Generate, Transduct, Adapt: Iterative Transduction with VLMs

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

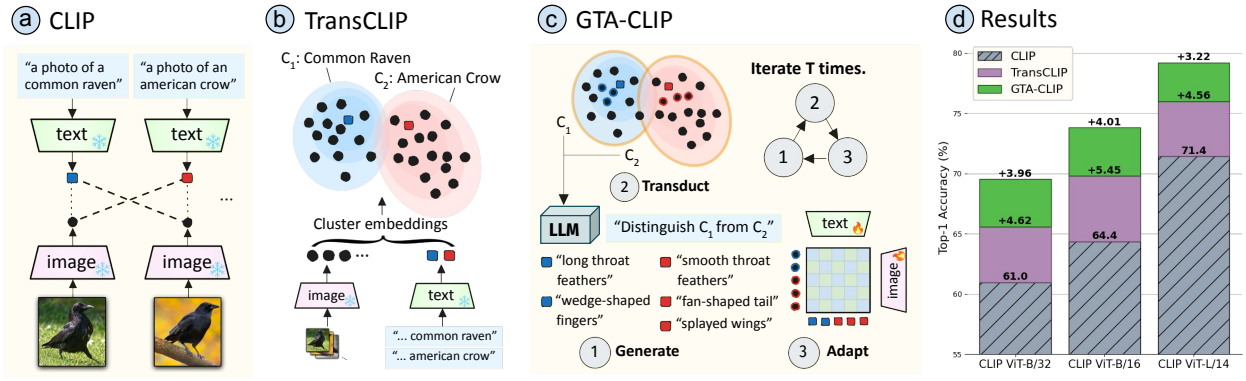


Figure 1. **Overview of GTA-CLIP. (a)** Vision-language models (VLMs) such as CLIP [36] enable zero-shot classification using similarity between text embeddings of class prompts and images. **(b)** Transductive CLIP [61] exploits the structure of the entire image dataset to assign images to classes improving accuracy. **(c)** Our approach, GTA-CLIP, iteratively (i) induces structure over the classes in language space by generating attributes driven by the pairwise confusions, (ii) performing attribute-augmented transductive inference, and (iii) adapting CLIP encoders using the inferred labels. **(d)** Across 12 datasets we improve upon CLIP and transductive CLIP by 9.5% and 4.0% using VIT-B/16, and similarly for other encoders. Significant improvements are also reported in the few-shot setting.

Abstract

Transductive zero-shot learning with vision-language models leverages image-image similarities within the dataset to achieve better classification accuracy compared to the inductive setting. However, there is little work that explores the structure of the language space in this context. We propose GTA-CLIP*, a novel technique that incorporates supervision from language models for joint transduction in language and vision spaces. Our approach is iterative and consists of three steps: (i) incrementally exploring the attribute space by querying language models, (ii) an attribute-augmented transductive inference procedure, and (iii) fine-tuning the language and vision encoders based on inferred labels within the dataset. Through experiments with CLIP encoders, we demonstrate that* GTA-CLIP *yields an average performance improvement of 9.5% and 4.0% across 12 datasets and 3 encoders, over CLIP and transductive CLIP respectively in the zero-shot setting. We also observe similar improvements in a few-shot setting. We present ablation studies that demonstrate the value of each step and visualize how the vision and language spaces evolve over iterations driven by the transductive learning.*

1. Introduction

Recent advances in vision-language models (VLMs) have enabled zero-shot image classification across diverse domains. These models, such as CLIP [36], assign images to classes based on the similarity between image and text embeddings, forming the basis of various zero-shot approaches in classification [63, 66, 67], segmentation [20, 21, 37, 54], and detection [27, 65] (Fig. 1a). However, in many practical scenarios, the images requiring classification are known in advance. For example, an ecologist might have a large collection of animal images that need to be categorized by species. In such cases, *transductive inference* is more suitable, as it leverages the dataset's inherent structure to refine predictions (Fig.1b).

Despite the success of transductive inference with VLMs, existing approaches often overlook the rich structure of the label space derived from language. For instance, linking semantically similar descriptions or attributes can yield more coherent class prototypes, while aligning these attributes with image features can enable model *adaptation* on the specific dataset. This strategy can be advantageous for zero-shot and few-shot recognition, especially in novel

or fine-grained domains where labeled data is scarce.

To address this gap, we propose GTA-CLIP, a transductive learning approach that exploits structure in both the language and vision spaces (Fig.1c and Alg.1). Our method begins by querying a language model to populate the language space: starting with an initial set of attributes per category, we dynamically expand this space by *generating* discriminative attributes based on pairwise confusion between classes. This strategy improves class separation while maintaining computational tractability. We then design a *transductive inference* procedure that refines predictions using these attributes. Finally, we *adapt* the underlying VLM to the target dataset using inferred labels and attributes. This iterative cycle of Generation, Transduction, and Adaptation—hence the name GTA—progressively improves recognition performance.

We present experiments on a benchmark of 12 datasets using various CLIP encoders, where our approach achieves 8.6% improvement over CLIP and 3.7% improvement over the current state-of-the-art transductive CLIP [61] on average (Fig. 1d and Table 1). Notably, on a dataset like CUB with about 12k images, the whole process completes in 12-20 minutes on a single A100 GPU (see § 5.6). Ablation studies demonstrate that each component of our method contributes to these gains. Specifically, while attribute-augmented transduction improves performance on average, it is most effective when paired with model fine-tuning. Similarly, dynamically expanding the attribute space benefits fine-grained domains while keeping learning efficient. We visualize how the language and vision spaces evolve over iterations providing insights into the performance improvements. Further, we demonstrate GTA-CLIP's advantages in a few-shot setting (Table 2) and a zero-shot setting where labeled examples from related categories are available during training (Table 3). In both cases, our approach outperforms transductive CLIP [61] and prior methods.

To summarize, our main contribution is to demonstrate that zero- and few-shot classification can be significantly improved in a transductive setting by integrating attribute generation, transductive inference, and model adaptation into a unified framework. While prior work has explored these components in isolation, to the best of our knowledge, this is the first work to show that their benefits are complementary and can be effectively leveraged in label-scarce scenarios. Our approach is of practical value as it provides end users with another avenue to improve labeling accuracy on their target dataset, alongside traditional labeling efforts. Our code is released at `https://github.com/cvl-umass/GTA-CLIP`.

2. Related Work

Transductive learning [46] is well-suited for scenarios where a model's predictions must be accurate on a specific dataset rather than on unseen future data. Access to the entire unlabeled test set enables inference through methods such as label propagation [10, 50], clustering [48, 50, 59], among others [4, 19]. This setting closely resembles semi-supervised learning, where techniques like pseudo-labeling [3, 6, 62], entropy minimization [13], and self-training [53, 56] have proven effective.

Zero-shot transduction has been previously explored using image generation [12, 49] and attribute-based approaches [55, 60], while more recent methods leverage VLMs to estimate initial class prototypes from language. For example, ZLaP [16] improves CLIP through label propagation, while [25] iteratively estimate assignments and class prototypes. TransCLIP [61] presents an efficient approach for large-scale zero-shot transduction, employing a block majorization-minimization (BMM) algorithm [15, 38] to optimize an objective comprising: a Gaussian mixture model, a Laplacian regularizer, and a KL divergence term that aligns assignments with image-text probabilities across the dataset. *We extend this state-of-the-art by incorporating class-specific attributes in the KL divergence term, enabling better alignment of image features with the semantic structure of the dataset.*

Improving Zero-shot with Attributes. Large language models (LLMs) have been used to improve zero-shot classification by expanding attribute spaces beyond simple class names. For example, [26, 29, 35] employ LLMs to generate rich category descriptions (e.g., describing a tiger as having stripes and claws) to improve both classification accuracy and interpretability in CLIP-based models. Beyond LLM-based augmentation, other approaches focus on identifying a concise and discriminative set of attributes for recognition [7, 57]. Inspired by both strategies, we leverage language models such as GPT [1] and LLaMA [45] to initially populate the attribute space. *However, rather than relying on static expansions, we introduce a dynamic refinement process: attributes are iteratively added to classes that are frequently confused, improving class separability.*

Adapting CLIP. Prior work has shown that augmenting CLIP with attributes does not significantly improve zero-shot recognition, particularly in out-of-domain or fine-grained datasets. In such cases, model adaptation is necessary. Existing techniques range from learning language and vision prompts [66, 67] to incorporating learnable layers [11, 63] or performing full fine-tuning [42, 44, 51, 64]. A different line of work addresses fine-tuning without paired image and text data. WiSE-FT [51] and LaFTer [28] use ensembles, while others [18, 24, 40] show the value of large-scale fine-tuning with image-text data aligned at the category level. We build on the approach of AdaptCLIPZS [40], which stochastically pairs images with attributes within a category and modifies the CLIP objective to accommodate weaker supervision. *However, while all the*

above approaches rely on labeled examples, such as images with class labels on the target domain, our method enables adaptation without any annotated data.

Our key contribution is unifying attribute generation, model adaptation, and transductive inference within a single framework for zero- and few-shot classification. Iterative and stage-wise learning can be viewed as the optimization of a single objective, enabling both label inference within the dataset and end-to-end fine-tuning of the underlying VLMs on target domains (Alg. 1). While these ideas have been explored individually, their integration is novel and leads to significant improvements over the current state-of-the-art across diverse datasets, with minimal additional computational requirements.

3. Methodology

The input to our approach is a set of images $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ and a set of classes $\mathcal{Y} = \{y_i\}_{i=1}^M$. In the zero-shot setting the goal is to assign each image to one of the M classes. In the few-shot setting we are also provided with a few labeled examples $\mathcal{D}_{train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^K$ with $\mathbf{x} \in \mathcal{X}_{train}$, $\mathcal{X}_{train} \cap \mathcal{X} = \emptyset$, and $y \in \mathcal{Y}$.

We also consider a setting where labeled data comes from a different set of classes, i.e., $y \in \mathcal{Y}_{train}$ where $\mathcal{Y}_{train} \cap \mathcal{Y} = \emptyset$. This setup is used in approaches where labeled data from a set of base categories is used to adapt CLIP on the target domain.

We report the mean per-class accuracy on the target set of images $\mathcal{X}$ given their ground-truth labels. To enable zero-shot learning we assume an image encoder $\theta(\cdot)$ and a text encoder $\phi(\cdot)$ such that $\theta(\mathbf{x})^\top \phi(\mathbf{y})$ is high for image $\mathbf{x}$ and text $\mathbf{y}$ pairs that are similar. We experiment with a variety of encoder pairs based on CLIP framework. In addition we assume access to a language model (e.g., Llama3 or GPT-4o) which we can query to generate attributes for each class.

3.1. GTA-CLIP formulation

GTA-CLIP maintains a list of attributes indexed by class denoted by $\mathcal{A} = (\mathcal{A}_j)_{j=1}^M$, where $\mathcal{A}_j = \{\mathbf{a}_{j,k}\}_{k=1}^{n_j}$ denotes the set of text attributes for the class j. The number of attributes n_j can vary across classes. Like the TransCLIP [61] formulation we maintain $\boldsymbol{\mu} = (\boldsymbol{\mu}_j)_{j=1}^M$ and $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_j)_{j=1}^M$ denoting the Gaussian mixture model (GMM) mean and diagnonal variance for each class.

In addition we maintain a matrix of softmax class assignments $\mathbf{z} \in [0,1]^{N \times M}$, where N is the number of query images and M is the number of classes. In other words $\mathbf{z}_{i,\cdot} \in \Delta_M$ reflects the probability of assignment over all the classes, where Δ_M is the M-dimensional probability simplex. Given a class $j \in \mathcal{Y}$ the vertical slices $\mathbf{z}_{\cdot,j} \in [0,1]^N$ represents the probability that a specific query image belongs to class j. After inference the class label for each image i can be obtained as $\operatorname{argmax}_j \mathbf{z}_{i,j}$.

Algorithm 1 GTA-CLIP

Require: Query images $\mathcal{X}$, list of classes $\mathcal{Y}$, list of initial attributes indexed by class $\mathcal{A}$, image encoder θ, text encoder ϕ, number of iterations T.
Ensure: Fine-tuned image and text encoders θ, ϕ, labels $\mathbf{z}$, class prototypes $\boldsymbol{\mu}, \boldsymbol{\Sigma}$, attributes indexed by class $\mathcal{A}$.

$\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow 0$
for $t \leftarrow 1$ to T **do**
  ▷ *mine attributes*
  $\mathcal{A} \leftarrow \text{GENERATEATTRIBUTES}(\mathcal{Y}, \mathcal{A}, \theta, \phi)$
  ▷ *transductive assignment with attributes*
  $\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow \text{TRANSDUCT}(\mathcal{X}, \mathcal{Y}, \mathcal{A}, \theta, \phi)$
  ▷ *fine-tune image and text encoders*
  $\theta, \phi \leftarrow \text{ADAPT}(\mathcal{X}, \mathcal{Y}, \mathbf{z}, \theta, \phi)$
end for
return $\theta, \phi, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathcal{A}$

Zero-shot Setting. The overall objective in this formulation is:

$$\mathcal{L}_{\text{zero-shot}}(\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{y}, \theta, \phi, \mathcal{A}) = \underbrace{-\frac{1}{N}\sum_{i=1}^N \mathbf{z}_i^\top \log(\mathbf{p}_i)}_{\text{Clustering objective}} \underbrace{-\sum_{i=1}^N \sum_{j=1}^N w_{i,j} \mathbf{z}_i^\top \mathbf{z}_j}_{\text{Laplacian regularizer}} + \underbrace{\sum_{i=1}^N \mathbf{KL}_\lambda(\mathbf{z}_i || \hat{\mathbf{y}}_i)}_{\text{Agreement with text}}. \quad (1)$$

The first term is a clustering objective under a Gaussian assumption for each class, and $\mathbf{p}_i = (p_{i,j})_{j=1}^M \in \Delta_M$ denotes the probability over classes for the image $\mathbf{x}_i$. Let $\mathbf{f}_i = \theta(\mathbf{x}_i)$, then this is defined as:

$$p_{i,j} \propto \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{f}_i - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{f}_i - \boldsymbol{\mu}_j)\right). \quad (2)$$

The second term is a Laplacian regularizer commonly seen in spectral clustering [30, 41] and semi-supervised learning settings [2, 58]. Here $w_{i,j}$ denotes the affinity between images $\mathbf{x}_i$ and $\mathbf{x}_j$, and this term encourages images with high affinity to have similar predictions $\mathbf{z}$. We set $w_{i,j} = \max(0, \mathbf{f}_i^\top \mathbf{f}_j)$ resulting in a positive semi-definite affinity matrix $\mathbf{W} = [w_{i,j}]$ and faster optimization procedure due to a convex relaxation.

The KL divergence term ensures alignment of predictions with text and is defined as:

$$\mathbf{KL}_\lambda(\mathbf{z}_i || \hat{\mathbf{y}}_i) = \mathbf{z}_i^\top \log \mathbf{z}_i - \lambda \mathbf{z}_i^\top \log \hat{\mathbf{y}}_i; \;\; \lambda > 0. \quad (3)$$

The text based predictions $\hat{\mathbf{y}}_i$ are obtained as softmax over the mean similarity between the image and the attribute

embeddings $\mathcal{A}_j = \{\mathbf{a}_{j,k}\}_{k=1}^{n_j}$

$$\hat{\mathbf{y}}_{i,j} = \frac{\exp(\bar{s}_{i,j})}{\sum_{j=1}^{M}\exp(\bar{s}_{i,j})}, \text{ where } \bar{s}_{i,j} = \frac{1}{n_j}\sum_{k=1}^{n_j}\theta(\mathbf{x}_i)\phi(\mathbf{a}_{j,k}). \tag{4}$$

The text and vision encoders – θ and ϕ output normalized and temperature-scaled features.

Few-shot Setting. In the few-shot setting we can incorporate the labeled examples $\mathcal{D}_{train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{K}$ by simply setting and fixing their $\mathbf{z}_i$ to the one-hot vector corresponding to the label y_i.

Zero-shot Setting with Seen Classes. In this setting, the labeled examples $\mathcal{D}_{train} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{K}$ come from a set of base classes different from the target classes, i.e., $y \in \mathcal{Y}_{train}$ where $\mathcal{Y}_{train} \cap \mathcal{Y} = \emptyset$, as part of the training set. We first fine-tune CLIP using AdaptCLIPZS on the base classes, followed by transductive inference on only the target images. While this approach does not incorporate the similarity between the training and target images, it provides a straightforward comparison against prior work on adapting CLIP to target domains.

3.2. Optimization

The key difference between our formulation and TransCLIP is that we also update θ, ϕ and $\mathcal{A}$. We initialize $\mathcal{A}$ with the per-class attributes in AdaptCLIPZS, which consists of prompting the 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 `[attribute]`".

where `[domain]` is coarse category, *e.g.* "birds" for CUB [47], `[class]` is the common name of the category, and `[attribute]` is a specific attribute. For example, one such description is *"A bird with a small, round body shape, indicative of a Baird's Sparrow."*

The algorithm iterates between: (1) incrementally generating class-specific attributes to update $\mathcal{A}$ driven by pairwise confusions; (2) attribute-augmented transductive inference to estimate $\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$; and (3) encoder fine-tuning using the inferred $\mathbf{z}$ to update the encoders θ and ϕ. This is outlined in Algorithm 1 and described below.

1. Generating Attributes. Our general strategy is to query large language models (LLMs) to explore the space of attributes driven by pairwise confusions. This is inspired by a long line of work on attribute discovery driven by pairwise discrimination in the computer vision literature [22, 32, 34]. These are appended to the corresponding lists in $\mathcal{A}$. For a given pair of classes, we do this by prompting the LLM as:

> I have a set of attributes for `[class1]` as: `[attrs1]`.
> I have a set of attributes for `[class2]` as: `[attrs2]`.
>
> Provide a few additional attributes for `[class1]` which can help to distinguish it from `[class2]`.
>
> Make sure none of the attributes already given above are repeated. The texts in the attributes texts should only talk about `[class1]` and should not compare it to `[class2]`.

To keep this tractable we only generate attributes for the most confused classes. We first update $\mathbf{z}$ by running attribute-augmented transductive inference given the current model and set of attributes $\mathcal{A}$ (Step 2)[1]. Then, we find the images $\mathbf{x}_i$ for which the difference in the top 2 probabilities in $\mathbf{z}_{i,\cdot}$ is lower than a threshold of α:

$$\mathcal{CC} = \{(i, \{c_1, c_2\}) \mid \mathbf{z}_{i,c_1} - \mathbf{z}_{i,c_2} \leq \alpha; c1 < c2\}.$$

Here c_1 and c_2 are the indices of the top 2 highest probabilities in $\mathbf{z}_{i,\cdot}$. We then find the class pairs $\{c_1, c_2\} \in \mathcal{CC}$ which occur more than β times.

2. Attribute-Augmented Transductive Inference Given the list of attributes $\mathcal{A}$ we can compute the text-driven labels $\hat{\mathbf{y}}_i$ for each class using CLIP encoders θ and ϕ as described in Eq. 4. Optimization of $\mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ can be done using the same formulation of TransCLIP [61]. In particular they propose an iterative procedure where they optimize $\mathbf{z}$ keeping $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ fixed using a Majorize-Minimization procedure (similar to EM) based on a tight-linear bound on the Laplacian term. This results in efficient decoupled updates on $\mathbf{z}$. This is followed by updates on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ keeping the remaining variables fixed using closed form updates. The algorithm converges in a few iterations and allows scaling to large datasets. We refer the reader to the details in [61].

3. Adapting CLIP. We finally fine-tune CLIP encoders θ, ϕ using the current set of attributes $\mathcal{A}$ and the inferred labels $\mathbf{z}$. For each class j we find the top k images with the highest scores based on $\mathbf{z}_{\cdot,j}$. The set of images and corresponding attributes provide a coarse form of supervision for fine-tuning. Specifically, we adopt the objective of AdaptCLIPZS [40] which takes into account class-level supervision and false negative associates since multiple text-image pairs can be considered correctly aligned in a single mini-batch training. For the few-shot setting, we simply include the labeled examples to our samples.

Summary. Algorithm 1 can be viewed as a block coordinate descent optimization of the objective in Eq. 1. While this is straightforward for the continuous variables—such as the GMM, assignments, and encoder parameters—optimization over the space of attributes is challenging due to its inherently discrete, non-differentiable nature.

[1] We find it beneficial to run the transductive step before invoking the generate step (see Appendix Table 9)

Our LLM-guided exploration provides a heuristic motivated by previous work showing that attribute-augmented CLIP improves predictions, thereby improving the **KL** term (if $\mathbf{z}$ is accurate) in Eq. 3. Class-confusion-guided exploration further enriches the attribute space, targeting areas where the model might benefit most. The attributes also provide a better signal for fine-tuning the CLIP to the target domain.

4. Experiments

Datasets. We evaluate GTA-CLIP and compare to previous work on a benchmark of 12 datasets including fine-grained ones like **CUB [47]** (200 classes), **Flowers 102 [31]** (102 classes), **Stanford Cars [17]** (196 classes), **FGVC Aircrafts [23]** (100 classes) and **Food101 [5]** (101 classes). The benchmark also includes datasets such as **EuroSAT [14]** (10 classes), **ImageNet [39]** (1000 classes), **CalTech101 [9]** (100 classes), **DTD [8]** (47 classes), **Oxford Pets [33]** (37 classes), **Sun397 [52]** (397 classes) and **UCF101 [43]** (101 classes).

Evaluation Metrics. For zero-shot evaluation, we assume all test images belong to the target classes, without using any labeled images. In this setting, GTA-CLIP utilizes only the set of unlabeled test images and target categories. For the few-shot setting, we use a few labeled images per class but report accuracy on the test images across all classes, consistent with zero-shot evaluation methods. We also evaluate in the AdaptCLIPZS setting, where half of the dataset classes are considered "seen" and the other half "unseen." Here, the model has access to labeled examples of the seen classes and unlabeled examples from the test set of the unseen classes in a transductive setting. Final accuracy is reported on the test images of the unseen classes. This setup enables comparison with prior work that uses labeled examples from the target domain to adapt CLIP while still measuring performance on future unseen classes.

Implementation Details. To generate attributes, we use Llama-3.1 with a maximum token length of 500. All experiments are run on a single A100 GPU. For each pair of classes, we use the prompt described in § 3.2 to generate attributes. The threshold α for selecting the confused images is set to 0.1, and the hyperparameter β is adjusted so that the cumulative count $\mathcal{CC}$ includes 5% of the most confused images. We run GTA-CLIP for 30 iterations (i.e., $T = 30$ in Algorithm 1) and select the top $k = 8$ images per class for fine-tuning using the labels in $\mathbf{z}$. These parameters remain fixed across all datasets, and we found our approach robust to these choices within a reasonable range (see Appendix for a sensitivity analysis).

For our experiments, we use the ViT-B/32, ViT-B/16, and ViT-L/14 architectures of OpenAI's CLIP models (Performance using CLIP models from Meta are included in the Appendix). Fine-tuning is performed with the AdamW optimizer, using betas of $(0.9, 0.98)$, an epsilon of 1E-6, and a batch size of 32. We set a learning rate of $\gamma =$ 2E-7 and weight decay of $\lambda =$ 1E-4 for the Transformer layers of the image and text encoders, and $\gamma =$ 1E-6 and $\lambda =$ 1E-4 for the final linear projection layers. All results are reported in terms of Top-1 accuracy, averaged over 3 runs.

5. Results

We present results for zero-shot (§ 5.1), few-shot (§ 5.2), and zero-shot with seen classes (§ 5.3) setting on various datasets, followed by ablation studies (§ 5.4) and a detailed analysis of our method (§ 5.5). We also provide further experiments in our Appendix, including similar performance improvements using MetaCLIP (§ 9), sensitivity analysis (§ 10), to show robustness of our method, and detailed visualization of generated attributes (§ 11).

5.1. Zero-Shot Performance

Table 1 shows the zero-shot performance of GTA-CLIP compared to the CLIP [36] and the current state-of-the-art, TransCLIP [61]. We report accuracy across 12 datasets using different CLIP architecture—ViT-B/32, ViT-B/16, and ViT-L/14—along with the overall average accuracy. GTA-CLIP improves over TransCLIP by **3.96%**, **4.01%**, and **3.22%**% and over CLIP by **8.58%**, **9.46%**, and **7.78%** on average using B/32, B/16, and L/14 respectively.

The highest percentage improvements are observed with ViT-B/16, though even the strongest architectures benefit from our method. Food101 [5] is the most challenging, where we see a modest average improvement of **0.14%**. In contrast, EuroSAT [14] shows the greatest improvement, with the highest single-architecture boost (**18.87%** for ViT-B/32) and the highest average improvement across architectures (**13.67%**). GTA-CLIP consistently outperforms both baselines in all settings except one—namely, UCF101 with ViT-L/14. These results demonstrate that our approach is broadly applicable and that reasoning over the attribute space yields significant improvements compared to transductive inference with images alone.

5.2. Few-Shot Performance

Table 2 shows the few-shot performance of our approach compared to TransCLIP. We report results using the ViT-B/16 architecture with 1-shot, 4-shot, and 16-shot settings denoting the number of labeled examples per class. We use the TransCLIP-FS [61] setting for this. Both ours and TransCLIP can incorporate labeled examples by simply setting the corresponding entries in $\mathbf{z}$ to the one hot vector corresponding to their labels, as described in § 3. Performance is reported on the same set of images in the zero-shot setting.

Table 1. **Zero-shot results.** Performance of CLIP, TransCLIP-ZS, and GTA-CLIP across datasets using ViT-B/32, ViT-B/16, and ViT-L/14 architectures. GTA-CLIP outperforms TransCLIP-ZS in all settings except for one – UCF101 with ViT-L/14.

	Method	CUB	Aircraft	Cars	Flowers	EuroSAT	Food	ImageNet	Caltech	DTD	Pets	SUN	UCF	Average
B/32	CLIP	52.33	19.17	60.33	66.91	45.01	80.51	62.06	91.16	42.67	87.44	61.95	62.12	60.97
	TransCLIP-ZS	56.70	20.13	63.57	74.54	58.51	81.38	65.15	91.72	50.59	89.32	67.44	68.01	65.59
	GTA-CLIP	**60.48**	**21.21**	**64.27**	**79.74**	**77.38**	**81.54**	**66.31**	**94.16**	**57.51**	**90.81**	**70.14**	**71.00**	**69.55**
B/16	CLIP	55.20	24.75	65.38	71.38	47.69	86.10	66.72	92.86	43.68	89.13	62.57	66.75	64.35
	TransCLIP-ZS	62.23	26.88	68.87	76.17	65.42	87.15	70.38	92.86	50.00	92.34	68.93	76.34	69.80
	GTA-CLIP	**66.76**	**29.31**	**72.09**	**82.05**	**76.35**	**87.38**	**71.87**	**95.46**	**58.51**	**93.43**	**73.47**	**79.06**	**73.81**
L/14	CLIP	62.03	32.43	76.82	79.54	58.07	90.99	73.48	94.85	53.66	93.62	67.59	74.17	71.44
	TransCLIP-ZS	70.18	35.01	78.50	84.29	69.64	91.88	77.59	95.17	59.69	94.55	73.75	**81.73**	76.00
	GTA-CLIP	**76.56**	**38.58**	**82.29**	**85.87**	**80.83**	**91.91**	**78.54**	**97.36**	**64.89**	**95.83**	**76.65**	81.28	**79.22**

Like the zero-shot case for CLIP ViT-B/16, we find that in every setting and every choice of k-shot, GTA-CLIP outperforms TransCLIP. We find an increase of **3.41%, 3.86%,** and **3.01%** for 1-shot, 4-shot, and 16-shot, respectively. We observe the most pronounced performance increase for 4-shot. Trends of improvements align with the zero-shot setting. Interestingly, we find that **zero-shot GTA-CLIP outperforms 1-shot TransCLIP**, saving human effort, as labeling even a single example per category can be labor-intensive for certain datasets. Furthermore, we find that the gains from transduction, attribute-guided transduction with adaptation (our approach) complement labeling efforts. This flexibility is of practical value, offering end users multiple ways to improve performance on a target dataset.

5.3. Zero-Shot Performance with Seen Classes

Previous work has also evaluated zero-shot learning in a setting where labeled data from a related but different set of classes is available during training, while performance is evaluated on images from unseen classes. Approaches such as CoCoOp [66], AdaptCLIPZS [40] and VDT [24] report results by splitting a dataset's categories in half, treating the first half as "seen" classes to adapt their model and measuring performance on the "unseen" second half. The results are shown in Table 3. Performance tends to be higher in this setting in comparison to the zero-shot and few-shot experiments, as only half of the classes are considered[2] and a smaller domain shift.

For a straightforward comparison, we initialize the CLIP model with the pre-trained weights from AdaptCLIPZS and report the accuracies of TransCLIP and GTA-CLIP on the "unseen" classes of each dataset, using the same framework as the zero-shot setting. Note that this setup does not include a transductive term between training and testing images, thus representing a lower bound on achievable performance. Despite this, we find that GTA-CLIP outperforms prior methods across all five datasets considered: CUB [47], Stanford Cars [17], FGVC Aircraft [23], Flowers102 [31], and Food101 [5].

While AdaptCLIPZS, VDT, CoOp, CoCoOp, and CLIP-A use an inductive setup, TransCLIP and GTA-CLIP adopt a transductive approach that benefits from having test images available in advance. This results in improvements in similar vein as the zero-shot setting. However, even with domain-specific fine-tuning of CLIP with labels, transductive inference proves advantageous, and our attribute-guided approach yields further improvements–an encouraging result. The improvements over CLIP are substantial, though this setup requires more supervision than the previous settings.

5.4. Ablation Studies

We next aim to quantify the performance contributions of each component in Algorithm 1. In Table 4, we selectively disable components of our method and report the average performance over five datasets.

We find that the largest performance gain comes from combining all the components of GTA-CLIP – GENERATEATTRIBUTES which corresponds to *dynamic* attributes in Table 4, TRANSDUCT, and ADAPT with an average of **6.96%** over the considered datasets. However, using *dynamic* attributes without ADAPT but with TRANSDUCT leads to similar performance as using *static* attributes in the same scenario. This shows that fine-tuning the model is necessary to take advantage of the *dynamic* attributes. TRANSDUCT offers an improvement of **3.70%** over baseline inductive CLIP. Adding in ADAPT to this setting results in an improvement of **2.50%** over the strong baseline of TRANSDUCT. We also observe that initializing TRANSDUCT with *static* text attributes offers a gain of **1.34%** over just using "a photo of a `[class]`" texts. Adding only the attributes from GENERATEATTRIBUTES to inductive CLIP offers low improvement (**1.03%**), but when used alongside TRANSDUCT and ADAPT, it increases performance.

[2]Only the CUB dataset has lower performance as the test split is harder than the overall dataset.

Table 2. **Few-shot Results.** Performance (1-shot, 4-shot, and 16-shot) of GTA-CLIP and TransCLIP-FS across datasets using CLIP ViT-B/16 network. We find that GTA-CLIP outperforms TransCLIP-FS in all cases.

	Method	CUB	Aircraft	Cars	Flowers	EuroSAT	Food	ImageNet	Caltech	DTD	Pets	SUN	UCF	Average
1	TransCLIP-FS	65.50	29.84	70.66	85.10	71.43	87.83	69.81	93.18	51.44	91.81	70.59	77.82	72.08
	GTA-CLIP	**68.50**	**31.90**	**71.24**	**92.65**	**80.87**	**88.03**	**71.26**	**93.71**	**60.11**	**94.13**	**73.65**	**79.83**	**75.49**
4	TransCLIP-FS	67.96	35.07	74.14	92.98	78.95	86.35	70.24	93.75	60.50	92.01	71.43	79.25	75.22
	GTA-CLIP	**74.01**	**38.57**	**76.75**	**96.59**	**91.03**	**86.77**	**72.76**	**94.20**	**66.76**	**92.87**	**74.66**	**83.94**	**79.08**
16	TransCLIP-FS	74.24	38.40	79.56	94.68	83.35	86.86	**71.89**	94.20	65.47	92.59	74.81	81.58	78.14
	GTA-CLIP	**78.23**	**43.10**	**81.79**	**97.44**	**91.17**	**86.96**	73.43	**95.94**	**71.55**	**93.20**	**76.31**	**84.62**	**81.15**

Table 3. **Zero-shot Results with Seen Classes.** In this setting examples from a "seen" classes are used to adapt CLIP ViT-B/16 and evaluated on "unseen" classes. We compare both inductive and transductive approaches. Some techniques such as VDT [24] use 3:1 split as opposed to the 1:1 used by other methods on CUB so we do not include their numbers. Tranductive inference remains beneficial, and GTA-CLIP improves over TransCLIP.

Type	Method	CUB	Aircraft	Cars	Flowers	Food
Ind.	CLIP	51.91	36.47	74.94	77.05	92.49
	CoOp [67]	—	22.30	60.40	59.67	82.26
	CoCoOp [66]	—	23.71	73.59	71.75	91.29
	CLIP-A [11]	—	33.50	73.30	71.50	91.20
	VDT [24]	—	33.00	72.90	75.30	91.20
	AdaptCLIPZS	55.63	40.75	75.78	81.26	95.08
Trans.	TransCLIP-ZS	61.98	37.37	78.04	86.45	95.12
	GTA-CLIP	**64.74**	**40.99**	**82.17**	**89.72**	**95.46**

Table 4. **Ablation Study.** Ablation study of the components of GTA-CLIP using ViT-B/16. Average Top-1 accuracy across five datasets is shown (see Appendix for the full table). Attributes $\mathcal{A} = \{\emptyset, \mathrm{S}, \mathrm{D}\}$ refer to *no*, *static*, and *dynamic* attributes, respectively. No attributes corresponds to standard CLIP, while static and dynamic refer to the initial set of attributes and confusion-driven attributes respectively. The first row shows the performance of CLIP, and the third row shows the performance of TransCLIP. Simply generating attributes leads to insignificant improvement in performance on these fine-grained datasets (row 2), but it improves transductive inference and subsequent adaptation. Dynamic attribute generation provides additional benefits (last row).

Attributes	Transduct	Adapt	Acc.	Δ CLIP
Ø	✗	✗	60.56	—
S	✗	✗	61.59	+1.03%
Ø	✓	✗	64.26	+3.70%
S	✓	✗	65.60	+5.04%
S	✓	✓	66.76	+6.20%
D	✓	✗	65.56	+5.00%
D	✓	✓	**67.52**	**+6.96%**

5.5. Class Confusion and Attribute Space

Table 5 presents the top confused class pairs identified by our method during the first epoch on the CUB dataset. We compare these pairs with confusion counts from a linear classifier trained on the full CUB training set using labeled data. The linear classifier is trained on the entire training set of CUB using labels, and we use ground truth labels to estimate its confusion. The comparison with our method, as described in § 3, reveals that 9 out of our top 10 selected confused pairs fall within the top 10% of confused pairs identified by the linear classifier. Overall, there is strong agreement between the most confused pairs, suggesting that the class confusions identified by our approach align well with those from a fully supervised model.

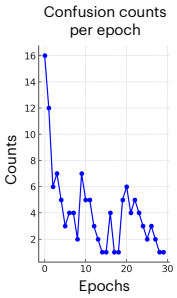
Table 5 also visualizes the progression of confusion counts for the top confused pair (Western Gull, California Gull). The number of images with a probability difference below $\alpha = 0.1$ generally decreases over epochs. The most significant drop occurs between the first and second epochs, highlighting the impact of the newly generated attributes.

Figure 2 illustrates the evolution of the attribute space across various categories. For each category, the class prototype ("photo of [class]"), the initial set of attributes, and the final set of attributes are shown in green, blue, and red, respectively. These visualizations were generated by projecting CLIP text embeddings of the attributes using t-SNE. Several new attributes were added through pairwise comparisons, and a few notable examples are highlighted in the figure. Many of the attributes discovered through pairwise comparison highlight differences in habitat, relative characteristics (e.g., "...more pronounced build compared to its length" for the Western Gull), and other distinguishing features. Bird images often include backgrounds indicative of habitat types, and this form of supervision enables CLIP to learn to associate these attributes with categorization. Larger versions of these figures are in the Appendix.

5.6. Computational Cost

For attribute expansion, we explore both open-source Llama-3.1-8b and GPT-4o, and find their effect on performance to be similar (see Table 8 in Appendix). The number

Table 5. **Class Confusions.** (Left) Progressive reduction of pairwise confusion between "Western Gull" and "California Gull" over training iterations of GTA-CLIP. (Right) Most confused class pairs according to our selection criteria. We show the counts of pairwise misclassified test images according to our procedure and according to a linear classifier trained on the labeled training images using the CLIP image features.



Confused Class Pairs	Ours	Linear
California Gull Western Gull	16	10
Least Flycatcher Olive sided Flycatcher	13	3
American Crow Common Raven	8	12
Least Flycatcher Western Wood Pewee	7	6
Eared Grebe Horned Grebe	7	11
Bronzed Cowbird Shiny Cowbird	6	3
Brewer Sparrow Harris Sparrow	6	2
Slaty backed Gull Western Gull	5	6
Baird Sparrow Grasshopper Sparrow	5	3
Philadelphia Vireo Warbling Vireo	5	8

of class pairs requiring attribute expansion decreases with each epoch, reaching zero for most datasets after about 10 epochs due to the chosen thresholds α and β, ensuring that attributes are not regenerated for duplicate class pairs.

In the CUB dataset, a total of approximately 30 pairs of confusing classes were selected over 30 iterations for prompting the LLM. Across datasets of various sizes, the number of sampled class pairs remains within a similar range, with the Flowers dataset requiring attribute expansion for only three pairs. For larger datasets such as ImageNet, we lower α to 0.05 to keep attribute expansion computationally feasible. Running Llama-3.1-8b on our hardware (a single A100) takes less than 10 minutes for all selected pairs, while using GPT-4o via API calls takes under 2 minutes and costs less than $1.

Our fine-tuning process is highly efficient, using only 8 examples per category. For a dataset like CUB, which contains approximately 12k images across 200 classes, 30 iterations of fine-tuning and transduction take less than 10 minutes on a single A100 GPU.

Overall, the runtime cost of GTA-CLIP is approximately 10-20 minutes higher than that of CLIP on most datasets. However, these additional costs can be justified given the performance gains, since manually labeling even a small fraction of the dataset would require significantly more time, such as in fine-grained domains.

6. Limitations

There are two main limitations to our work. The first is the use of LLMs to generate fine-grained attributes. LLMs are known to hallucinate data, posing a risk of generating incorrect attributes. However, similar to AdaptCLIPZS [40], which demonstrated through human evaluation that the generated attributes are highly accurate, we did not find this to be a significant issue for the datasets we considered. Additionally, this step can be guided by domain experts.

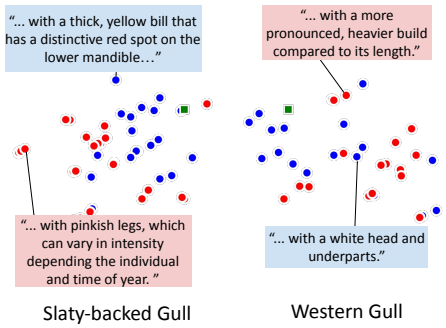


Figure 2. **t-SNE Plots of Class Attributes.** For each category the prototype, initial set of attributes, and the final set of attributes are shown in green, blue, and red respectively. Habitat, relative characteristics, and other distinguishing features are often identified through pairwise comparisons, while the initial attributes tend to describe the prominent visual features. These plots were obtained by mapping the CLIP text embeddings of the attributes using t-SNE. Please see the Appendix for detailed figures.

The second limitation is the applicability of the transductive setup. It requires access to the entire test set at once, which may not always be feasible (e.g., in a streaming setting). Additionally, we have not demonstrated robustness to imperfect knowledge of class distributions and data structure. For instance, we assume that images within a class cluster together, allowing us to model each class as a Gaussian distribution, which may not always be true.

7. Conclusion

The transductive setting offers a compelling approach for practitioners and domain experts who need precise answers for specific datasets. For instance, an ecologist might be interested in estimating species counts from data gathered via a network of camera traps, while a scientist might want to determine land-cover distribution using satellite imagery. VLMs enable straightforward labeling through language-based descriptions of categories, but their initial accuracy is often insufficient. Our work demonstrates that expanding categories based on attributes, when combined with transductive learning, enables model fine-tuning to achieve significant accuracy improvements. Additionally, this approach offers complementary advantages to traditional labeling methods, such as providing a few labeled examples per class. While we use large language models for convenience, this iterative procedure is naturally suited to human-in-the-loop approaches, allowing practitioners to incrementally add attributes and labels for ambiguous classes. These findings are practically valuable, as they offer end-users multiple pathways to improve labeling precision on their target dataset without investing significant efforts on training dataset curation and model training.

8. Acknowledgements

The research is supported in part by grant #2329927 from the National Science Foundation (USA). Our experiments were performed on the GPU cluster funded by the Mass. Technology Collaborative.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Rie Ando and Tong Zhang. Learning on graph with laplacian regularization. *Advances in neural information processing systems*, 19, 2006. 3

[3] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020. 2

[4] Liu Bo, Qiulei Dong, and Zhanyi Hu. Hardness sampling for self-training based transductive zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16499–16508, 2021. 2

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014. 5, 6, 2, 3

[6] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 6912–6920, 2021. 2

[7] Mia Chiquier, Utkarsh Mall, and Carl Vondrick. Evolving interpretable visual classifiers with large language models. *arXiv preprint arXiv:2404.09941*, 2024. 2

[8] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 5

[9] 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. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 5

[10] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-view zero-shot learning. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2332–2345, 2015. 2

[11] 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*, 132(2): 581–595, 2024. 2, 7

[12] Rui Gao, Xingsong Hou, Jie Qin, Jiaxin Chen, Li Liu, Fan Zhu, Zhao Zhang, and Ling Shao. Zero-vae-gan: Generating unseen features for generalized and transductive zero-shot learning. *IEEE Transactions on Image Processing*, 29:3665–3680, 2020. 2

[13] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004. 2

[14] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 5

[15] Mingyi Hong, Xiangfeng Wang, Meisam Razaviyayn, and Zhi-Quan Luo. Iteration complexity analysis of block coordinate descent methods. *Mathematical Programming*, 163: 85–114, 2017. 2

[16] Yannis Kalantidis, Giorgos Tolias, et al. Label propagation for zero-shot classification with vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23209–23218, 2024. 2

[17] 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. 5, 6, 2, 3

[18] 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. 2

[19] Bo Liu, Lihua Hu, Qiulei Dong, and Zhanyi Hu. An iterative co-training transductive framework for zero shot learning. *IEEE Transactions on Image Processing*, 30:6943–6956, 2021. 2

[20] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7086–7096, 2022. 1

[21] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *International Conference on Machine Learning*, pages 23033–23044. PMLR, 2023. 1

[22] Subhransu Maji. Discovering a lexicon of parts and attributes. In *Computer Vision–ECCV 2012. Workshops and Demonstrations: Florence, Italy, October 7-13, 2012, Proceedings, Part III 12*, pages 21–30. Springer, 2012. 4

[23] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5, 6, 2, 3

[24] 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. 2, 6, 7

[25] Ségolène Martin, Yunshi Huang, Fereshteh Shakeri, Jean-Christophe Pesquet, and Ismail Ben Ayed. Transductive

zero-shot and few-shot clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28816–28826, 2024. 2

[26] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *The Eleventh International Conference on Learning Representations*. 2

[27] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[28] Muhammad Jehanzeb Mirza, Leonid Karlinsky, Wei Lin, Horst Possegger, Mateusz Kozinski, Rogerio Feris, and Horst Bischof. Lafter: Label-free tuning of zero-shot classifier using language and unlabeled image collections. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[29] 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. 2

[30] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001. 3

[31] 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. 5, 6, 2, 3

[32] Devi Parikh and Kristen Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR 2011*, pages 1681–1688. IEEE, 2011. 4

[33] 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

[34] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2751–2758. IEEE, 2012. 4

[35] 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. 2

[36] 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, 5

[37] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18082–18091, 2022. 1

[38] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013. 2

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 5

[40] Oindrila Saha, Grant Van Horn, and Subhransu Maji. Improved zero-shot classification by adapting vlms with text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17542–17552, 2024. 2, 4, 6, 8

[41] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 3

[42] 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. 2

[43] K Soomro. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[44] 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. 2

[45] 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. 2

[46] Vladimir Vapnik. The support vector method of function estimation. In *Nonlinear modeling: Advanced black-box techniques*, pages 55–85. Springer, 1998. 2

[47] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 4, 5, 6, 2, 3

[48] Ziyu Wan, Dongdong Chen, Yan Li, Xingguang Yan, Junge Zhang, Yizhou Yu, and Jing Liao. Transductive zero-shot learning with visual structure constraint. *Advances in neural information processing systems*, 32, 2019. 2

[49] Wenlin Wang, Yunchen Pu, Vinay Verma, Kai Fan, Yizhe Zhang, Changyou Chen, Piyush Rai, and Lawrence Carin. Zero-shot learning via class-conditioned deep generative models. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 2

[50] Wenlin Wang, Hongteng Xu, Guoyin Wang, Wenqi Wang, and Lawrence Carin. Zero-shot recognition via optimal transport. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3471–3481, 2021. 2

[51] 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. 2

[52] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio 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. IEEE, 2010. 5

[53] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 2

[54] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *European Conference on Computer Vision*, pages 736–753. Springer, 2022. 1

[55] Xun Xu, Timothy Hospedales, and Shaogang Gong. Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision*, 123:309–333, 2017. 2

[56] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019. 2

[57] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3090–3100, 2023. 2

[58] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954, 2022. 3

[59] Yunlong Yu, Zhong Ji, Jichang Guo, and Yanwei Pang. Transductive zero-shot learning with adaptive structural embedding. *IEEE transactions on neural networks and learning systems*, 29(9):4116–4127, 2017. 2

[60] Yunlong Yu, Zhong Ji, Xi Li, Jichang Guo, Zhongfei Zhang, Haibin Ling, and Fei Wu. Transductive zero-shot learning with a self-training dictionary approach. *IEEE transactions on cybernetics*, 48(10):2908–2919, 2018. 2

[61] Maxime Zanella, Benoît Gérin, and Ismail Ayed. Boosting vision-language models with transduction. *Advances in Neural Information Processing Systems*, 37:62223–62256, 2025. 1, 2, 3, 4, 5

[62] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021. 2

[63] 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. 1, 2

[64] Xin Zhang, Shixiang Shane Gu, Yutaka Matsuo, and Yusuke Iwasawa. Domain prompt learning for efficiently adapting clip to unseen domains. *Transactions of the Japanese Society for Artificial Intelligence*, 38(6):B–MC2_1, 2023. 2

[65] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16793–16803, 2022. 1

[66] 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. 1, 2, 6, 7

[67] 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. 1, 2, 7