Complexity Analysis.

The computational analysis in Table 13 offers insights into the efficiency of our method. While CLIP stands out for its computational efficiency, it falls short in performance compared to other methods. In contrast, CoOp with its high

Table 10. Top-1 accuracy (%) for zero-shot visual classification conducted on various VLMs in ImageNet. Each column represents a different prompting method, e.g., CLIP refers to "A photo of {label}". σ represents the standard deviation. Δ stands for the performance gain achieved by our method over the best baseline, which is denoted as underlined. Each VLM's top score is shown in **bold**.

VLM	CLIP	CLIP-E	CLIP-D	Waffle	CuPL	WCA (Ours)	σ	Δ
ALIGN (Jia et al., 2021)	65.24	65.79	65.08	65.22	66.24	**66.77**	0.09	+0.53
AltCLIP (Chen et al., 2022)	73.79	74.86	74.48	74.29	75.74	**76.20**	0.04	+0.46
GroupViT (Xu et al., 2022)	37.11	42.72	40.10	42.42	44.53	**45.27**	0.05	+0.74

Table 11. Impact of varying λ on accuracy (%). The table shows the accuracy values for different λ settings, indicating how the accuracy decreases as λ increases from 0.0 to 1.0.

λ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy	66.84	66.90	66.88	66.77	66.65	66.48	66.28	65.99	65.65	65.28	64.83

computational cost only achieves similar performance levels to WCA. Our method, WCA, strikes a commendable balance between computational load and performance efficiency. It surpasses CLIP in terms of performance while maintaining a significantly lower computational complexity compared to CoOp. This analysis underscores WCA's effectiveness as a more optimized solution in the vision-language modeling space, offering a desirable compromise between computational demands and model performance.

This observation highlights a key strategy for optimizing the time complexity of the WCA method. The primary source of computational demand stems from the image-prompting process. However, this challenge can be effectively mitigated by pre-computing and storing the embeddings of image prompts on a hard drive. By implementing this approach, the computation cost of WCA can be brought in line with that of CLIP, effectively neutralizing the additional computational overhead associated with our method. This not only enhances the efficiency of WCA but also preserves its superior performance capabilities, making it a highly practical and competitive option in the realm of vision-language models.

D. Limitation

Our approach, while effective in certain scenarios, particularly in object recognition tasks such as identifying an image containing a dog, exhibits limited success in contexts requiring comprehensive image understanding. An example of this can be seen with the EuroSAT dataset (Helber et al., 2018), which is used for land use and land cover classification. This dataset demands a holistic grasp of the entire image, a requirement evident in Figure 13. The performance in this context challenges our initial assumption that an area of the image would align perfectly with its textual description as described in Theorem 4.1. It suggests that further refinements or a different approach might be necessary for tasks that require a deeper, more holistic understanding of images, as opposed to those that focus on identifying individual objects.

Figure 13. Overview of the EuroSAT dataset.

We have also identified another limitation, particularly when dealing with images containing multiple objects of varying sizes. For example, if the task is to identify a small cat in an image crowded with larger dogs, the patch weights might inadvertently emphasize the dogs while downplaying the cat, potentially hindering performance. Additionally, the current approach for text weighting, which relies on cosine similarity to a base description, such as "a photo of a {category}", might not always be optimal, resulting in sub-optimal text weighting.

E. Example PaLM Generated Prompts

To generate these prompts, we utilized the PaLM model (Anil et al., 2023) with a specific configuration aimed at producing diverse and detailed responses. Here is a step-by-step breakdown of the process:

Table 12. Experiment results comparing the accuracy (%) of different methods using Google LLM PaLM. The table shows the accuracy percentages for CLIP, CuPL-PaLM, and WCA-PaLM.

Method	CLIP	CuPL-PaLM	WCA-PaLM
Accuracy (%)	62.05	62.16	**64.03**

Table 13. Computation analysis. This table outlines CLIP, 16-shot CoOp, and our method, which demonstrates that our method is very efficient compared to the prompting turning method, and achieves comparable results. Even though zero-shot CLIP is the most efficient but underperforms other methods.

Method	#Params	Accuracy (%)	Time (hh:mm:ss)
CLIP-D	0	63.01	00:00:35
CoOp	2,048	66.85	11:34:05
WCA (Ours)	0	66.84	00:06:42

1. **Template Selection:** We started by selecting appropriate templates for generating the prompts. These templates were based on the structure provided by (Pratt et al., 2023), ensuring consistency and clarity in the generated prompts.
2. **Prompt Generation:** For each category, such as the "Koala" category, we fed the selected templates into the model. The prompt question, in this case, was "Describe what a(n) {koala} looks like." The model then generated multiple responses based on this prompt.
3. **Response Collection:** The model was set to generate around eight responses for each prompt. These responses were collected and reviewed to ensure they met the criteria of being descriptive and relevant.
4. **Post-processing:** After generating the responses, minor post-processing was performed to format the output for clarity and presentation. This included organizing the responses into a list format and ensuring they adhered to the prompt's context.

Below is an example of the generated descriptions for each prompt:

```
Prompt:  Describe what a(n) koala looks like.
Responses:

1. Koalas are small, furry marsupials found in Australia.
2. A koala is a small, tree-dwelling marsupial found in Australia.
3. A koala looks like a small, stocky bear with a large head and a long tail.
4. Koalas are small, furry marsupials that are found in Australia.
5. A koala is a small, furry animal that lives in Australia.
6. A koala is a small, furry animal with a large head and a long tail.
7. A koala is a small, furry animal with a thick, woolly coat.
8. Koalas are small, furry marsupials found in Australia.
```

F. Examples of Decisions and Justifications

As Menon & Vondrick (2022) discussed, LLM-based CLIP not only provides better performance but also explains the model. Therefore, we randomly selected 10 examples in ImageNet, where we made correct predictions but not for our baseline CLIP-D as shown in Figure 12 and 14. This illustrates the effectiveness of our method. For instance, the top row in 12 describes a photo of a gas mask, where CLIP-D incorrectly predicts that as a vacuum cleaner, CLIP-D witnesses a high score for its description "wheels or casters for easy movement.", while our model correctly predicts it as gas mask since our model shows the high scores for the descriptions of gas mask, such as "a filter attached to the mask". This means our method can recognize this feature inside this photo.

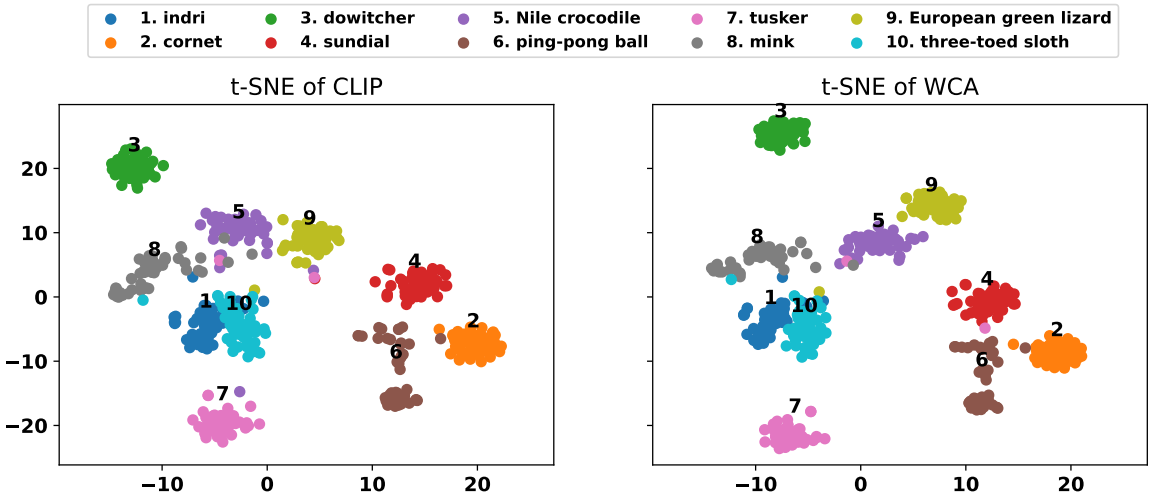


Figure 8. Visualization of t-SNE plots comparing CLIP and WCA image embeddings of 10 classes in ImageNet. Each plot represents 50 samples per class, showing the spatial distribution of the embeddings. The classes include indri, cornet, dowitcher, sundial, Nile crocodile, ping-pong ball, tusker, mink, European green lizard, and three-toed sloth. The left plot illustrates the t-SNE of CLIP embeddings, while the right plot shows the t-SNE of WCA embeddings, highlighting the differences in how each model represents the image data.

Figure 9. Example images of two animal species from ImageNet: (a) indri, shown in two images capturing its arboreal lifestyle, typically found in the forest canopies of Madagascar; (b) three-toed sloth, depicted in its natural habitat, displaying its characteristic slow movement and hanging posture in the trees of Central and South American rainforests. These images illustrate the visual diversity and distinct ecological niches of each species.

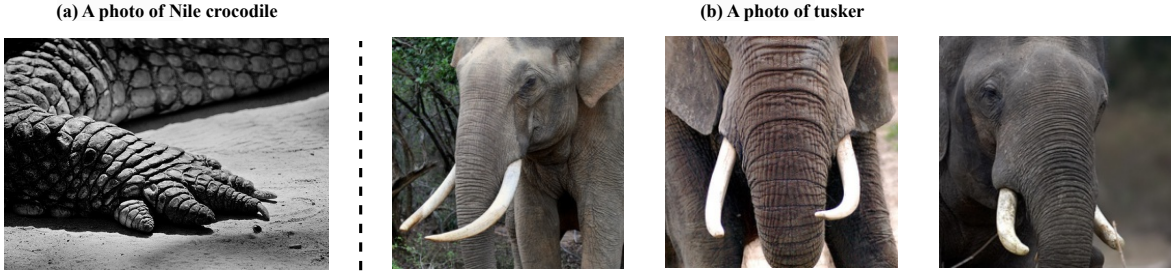


Figure 10. Example images of two animal species from ImageNet: (a) Nile crocodile, depicted with a focus on its textured scales and powerful limbs, highlighting its adaptations for an aquatic lifestyle; (b) tusker, shown in three images featuring close-ups of its large tusks and distinctive features, emphasizing its status as a majestic and significant member of the elephant family. These images illustrate the visual details and distinguishing characteristics of each species.

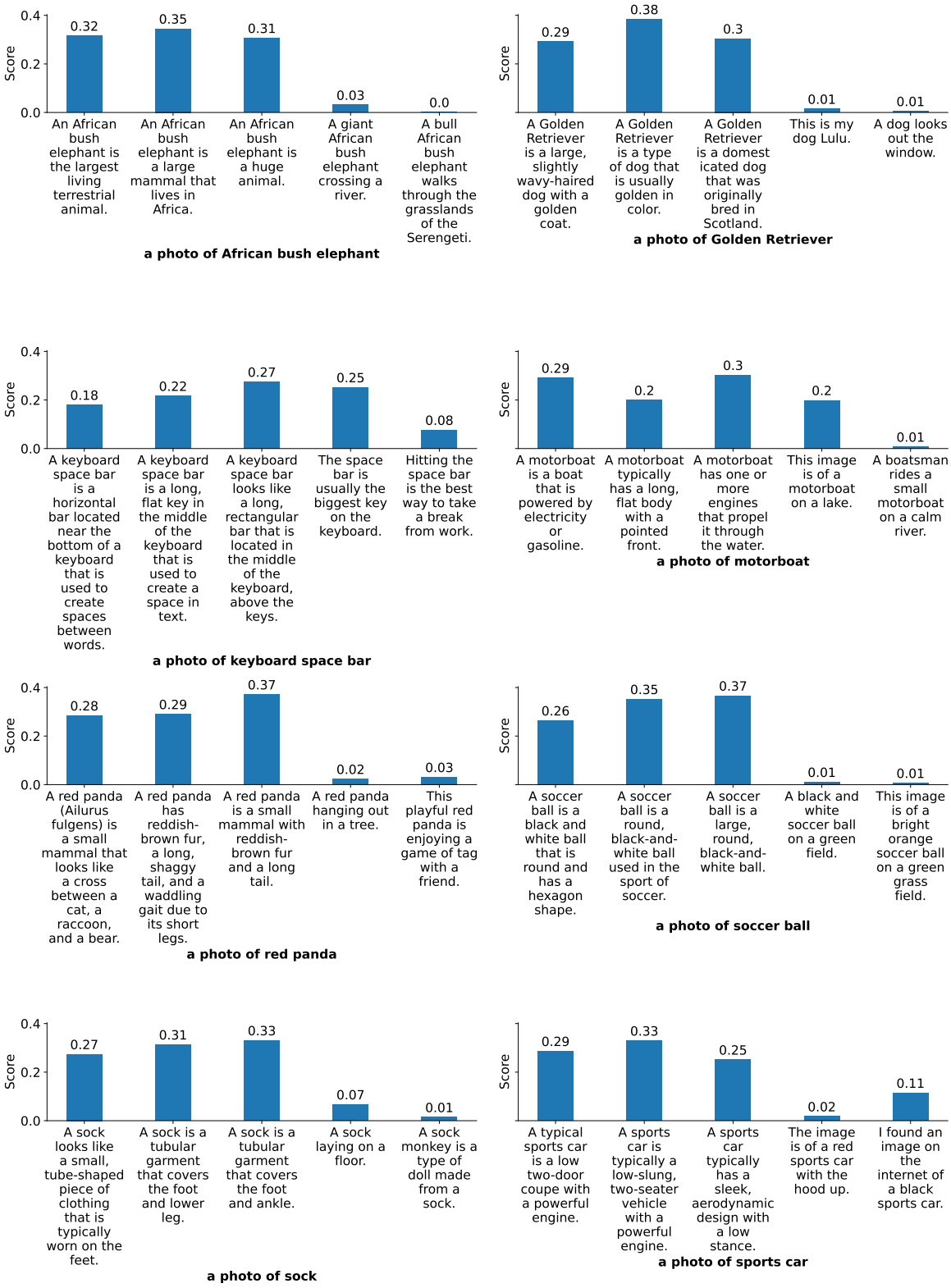


Figure 11. Feature plots comparing the textual alignment scores for different categories: African bush elephant, Golden Retriever, keyboard space bar, motorboat, red panda, soccer ball, sock, and sports car. Each plot shows the alignment scores for five different textual descriptions, highlighting the variation in performance across different categories.

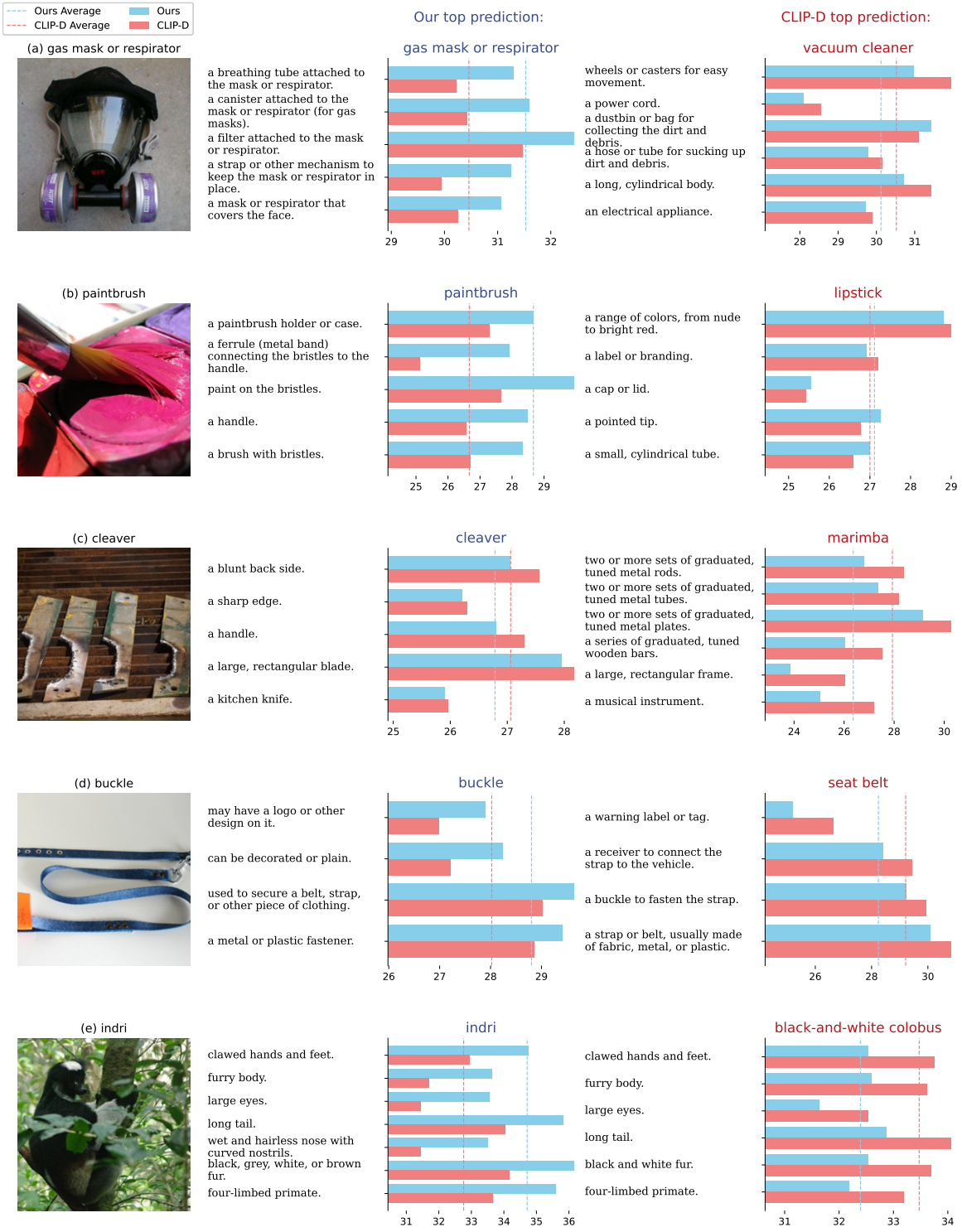


Figure 12. An illustration of the prediction and explanation of comparison between CLIP-D and our methods. The value in the plot represents the similarity score (higher denotes a high similarity). We can use the LLM-generated descriptions to explain the decisions by the model. For example, the top row means our method predicts it correctly as a gas mask as our method can recognize a filter in the image, while CLIP-D recognizes it as a vacuum cleaner.

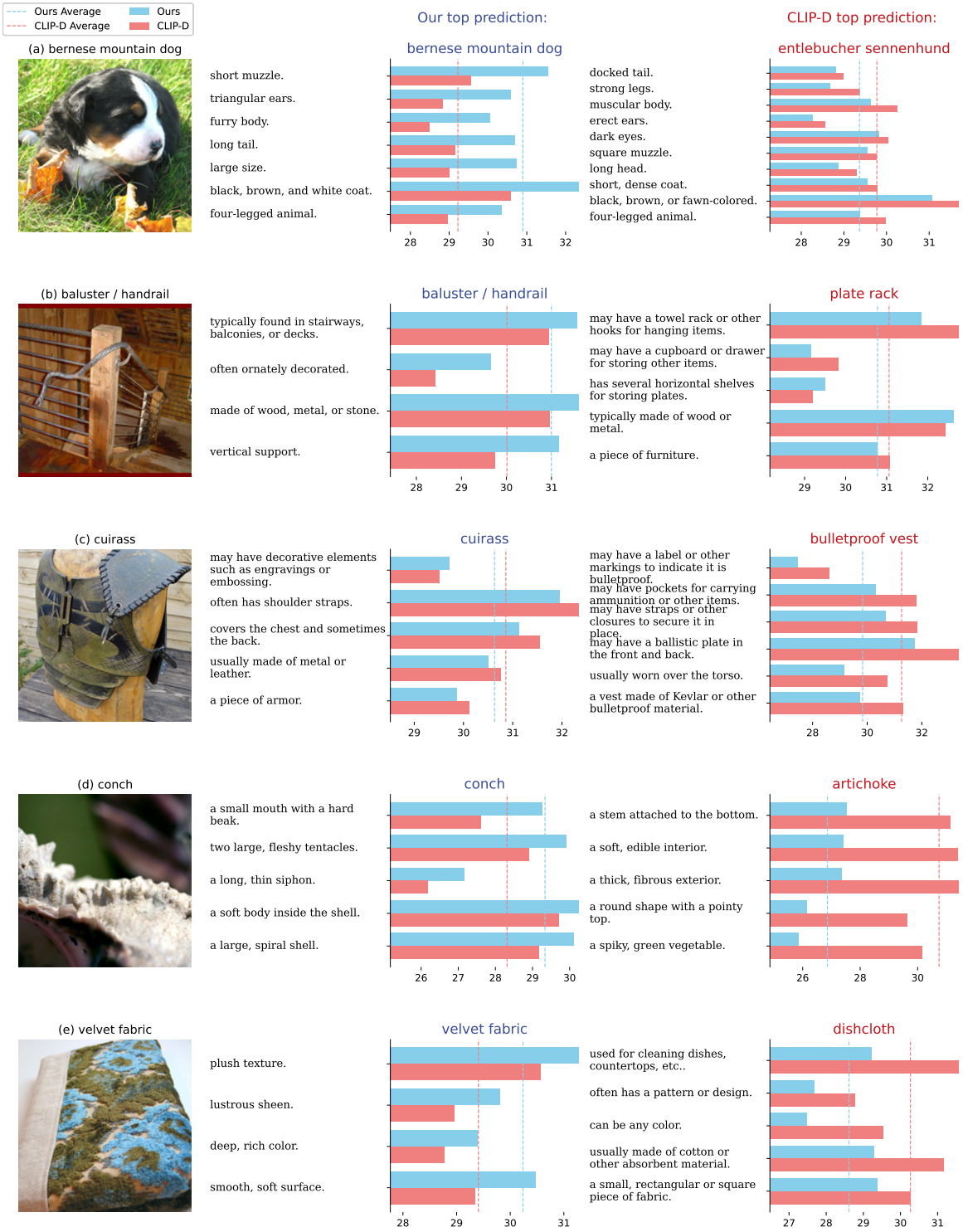


Figure 14. This figure is set in the same context as Figure 12 but with different images.