Improved Zero-Shot Classification by Adapting VLMs with Text Descriptions

Oindrila Saha Grant Van Horn Subhransu Maji
University of Massachusetts, Amherst
{osaha, gvanhorn, smaji}@umass.edu

Abstract

The zero-shot performance of existing vision-language models (VLMs) such as CLIP [29] is limited by the availability of large-scale, aligned image and text datasets in specific domains. In this work, we leverage two complementary sources of information—descriptions of categories generated by large language models (LLMs) and abundant, fine-grained image classification datasets—to improve the zero-shot classification performance of VLMs across fine-grained domains. On the technical side, we develop methods to train VLMs with this "bag-level" image-text supervision. We find that simply using these attributes at test-time does not improve performance, but our training strategy, for example, on the iNaturalist [41] dataset, leads to an average improvement of 4-5% in zero-shot classification accuracy for novel categories of birds [42] and flowers [23]. Similar improvements are observed in domains where a subset of the categories was used to fine-tune the model. By prompting LLMs in various ways, we generate descriptions that capture visual appearance, habitat, and geographic regions and pair them with existing attributes such as the taxonomic structure of the categories. We systematically evaluate their ability to improve zero-shot categorization in natural domains. Our findings suggest that geographic priors can be just as effective and are complementary to visual appearance. Our method also outperforms prior work on prompt-based tuning of VLMs. We release the benchmark, consisting of 14 datasets at https://github.com/cvl-umass/AdaptCLIPZS, which will contribute to future research in zero-shot recognition.

1. Introduction

Recent improvements in zero-shot classification have been due, in part, to success in training VLMs at scale. Models such as CLIP [29], ALIGN [11], and BLIP [16] use massive datasets of image and text pairs to learn a common embedding between visual and natural language domains. However, we find existing VLMs show poor performance in encoding visual attributes in fine-grained domains, beyond simply recognizing the name of the category. For example, we observe concatenating visual attribute descriptions to the species name for the bird species in the CUB [42] dataset improves the zero-shot classification from 50.5% to only 50.7%, while for Cars [14], the performance even drops slightly (see Tab. 1). Although the datasets on which VLMs are trained are extensive, they often lack the details that experts may require for fine-grained categorization. At the same time, collecting large-scale image-caption datasets in these domains requires significant effort, making training similar models challenging.

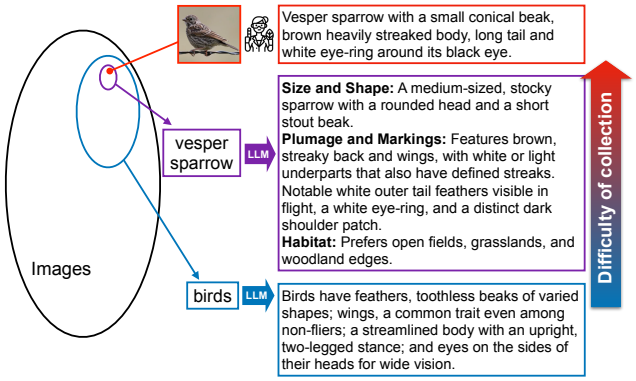


Figure 1. **Motivation.** Collecting image captions in fine-grained domains requires expertise (top row), but LLMs can generate structured (e.g., shape or appearance) and accurate descriptions of categories at both the coarse (e.g., birds) and fine-grained level (e.g., Vesper Sparrow). Rich descriptions of fine-grained categories can be paired with existing datasets, such as iNaturalist [41] and NABirds [39] to generate coarsely-aligned image-text datasets for fine-tuning VLMs. This improves their zero-shot performance on a range of benchmarks, generalizing to novel categories and tasks.

In this work, we leverage two complementary sources of information—large language models (LLMs) and abundant, fine-grained image classification datasets—to improve the zero-shot classification performance of VLMs across a variety of fine-grained domains. Concretely, we generate large datasets of images aligned with text by pairing images within a category with descriptions of that category generated by LLMs as seen in Fig. 1. We find this approach works well, as images within a fine-grained domain share many attributes, unlike in coarse categories with

larger intra-category variation. At the same time, we find that LLMs are capable of accurately describing appearance, habitat, and other properties for a wide range of categories, allowing us to systematically generate datasets in a scalable manner. In other words, fine-grained labels allow us to bridge the gap between image-level captions required for VLM training, and general information about visual categories contained in LLMs.

On the technical side, we develop methods to train VLMs with "bag-level" supervision. In our dataset a set of images are grouped with a set of descriptions and lack the image-text correspondences. Some of the descriptions may not apply to an image (e.g., the part may be occluded). However, we find that training by stochastically pairing the images and text within a category, followed by a category-level contrastive loss similar to CLIP objective offers robust improvements in performance. Adapting semi-supervised learning approaches such as FixMatch [35] or Knowledge distillation [3] results in minor improvements (§ 5.7). A detailed investigation of the image-text association within a category suggests the model is able to correctly associate the visual attributes with the corresponding text even they are paired randomly during training (Fig. 2).

We systematically evaluate the effectiveness of our method by assessing the zero-shot classification performance on novel classes. We find that simply using these attributes of novel classes generated by LLMs does not improve performance when using CLIP (Tab. 1). However, our training strategy leads to an average improvement of 4-5% in accuracy across 12 datasets, and outperforms baselines (Tab. 2). For natural domains (e.g., iNaturalist and NABirds), we prompt LLMs in various ways to generate descriptions that capture visual appearance, habitat, and geographic regions, and pair them with existing attributes within the dataset, such as taxonomic structure. Our results indicate that geographic priors are equally effective and complementary to visual appearance cues (see Tab. 3). Training on iNaturalist without any bird classes improves the performance of CLIP on CUB by more than 3%, and we observe similar improvements when evaluating across other domains (see Tab. 4). Improvements are consistent across text generated by different LLMs, as well as by humans (see Tab. 5). Our model also results in relative error reduction of 4.1% over CLIP on the challenging NeWT dataset [40].

2. Related Work

Zero-shot Image Classification using VLMs. Vision Language models (VLMs) [6, 11, 29, 33, 45] learn to associate images with their corresponding text captions. Learning a shared embedding makes them perform exceptionally well in zero-shot classification tasks when paired with an appropriate text such as the class name during test time. FLAVA [33] learns using paired as well as unpaired images and texts using different losses for multimodal and unimodal understanding. ALIGN [11] uses a large number of noisy image-text data by obtaining alt-texts for images and trains using a contrastive loss. CLIP [29] is trained on a smaller and cleaner dataset of image-text pairs using a similar objective function. It employs a vision model and a language model to learn joint embeddings of images and text. While training, it maximizes the similarity between related image-text pairs and minimizes similarity between unrelated pairs. At test time the similarity over captions such "a photo of a `[class name]`" over all classes in the domain for each image is found. The image is classified to the class with the caption with highest probability. The original paper shows that manual prompt tuning can boost zero-shot classification accuracy.

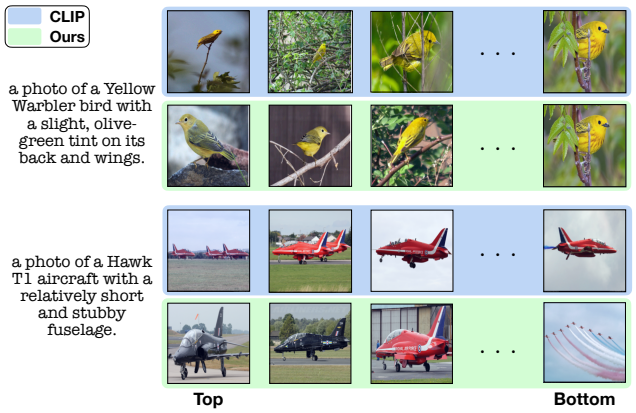


Figure 2. **Visualizing image-text similarity.** All images within a category are sorted in order of similarity to a given text predicted by CLIP and our fine-tuned CLIP^{FT}+A. For example, our method identifies birds which show olive-green tint on their back as the top images, whereas CLIP selects birds with visibly brown upperparts or occluded back. The image with lowest similarity which has the occluded back remains the same for both models, showing our model does not learn incorrect attribute associations even though we stochastically pair every attribute with every image during training. On the aircraft example our model predicts higher similarity to images with prominently visible fuselage. CLIP identifies the least similar image as one in which fuselage is visible, but ours chooses one where aircrafts are too far to make out the shape of fuselage.

Generating Better Prompts. Prior work on prompt tuning [7, 12, 13, 32, 51] has focused on improving the text descriptions of classes. For example, CoOp [50] appends learnable context vectors to the class name texts to improve classification. CoCoOp [49] and related methods have also explored prompting the vision encoder simultaneously. While prompt tuning has proven useful for adapting models to a set of "base categories", its performance on novel categories still falls short of the CLIP baseline.

Another line of research aims at querying LLMs to generate prompts or attributes of categories. CHiLS [24] refines classes based on GPT descriptions (e.g., taxonomic structure) and maps the image to one of the subcategories to improve classification. We also explore the ability to

learn taxonomy-based attributes in our work. Menon *et al*. [20] and CuPL [28] append class-specific attributes obtained from GPT [2] to simple prompts, e.g. "a photo of a `[class]`" to improve zero-shot performance at test-time, similar to our approach. However, we find that CLIP struggles to recognize nuanced attributes in fine-grained domains where we observe little to no improvement in classification performance, motivating the need for fine-tuning VLMs.

Fine-tuning VLMs. Most work on fine-tuning VLMs has focused on parameter efficient updates using lightweight adapters [8, 19, 25, 27, 46, 47] for improving few-shot classification. For example, CLIP adapter [8] trains a few learnable layers on top of the encoders, while Maniparambil *et al*. [19] query LLMs for class-wise descriptions for tuning an external adapter network. Their work improves over CoCoOp and CLIP adapter for unseen classes, however, as before, most approaches show no improvement over CLIP, especially on novel classes in fine-grained datasets.

Another line of research [9, 34, 37, 43, 48] involves fine-tuning CLIP for robustness to domain shifts. For instance, WiSE-FT [43] utilizes weight-space ensembling to improve performance on a sketch version of ImageNet. Similarly, LaFTer [21] employs fine-tuning both the image and text encoders using unpaired images and texts obtained by querying LLMs. These techniques focus on adapting to a target distribution, such as a specific set of test images or classes, rather than on generalizing to novel classes.

Two works similar to ours, GIST [15] and I2MVFormer [22], also utilize GPT to generate category-specific texts for fine-grained domains. GIST pairs each image with the n most similar texts within the category based on CLIP similarity and consolidates them into a caption. In contrast, our approach stochastically pairs images with text, a strategy we found to be more robust. Notably, our experiments showed that biasing sampling towards similar image and text pairs led to inferior results (see § 5.7). I2MVFormer uses a LLM to generate class descriptions based on texts provided by annotators and Wikipedia documents. The key differences between their work and ours lie in the use of human effort in their training data generation and starting from scratch resulting in much lower performance compared to ours. Finally, in contrast to both these works, we explore training using visual, habitat, and location information, as well as training on larger datasets such as NABirds and iNaturalist.

Summary To the best of our knowledge, ours is the first method demonstrating that fine-tuning CLIP with class-specific descriptions obtained by querying LLMs improves the zero-shot performance in fine-grained domains. Our approach leverages LLMs to generate image-text data that are coarsely aligned, making it particularly effective for fine-grained categories. Moreover, unlike prior work [19, 20], our method queries LLMs along various dimensions such as visual, taxonomy, habitat and geographic priors, and systematically evaluates their effectiveness.

3. Method

Consider a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ consisting of images $x_i \in \mathcal{X}$ and labels $y_i \in \mathcal{Y}$. A VLM such as CLIP [29] consists of an image encoder Θ and a text encoder Φ such that $\Theta(x) \approx \Phi(y)$ for images x with label y. We want to improve the **zero-shot performance** of CLIP on novel categories in fine-grained domains by fine-tuning the image and the text encoders. We do so either by splitting a dataset $\mathcal{D}$ into $\mathbf{K_{train}}$ training and $\mathbf{K_{test}}$ testing classes, or train our model on large datasets such as iNaturalist and NABirds by excluding classes or domains overlapping with our test set. Our framework consists of: 1) generating textual descriptions given the class names by prompting LLMs in different ways (§ 3.1); 2) fine-tuning CLIP using these descriptions using our proposed approach (§ 3.2); and 3) evaluating the models on downstream tasks (§ 3.3). Fig. 3 provides an illustration of our method.

3.1. Dataset Generation

For each dataset we generate texts for every class which can be used to differentiate it from other classes in the domain using **visual attributes**. We query an LLM as:

> What characteristics can be used to differentiate `[class]` from other `[domain]` based on just a photo? Provide an exhaustive list of all attributes that can be used to identify the `[domain]` uniquely. Texts should be of the form "`[domain]` with `[characteristic]`".

Here `[class]` is the class name for the K classes in the dataset `[domain]`. Each domain is associated to different datasets, for example for the CUB200 dataset, `[domain]` is "bird" and for iNaturalist dataset it is "organism". Appending the domain `[domain]` is helpful because it provides context about the set of the other classes to distinguish from, and reduces confusion across similar class names in other domains. The LLM produces l_k descriptions for category k. We append "a photo of `[class]`" to the generated texts resulting in descriptions of the form " a photo of a `[class]` `[domain]` with `[characteristic]`" for each class (more details in the experiment section). This results in a set of descriptions $\mathbf{Y^k}$ for each category.

We also separately query about the **habitat and geographic location** of occurrence for the classes in CUB200, Flowers102, NABirds and iNaturalist datasets. For this purpose we use the prompt:

> Where can we find a `[class]`? Produce a list of habitat and geographic location information that can be used to identify the `[domain]`.

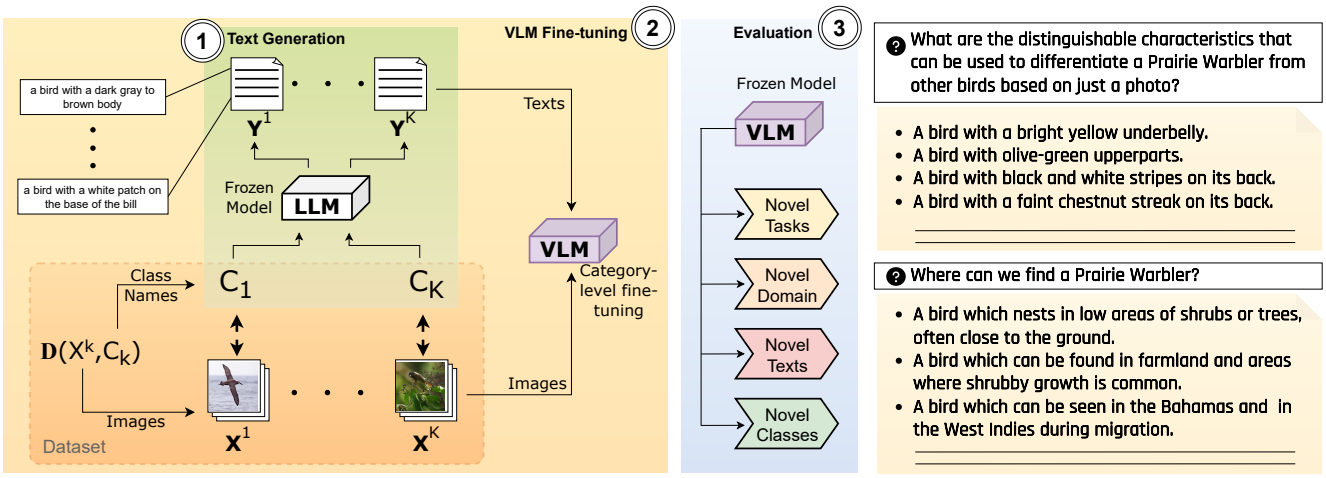


Figure 3. **Fine-tuning VLMs to improve zero-shot performance.** a) Our framework for ① generating fine-grained attributes per class using LLMs, ② category-level fine-tuning of VLMs and ③ evaluating on a series of challenging unseen scenarios. b) We show examples of texts produced in step ①.

We add the texts obtained to the category-level corpus $\mathbf{Y^k}$. The impact of these location-specific texts and the improvements offered are described in the results section. Examples of visual and habitat descriptions are in the Appendix. We determine the **correctness of the texts produced** for 4-6 classes of CUB, Aircraft and Cars by manual fact-checking texts with the help of online sources. We find that 96% (CUB), 90% (Aircraft) and 96% (Cars) are marked as correct by study participants (more details in Appendix). However, a challenge is that this manual vetting does not scale to large datasets, and we therefore rely on empirical results to support the utility of the generated text.

3.2. VLM Fine-Tuning

CLIP [29] is trained with image caption pairs. However, in our case we have a set of images $\mathbf{X^k}$ and a set of texts $\mathbf{Y^k}$ for classes $k \in \mathbf{K_{train}}$ in our training set. We address this by pairing every image with randomly sampled text from the corresponding category during training. However, we cannot directly use the batch-level cross-entropy loss used by CLIP which treats the paired text as positive and rest of the texts as negative. This is because the same batch can contain multiple pairs with images and texts belonging to the same category. Below we describe our modification of the objective function that addresses this.

In each iteration of training we sample a batch of size N consisting of $\{(x_i, y_i)\}_{i=1}^{N}$ pairs where both x_i and y_i is an image and text from the same class. Let the similarity score obtained using the forward pass of CLIP for image x_i and text y_j be S_{ij}. Let $\mathrm{c}(i)$ be the category of image-text pair (x_i, y_i). Let $G_i = \{j \mid \mathrm{c}(j) = \mathrm{c}(i)\}$ denote the indices of pairs that belong to the same class as pair (x_i, y_i). Then the loss function for images is:

$$\mathcal{L}_{image} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|G_i|}\sum_{j\in G_i}\log\frac{\exp(S_{i,j}/\tau)}{\sum_{r=1}^{N}\exp(S_{i,r}/\tau)} \quad (1)$$

and the corresponding one for texts is:

$$\mathcal{L}_{text} = -\frac{1}{N}\sum_{j=1}^{N}\frac{1}{|G_j|}\sum_{i\in G_j}\log\frac{\exp(S_{i,j}/\tau)}{\sum_{r=1}^{N}\exp(S_{r,j}/\tau)} \quad (2)$$

where τ is a learnable temperature parameter. The overall loss for fine-tuning is:

$$\mathcal{L}_{ft} = \mathcal{L}_{image} + \mathcal{L}_{text}$$

The objective aggregates the image text similarity across all image and text pairs from the same category within the batch. To avoid overfitting on small datasets we maintain momentum encoders whose weights (θ_{EMA}, ϕ_{EMA}) are updated with the exponential moving average (EMA) of the weights of the encoders (θ, ϕ) which is trained using the objective $\mathcal{L}_{ft}$:

$$\theta_{EMA} \leftarrow m\theta_{EMA} + (1-m)\theta_E$$
$$\phi_{EMA} \leftarrow m\phi_{EMA} + (1-m)\phi_E,$$

where m is a momentum parameter. All encoders are initialized using the pre-trained weights of CLIP.

3.3. Evaluation for Zero-shot Classification

To evaluate a model on unseen classes we similarly query the LLM as described in §3.1 to obtain texts $\mathbf{Y^k}$, for $k \in \mathbf{K_{test}}$. For any given image x we can find the similarity score using a VLM for every text y_m^k for $m \in \{1, \ldots, l_k\}$

and $k \in \mathbf{K_{test}}$. Denote the similarity between image x and text y_m^k as S_m^k. The predicted class is:

$$\underset{k}{\text{argmax}} \frac{1}{l_k} \sum_{m=1}^{l_k} \frac{\exp(S_m^k)}{\sum_{p \in \mathbf{K_{test}}} \sum_{q=1}^{l_p} \exp(S_q^p)} \tag{3}$$

The score represents average similarity between an image and the texts corresponding to each class. Our initial experiments suggested that simple averaging of probabilities is more robust than alternatives such as the geometric mean.

4. Experiments

In this section we present the experimental details of our approach. We outline the datasets we use, the particulars of implementation for each part of the method as well as the details of the baselines we compare our method to.

4.1. Datasets

We use a variety of fine-grained classification datasets including **CUB [42]** (200 classes), **Flowers 102 [23]** (102 classes), **Stanford Cars [14]** (196 classes), **FGVC Aircrafts [18]** (100 classes) and **Food101 [1]** (101 classes). We also apply our method on some coarser datasets including **EuroSAT [10]** (10 classes), **ImageNet [31]** (1000 classes), **CalTech101 [5]** (100 classes), **DTD [4]** (47 classes), **Oxford Pets [26]** (37 classes), **Sun397 [44]** (397 classes) and **UCF101 [36]** (101 classes). For all these datasets, we use the first half of the classes (ordered by ids of the original dataset) for training and second half for zero-shot testing.

We also use **NABirds [39]** which contains 404 bird classes at species level. We remove the overlapping classes of the CUB testing set from these to obtain 331 training classes. Along with train and test classes being different, this setting also represents a *distribution shift* in the images of training and testing as images for CUB and NABirds have been obtained in different manners.

iNaturalist [41] 2021 is another dataset we utilize to illustrate that our method scales and generalizes. iNaturalist contains 10k classes belonging to 11 general categories (such as birds, plants, fishes). First, in a similar setting to NABirds we remove overlapping test classes of CUB to train a model for testing on CUB. Secondly, we remove all bird classes from iNat and train a model on the remaining classes to test on CUB. We follow similar settings for testing on Flowers 102. Even in these challenging circumstances our method offers improvement over the baselines(§ 5.3).

For CUB, NABirds and iNaturalist we also have taxonomy information including family, order and scientific name. We also append separate texts containing these to the category-wise text corpus to show improvements (Tab. 3).

NeWT [40] provides a benchmark for a set of 164 complex binary classification tasks in the natural world that extend beyond species classification. These tasks include determining 1) appearance 2) behavior, 3) context, 4) counting and 5) gestalt. NeWT contains 36k images with 200-400 images per task. We randomly select 50 of the 164 tasks to evaluate our trained model. We manually associate two texts for each task, positive and negative. For example, "a photo of a raptor bird which is not on a utility pole" and "a photo of a raptor bird which is on a utility pole". We show improvements over CLIP (§ 5.5). All details of texts used as well as categories selected are in the Appendix.

4.2. Implementation Details

For **generating category-level texts** for training, we utilize the "gpt-4-0613" API. We set the temperature parameter as 0 so that texts generated are deterministic.

For all queries concerning the classes of iNaturalist (both visual and location) we also append the type of organism as well as it's scientific name in the question. For example, for the class "Bay Laurel" the query for location information is

> Where can we find a Bay Laurel, a type of plant with scientific name Laurus nobilis? Produce a list of habitat and geographic location information that can be used to identify the plant.

This is required because there exist organisms with the same common name but different domains. Also, we need to append the scientific name as otherwise GPT4 does not recognise the organism in many cases. We provide more details in Appendix.

Additionally, we experiment with using **taxonomy information** for training and testing on datasets where it is available (Tab. 3). We form the following texts

> 1. a photo of `[class]` `[domain]`, with scientific name `[s_name]`
> 2. a photo of `[class]` `[domain]`, with family name `[family]`
> 3. a photo of `[class]` `[domain]`, of the order `[order]`

While **fine-tuning** using the texts obtained from an LLM, we train for only 15 epochs on each dataset. On iNaturalist, we train for only 5 epochs. We find hyperparameters by splitting the train classes into two equal parts, training on the first half, and validating on the second. We then fix the best hyperparameters found and train on all train classes.

The CLIP architecture consists of an image encoder and a text encoder. Both contain transformers followed by a linear projection layer at the end. We use different learning rate and weight decay for the projection layers compared to rest of the encoders. The temperature parameter τ in our model (§ 3.2) is trainable. We provide the details of the initialization of τ, the momentum parameter for the EMA encoder as well as learning rates and weight decays of every parameter for all datasets in the Appendix.

Methods	CUB	Stanford Cars	FGVC Aircrafts	Flowers 102	Food 101
CLIP	50.54	69.72	29.27	71.78	88.32
CLIP + A	50.71	69.47	30.35	75.37	90.08
CLIP^{FT}	50.81 $\pm$ 0.04	69.61 $\pm$ 0.07	31.10 $\pm$ 0.02	73.68 $\pm$ 0.00	88.32 $\pm$ 0.00
CLIP^{FT} + A	**53.34** $\pm$ 0.08	**71.63** $\pm$ 0.06	**36.41** $\pm$ 0.02	**77.05** $\pm$ 0.00	**93.71** $\pm$ 0.01

Table 1. **Comparison with CLIP ViT-B/32 on zero-shot performance on fine-grained domains.** We compare our method CLIP^{FT} + A to the baselines defined in § 4.3. We significantly improve over baseline CLIP evaluated with both "a photo of a `[class] [domain]`" and LLM attributes. We also fine-tune CLIP with only "a photo of a `[class] [domain]`" text and compare with our method to show that our improvements are not due to seeing domain-specific images but also by learning correlations between images and fine-grained attributes.

	Stanford Cars	FGVC Aircrafts	Flowers 102	EuroSAT	Food 101	ImageNet
CLIP	74.94	36.47	77.05	64.05	92.49	67.41
CLIP + A	73.83	36.47	80.84	71.51	93.72	69.74
CLIP-A-self [19]	72.90	33.00	75.30	70.50	91.20	68.30
CLIP^{FT} + A	**75.78**	**40.75**	**81.26**	**81.82**	**95.08**	**71.87**

Table 2. **Comparison to prior work using ViT B/16 architecture on zero-shot classification.** We show that across a variety of datasets from finer to coarser domains we considerably boost performance over baselines. Here we train using only 16 images per class and test on **unseen classes** for fair comparison to CLIP-A-self. We do not compare on CUB dataset as CLIP-A-self uses a 3:1 split on CUB, whereas we use 1:1 across all datasets.

Texts	CUB	Flowers 102
Visual	53.34	77.05
Taxonomy	53.07	-
Habitat	53.69	76.00
Vis. + Hab.	54.01	77.22
Vis. + Tax. + Hab.	**54.23**	-

Table 3. **Evaluating CLIP^{FT} + A using different types of text.** We query LLMs to produce visual (vis.) and habitat (hab.) information separately and use taxonomy (tax.) information available with dataset. We train with the type of text specified in each row and test with the same type. Using habitat information works slightly better than using visual information for CUB. All three types of texts are complementary.

Train Set	CUB	Flowers 102
NABirds\overlap	55.32	-
iNat\overlap	54.58	77.05
iNat\Birds	53.89	-
iNat\Plants	-	76.63

Table 4. **Evaluating domain transfer performance.** Our method offers substantial gain over baseline CLIP (Tab. 1) even when trained on external datasets. Performance boost is competitive even when removing all bird or plant classes from iNat to test on CUB and Flowers respectively.

4.3. Baselines

In this section, we discuss the various methods for which we compare zero-shot classification accuracy.

CLIP refers to pre-trained CLIP tested with "a photo of a `[class] [domain]`" texts like the original paper.

CLIP + A is evaluating pre-trained CLIP with attributes obtained from LLMs as outlined in § 3.3.

CLIP^{FT} involves fine-tuning CLIP on training classes using "a photo of `[class] [domain]`" texts and evaluating on test classes using "a photo of `[class] [domain]`" texts.

CLIP^{FT} + A is our method where we fine-tune CLIP (§ 3.2) using attributes obtained from a LLM (§ 3.1) for training classes and evaluate using LLM attributes of testing classes at test time as described in § 3.3.

Testing Texts	CLIP + A	CLIP^{FT} + A
`[class] [domain]`	50.54	52.29
GPT 4 Vis.	50.71	53.34
GPT 3 Vis.	51.08	53.35
LLaMA Vis.	50.10	52.52
Ground Truth Vis.	52.53	53.99
GPT 4 Vis. + Tax. + Hab.	52.83	54.23
GPT 3 Vis. + Tax. + Hab.	52.63	53.58
LLaMA Vis. + Tax. + Hab.	50.85	52.64

Table 5. **Evaluating model trained using GPT4 with texts obtained from other models.** Our model consistently improves over pre-trained CLIP when evaluated with texts obtained from different LLM models (GPT3.5 and LLaMA2-7B) as well as GT aggregated captions and "a photo of a `[class] [domain]`".

CLIP-A-self [19] is prior work which uses text obtained from GPT to train an adapter network attached after the text and image encoders of CLIP. For comparing with this, we use the numbers stated by them under their training and evaluation scheme. We test our model on the same classes as them to show improvement.

5. Results

In this section, we compare our method to baselines and evaluate it under various settings. We discuss our performance improvements over various datasets and architectures. We show that for natural domains using taxonomy and habitat information offers improvements with habitat information especially being a strong factor. Our model scales across architectures and needs only a few epochs of training. We further show that our method performs better than baseline CLIP even under more difficult evaluation settings such as 1) using texts from different LLM models during testing and training; 2) training a model in a do-

main very different from testing domain, and 3) evaluating on tasks other than identifying categories at test time. Additionally, we discuss other training strategies for category-level fine-tuning and how they perform.

5.1. Comparison with Baselines

Tab. 1 compares our method to three baselines CLIP, CLIP + A and CLIP^{FT}, all evaluated on **unseen classes**. Here we use the ViT B/32 architecture for all methods. Our method offers considerable improvements over pre-trained CLIP when using "a photo of a `[class] [domain]`" text and when using GPT generated text. In difficult fine-grained domains such as CUB, Stanford Cars and FGVC Aircrafts pre-trained CLIP does not utilize text attributes generated by GPT resulting in negligible improvement (decrease on Stanford Cars) compared to "a photo of a `[class] [domain]`" (see CLIP + A vs CLIP). This motivates the need to fine-tune using these attributes, resulting in significant improvement across all datasets. We also compare to fine-tuning CLIP with "a photo of a `[class] [domain]`" texts (CLIP^{FT}) to show that the improvement our method achieves is not due to just being trained on images of concerned domain.

We compare our method to previous work CLIP-A-self [19] in Tab. 2. We follow [19] and use the ViT B/16 architecture and only 16 images per class for training. Again we evaluate on **unseen classes**. Our method outperforms CLIP-A-self significantly across all datasets. Also, our method offers substantial improvement over CLIP, showing that it **scales across architectures**. We discuss why CLIP-A-self underperforms in detail in the Appendix. We also show results on the **14 datasets** benchmark in Appendix.

5.2. Using more than just Visual Information

We explore using information other than visual attributes for natural domains such as birds and flowers. For humans identifying a bird in an image it is crucial to know where the image was taken, because that reveals habitat and location information. We therefore query GPT for an organism's habitat and geographic range. In Tab. 3 we show that for CUB using **only habitat information performs better than using only visual information**. A reason for this is that habitat information describes the background of the images of birds, which is helping to differentiate between categories. We also show that combining visual + taxonomy + habitat information for CUB and visual + habitat information for Flowers102 offers best improvement.

5.3. Training on External Domains

We now evaluate under more difficult settings. We train and test on different datasets, always removing any overlapping classes. For training on iNat and NABirds we use all visual + taxonomy + habitat information. While testing on CUB we use visual + taxonomy + habitat. For Flowers 102 we use visual + habitat information. Tab. 4 we show the accuracy on CUB test classes improves considerably when training using NABirds and iNat even though the images of these datasets have a distribution shift w.r.t. CUB. More strikingly, we show that even when we remove all bird classes from iNat we still offer improvement on CUB test classes compared to CLIP + A (52.83 $\rightarrow$ 53.89). Similarly when we remove all plant classes from iNat, we still get improvement on Flowers 102 test set. This proves that our **model is also able to generalize well**. It is learning to associate fine-grained attributes to images irrespective of the domain differences in training and testing.

5.4. Using Novel Texts during Test Time

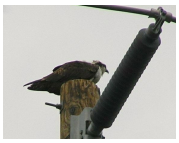
We evaluate how our model would perform in the absence of the LLM used to generate training texts, during test time (Tab 5). We use GPT3.5 turbo (0613) and LLaMA2-7B [38] for generating visual and habitat texts. We show that our method **consistently improves performance over pre-trained CLIP for all types of texts** explored. [1] Please refer to Appendix for examples of texts produced. Our model also improves performance over pre-trained CLIP while using "a photo of a `[class] [domain]`" texts.

Reed *et al*. [30] present a dataset of human-labelled captions per image for the CUB dataset. We use these **ground truth texts at test time** to evaluate our model. The dataset contains 10 captions for every image, which are all visual attributes of the bird in the image. Since many attributes in the captions are repeated for each image as well as across images of the same class, and to limit the size of the text corpus, we randomly select one caption per image of a given class and aggregate them to form a category corpus. In Tab. 5 last row, we notice that CLIP does better using these image-level GT texts compared to using GPT4 Visual texts which were category-level (row 2). However, our method still outperforms showing that it is able learn meaningful attributes through noisy labels.

5.5. Evaluation on Novel Tasks

We use the NeWT [40] benchmark to evaluate on tasks beyond categorization. These include identifying age, attribute, health, photo quality, species, context and behavior. We evaluate the model and baseline CLIP using average Mean Average Precision (MAP) across tasks. Our model trained on iNaturalist using visual + taxonomy + location information outperforms baseline CLIP: 60.25 vs 61.90 MAP $\rightarrow$ **4.1% relative error reduction**. We present all tasks and texts we use to evaluate as well as MAP per task in the Appendix. Below is an example prediction:

[1] LLaMA model does not always produce texts in the specified format and thus needs post-processing. The texts formed finally are considerably different grammatically from the sentences our model has been trained on.



a photo of a raptor bird which is not on a utility pole

Probability: 0.38

a photo of a raptor bird which is on a utility pole

Probability: 0.62

5.6. Resource Requirements

We fine-tune our model using a 1024 batch size on a single NVIDIA A100 80GB. We need to train only for a maximum of 15 epochs. For smaller datasets such as CUB, FGVC Aircraft, and Stanford Cars, this takes less than 5 minutes. For iNaturalist, which is a large dataset, we train only for 5 epochs, taking about 4 hours.

The cost of using the GPT-4 API to query text descriptions for a dataset with 100-200 classes (such as CUB) is about $1-$5. This is low because it scales with the number of categories and not the number of images. Generating captions per image is both time-consuming and expensive; an estimate for doing this for the CUB dataset using the GPT-4 Vision API is more than $100. This cost scales significantly for larger datasets with more images.

5.7. Alternate Training Strategies

Here we discuss other training strategies for improving VLMs using category-level training data. Our ***simple fine-tuning strategy*** of stochastically pairing images with texts within categories **is simple, efficient, and offers similar improvements** compared to more complex approaches.

Firstly, since we pair a given image of a category to every text of that category, we might be pairing texts that describe attributes that are not visible. For CUB dataset we have the **ground truth visibility annotations** of various bird parts, which we use to ignore texts that are occluded for each image. This strategy offers improvement to our scores (54.23 $\rightarrow$ 54.47). However, this depends on visibility information that is time-consuming to generate.

The next step is to assume that pre-trained CLIP itself is able to correctly identify if a part is visible. Assuming this we **mask texts during training time based on CLIP predictions** by doing 1) a forward pass for a image and all texts of the category to find the texts above a threshold that can be paired with the image, 2) max pooling at instance level for images and texts. We find that none of these strategies offer any improvement and that pairing images with low scoring texts also (like in our method) is improving performance because CLIP does not accurately identify which fine-grained attributes correctly correspond to given image.

We also try well-known semi-supervised learning strategies such as FixMatch [35] and knowledge distillation [3]. We find that these offer small ($< 0.2\%$) to no improvements over our method. Please see the Appendix for details of implementation of all the methods and accuracies.

5.8. Performance of image captioning models.

We test the recently released GPT4 Vision API for checking quality of image captions obtained. Even though it performs better than previous captioning models such as LLaVA [17] and BLIP2 [16], we find that the captions obtained are general descriptions without fine-grained details. The captions are specific to the image but do not describe information helpful to identify the category. An example is:

A slender, streaked brown songbird with keen eyes and a pointed beak perches atop a weathered wooden fence post amidst a backdrop of natural grassland under a clear blue sky.

For this image of a Vesper Sparrow, GPT4 provides a general description of the bird and suggests the presence of clear blue sky which is not visible in the image. We provide more detail including prompt and examples in Appendix.

6. Limitation

Since our method is trained on texts generated by LLMs, it is important to verify the correctness of these texts to assess the level of noise in our training dataset. As described in § 3.1, we conduct spot checks across some categories on our evaluation sets. However, for our larger training datasets, vetting becomes impractical. Improved performance on human-generated texts on CUB, as well as the vetted evaluation sets, supports our model's ability to learn meaningful information from somewhat noisy training data.

7. Conclusion

We present a method to improve the zero-shot performance of VLMs using attributes generated by LLMs on fine-grained domains. Our evaluation strategy involves testing the trained model on unseen classes, texts generated from different LLMs as well as humans, dissimilar domains, and novel tasks. We show that fine-tuning CLIP using category-level descriptions from GPT4 significantly improves performance compared to baselines in this challenging downstream evaluation framework. Our findings suggests that habitat and geographic priors are equally effective and complementary to visual information for zero-shot classification in natural domains. We publicly release our benchmark across all 14 datasets.

8. Acknowledgements

We thank Aaron Sun, Gustavo Perez, Rangel Daroya, and Mustafa Chasmai for participating in the verification of attributes generated by GPT4. The project is supported in part by NSF Grants #2329927 and #1749833. Our experiments were performed on the GPU cluster funded by the Mass. Technology Collaborative.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 5

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[3] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020. 2, 8

[4] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5

[5] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Pattern Recognition Workshop*, 2004. 5

[6] Andreas Fürst, Elisabeth Rumetshofer, Johannes Lehner, Viet T Tran, Fei Tang, Hubert Ramsauer, David Kreil, Michael Kopp, Günter Klambauer, Angela Bitto, et al. Cloob: Modern hopfield networks with infoloob outperform clip. *Advances in neural information processing systems*, 35:20450–20468, 2022. 2

[7] Yulu Gan, Yan Bai, Yihang Lou, Xianzheng Ma, Renrui Zhang, Nian Shi, and Lin Luo. Decorate the newcomers: Visual domain prompt for continual test time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7595–7603, 2023. 2

[8] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, pages 1–15, 2023. 3

[9] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19338–19347, 2023. 3

[10] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification, 2017. 5

[11] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 1, 2

[12] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 2

[13] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 2

[14] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 1, 5

[15] Kathleen M Lewis, Emily Mu, Adrian V Dalca, and John Guttag. Gist: Generating image-specific text for fine-grained object classification. *arXiv e-prints*, pages arXiv–2307, 2023. 3

[16] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 1, 8

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 8

[18] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 5

[19] Mayug Maniparambil, Chris Vorster, Derek Molloy, Noel Murphy, Kevin McGuinness, and Noel E O'Connor. Enhancing clip with gpt-4: Harnessing visual descriptions as prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 262–271, 2023. 3, 6, 7, 17

[20] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *arXiv preprint arXiv:2210.07183*, 2022. 3

[21] 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 preprint arXiv:2305.18287*, 2023. 3

[22] Muhammad Ferjad Naeem, Muhammad Gul Zain Ali Khan, Yongqin Xian, Muhammad Zeshan Afzal, Didier Stricker, Luc Van Gool, and Federico Tombari. I2mvformer: Large language model generated multi-view document supervision for zero-shot image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15169–15179, 2023. 3

[23] 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. 1, 5

[24] Zachary Novack, Julian McAuley, Zachary Chase Lipton, and Saurabh Garg. Chils: Zero-shot image classification with hierarchical label sets. In *International Conference on Machine Learning*, pages 26342–26362. PMLR, 2023. 2

[25] Omiros Pantazis, Gabriel Brostow, Kate Jones, and Oisin Mac Aodha. Svl-adapter: Self-supervised adapter

for vision-language pretrained models. *arXiv preprint arXiv:2210.03794*, 2022. 3

[26] 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. 5

[27] Fang Peng, Xiaoshan Yang, Linhui Xiao, Yaowei Wang, and Changsheng Xu. Sgva-clip: Semantic-guided visual adapting of vision-language models for few-shot image classification. *IEEE Transactions on Multimedia*, 2023. 3

[28] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15701, 2023. 3

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 4

[30] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 49–58, 2016. 7

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[32] 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. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022. 2

[33] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, 2022. 2

[34] Mainak Singha, Harsh Pal, Ankit Jha, and Biplab Banerjee. Ad-clip: Adapting domains in prompt space using clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4355–4364, 2023. 3

[35] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 2, 8

[36] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[37] Changyao Tian, Wenhai Wang, Xizhou Zhu, Jifeng Dai, and Yu Qiao. Vl-ltr: Learning class-wise visual-linguistic representation for long-tailed visual recognition. In *European Conference on Computer Vision*, pages 73–91. Springer, 2022. 3

[38] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 7

[39] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 595–604, 2015. 1, 5

[40] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12884–12893, 2021. 2, 5, 7

[41] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 1, 5

[42] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 1, 5

[43] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022. 3

[44] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, June 2010. 5

[45] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 2

[46] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 3

[47] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 3

[48] Xin Zhang, Shixiang Shane Gu, Yutaka Matsuo, and Yusuke Iwasawa. Domain prompt learning for efficiently adapting

clip to unseen domains. *arXiv preprint arXiv:2111.12853*, 2021. 3

[49] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 2

[50] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2

[51] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669, 2023. 2