This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

LATTECLIP: Unsupervised CLIP Fine-Tuning via LMM-Synthetic Texts

Anh-Quan Cao[2*] Maximilian Jaritz[1] Matthieu Guillaumin[1] Raoul de Charette[2] Loris Bazzani[1]
[1]Amazon [2]Inria

Abstract

*Large-scale vision-language pre-trained (VLP) models (*e.g.*, CLIP [46]) are renowned for their versatility, as they can be applied to diverse applications in a zero-shot setup. However, when these models are used in specific domains, their performance often falls short due to domain gaps or the under-representation of these domains in the training data. While fine-tuning VLP models on custom datasets with human-annotated labels can address this issue, annotating even a small-scale dataset (*e.g.*, 100k samples) can be an expensive endeavor, often requiring expert annotators if the task is complex. To address these challenges, we propose* LATTECLIP*, an unsupervised method for fine-tuning CLIP models on classification with known class names in custom domains, without relying on human annotations. Our method leverages Large Multimodal Models (LMMs) to generate expressive textual descriptions for both individual images and groups of images. These provide additional contextual information to guide the fine-tuning process in the custom domains. Since LMM-generated descriptions are prone to hallucination or missing details, we introduce a novel strategy to distill only the useful information and stabilise the training. Specifically, we learn rich per-class prototype representations from noisy generated texts and dual pseudo-labels. Our experiments on 10 domain-specific datasets show that* LATTECLIP *outperforms pre-trained zero-shot methods by an average improvement of +4.74 points in top-1 accuracy and other state-of-the-art unsupervised methods by +3.45 points.*

1. Introduction

Large-scale vision-language pre-training [46] has emerged recently and demonstrated impressive generalization performance on various downstream tasks [7, 8, 23, 30, 44], especially in zero-shot classification [46, 47]. This success is attributed to its robust visio-linguistic representation, learned from a vast amount of large-scale web-scraped datasets [48]. However, these models often face challenges in specialized domains due to domain discrepancies and insufficient representation in the training data. Prior studies have demonstrated improvements on custom datasets through supervised fine-tuning [15, 60] or few-shot learning [50, 68]. Nevertheless, acquiring human-annotated labels is costly, even for relatively small datasets (*e.g.,* 100k samples), and often requires expert annotators for complex tasks. To address this, we propose LATTECLIP, which fine-tunes CLIP for classification on unlabeled training data to maximize performance on a test set from the same domain. Here, a *domain* refers to a set of shared characteristics within a dataset (e.g., cars, flowers, textures). Like in Unsupervised Domain Adaptation (UDA) [12, 19, 58], we consider the list of class names to be known a priori. An overview of LATTECLIP is shown in Fig. 1.

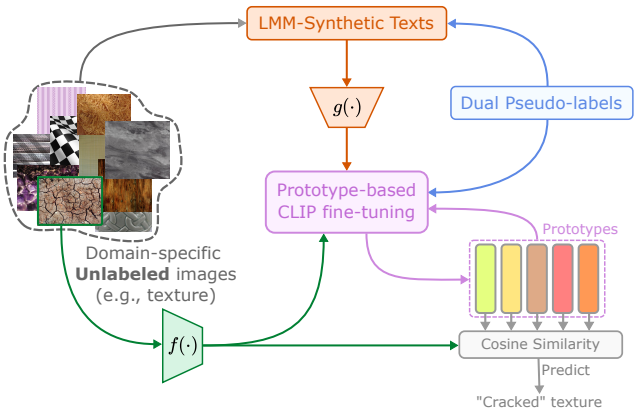


Figure 1. **Overview of LATTECLIP.** Our prototype-based method leverages different types of pseudo-labels and LMM-synthetic texts for improved unsupervised CLIP fine-tuning on domain-specific datasets (*e.g.,* texture). During inference, image features are compared with prototypes to generate predictions. Here, $f(\cdot)$ and $g(\cdot)$ are the CLIP image and text encoders, respectively.

Recent progress of Large Language Models (LLMs) [3, 24, 41, 55, 56] and Large Multimodal Models (LMMs) [4, 34] have led to a fundamental shift in training and fine-tuning methodologies. The research community is transitioning from a class-focused paradigm towards a more descriptive approach, where data is annotated with detailed textual de-

*The main work was done while interning at Amazon.

scriptions for training, and rich textual answers are provided at inference time. Consequently, an increasing number of methods [10, 16, 28] now leverage synthetically-generated text from LMMs as an additional source of supervision or contextual information to improve performance. Similar to these approaches, we harness the power of LMMs to generate descriptions for training, but with a strong emphasis on producing more expressive descriptions. Instead of only generating per-image descriptions, we also generate descriptions for groups of images, capturing their common characteristics, as well as class-level descriptions for all images within a category. These descriptions provide better contextual information, offering richer supervision for training, leading to improved classification accuracy in specific domains compared to the limited information from pseudo-labels and label propagation [20, 29].

However, directly fine-tuning CLIP with LMM-generated texts leads to poor performance due to CLIP overfitting to hallucinations and noise present in these texts. To address this, we propose a fine-tuning framework based on prototype learning [1, 39], where classes are represented as a set of prototypes, typically as feature vectors. Prototypes provide better control and interpretability of class representations through direct manipulation in the embedding space, helping regulate the influence of each synthetic description during training. To further improve the per-class prototype representations, we combine the synthetic texts with two types of pseudo-labels derived from both zero-shot and fine-tuning models. The zero-shot model offers better generalization thanks to pre-training knowledge, while the fine-tuning model provides stronger in-domain performance. During inference, these prototypes are compared with image features for classification. As LMMs are only employed during during fine-tuning, the inference time remains consistent with standard CLIP methods. We validate the effectiveness of our method across 10 domain-specific datasets. Compared to pre-trained CLIP models, LATTECLIP achieves an average improvement of +4.74 points in top-1 accuracy, surpassing other unsupervised fine-tuning baselines by +3.45 points.

Our contributions can be summarized as follows:

- We propose LATTECLIP, a novel method that synthesizes multiple types of image descriptions to enhance the unsupervised fine-tuning of CLIP models on domain-specific datasets, leveraging the language expressiveness of LMMs.
- To make training robust to noisy texts and pseudo-labels, we employ a prototype framework with a momentum update, enabling us to control the influence of synthetic text features. To further refine the useful image descriptions, we introduce a Dynamic Feature Mixer module that assigns higher weight to important text, resulting in better-combined text embeddings.
- We show that mixing pseudo-labels from zero-shot model and fine-tuning model significantly improves performance; the former preserves pre-trained knowledge, while the latter improves the accuracy on the target distribution. Experiments show that LATTECLIP significantly outperforms all baselines on average across 10 domain-specific datasets.

2. Related works

Adapting CLIP for Classification. CLIP-based methods [46, 62, 63, 65] exhibit competitive zero-shot classification performance. For further improvement on downstream classification datasets, CLIP can be adapted to close the gap between pre-trained representations and specific domains.

In few-shot learning, one has access to a small number of labels, typically between 1-16 samples per class, and many works have adapted it to CLIP [9, 13, 14, 38, 45, 50, 64, 66, 68, 69]. Prototypical learning [51] is a seminal work in few-shot learning and builds an average embedding (prototype) for each class. During inference, one then matches the test sample to the nearest prototype. This concept recently re-emerged for adapting CLIP, by building a cache model holding the knowledge from the few-shot training set [66,69]. Different from that, we continuously update the prototypes with momentum during the training process with multimodal features from *unsupervised* texts and images. Other works leverage prompt learning [68] or efficient fine-tuning [9, 64].

Supervised fine-tuning methods require a significant amount of labeled examples for training [15, 26, 46, 59, 60]. Linear probing [46, 60] is a simple technique that trains a classifier on top of frozen image features, but can lead to worse results due to overfitting. This problem has been tackled by using two-step training schedules of linear probing and full fine-tuning [26], masked image modeling [59] and by fine-tuning with contrastive loss by aligning the image with a template text including the class label (FLYP [15]). In contrast to FLYP, we add LMM-generated descriptions to the contrastive loss, and stabilize unsupervised training by learning prototypes with momentum.

Different from few-shot and fine-tuning, we focus on the challenging scenario of unsupervised fine-tuning, where no labels are available, because they are too costly to annotate.

Unsupervised Model Adaptation Our Unsupervised fine-tuning task is related to Unsupervised Domain Adaptation, where one typically reduces the discrepancy between the source and target data. However, lately, the task of source-free domain adaptation (SFDA) has emerged, where target adaptation is performed without access to the source data, see survey paper [31]. Many methods exploit that the source model can partially generalize to the target domain, and fine-tune with pseudo-labels [33], adversarial learning [32], historical contrastive learning [21] or mixup [27]. While [21] perform momentum contrastive learning [17] on different

image augmentations, we contrast image-text pairs. The above SFDA works train on a narrow source distribution. Instead, ReCLIP [20] leverages CLIP, which is pre-trained on wide-distribution large-scale data. ReCLIP leverages pseudo labels, cross entropy between separate modalities and focus on transductive setup (train/test on test set). Test-time adaptation methods [35, 49, 53, 57, 67] update the model to align with the target distribution at test time using a single image in self-supervised manner, requiring optimization at inference. Unlike these approaches, we leverage LMM-generated texts to maximize test performance by fine-tuning model parameters on unlabeled training data, thus keeping the model parameters fixed during testing.

LMMs for Synthetic Labels. Using synthetically-generated labels and textual descriptions is becoming a standard in the field, because of the general availability of LLMs and LMMs that can be prompted and guided with task-specific examples [2, 10, 16, 28, 34, 42, 54]. This provides an opportunity for VLP, that typically uses large-scale image-text pair data scraped from the internet, *e.g.*, LAION-5B [48]. Instead of using noisy and inconsistent captions or annotating a large set, we synthetically-generate descriptions. While LaCLIP [10] rewrites existing captions with LLMs (text-only input), VeCLIP [28] prompts an LMM to caption the image, followed by LLM text processing. SynthCLIP [16] synthesizes first the text, and then the images with text-to-image generative models. In our work, we leverage LMMs to caption images, focusing on fine-tuning rather than pre-training, combining the pre-trained model's knowledge with new synthetic captions in a balanced approach.

3. Method

Fine-tuning CLIP with combination of predefined templates, such as `"a photo of a [class]."`, was shown to yield effective results when using ground-truth class labels [15, 26, 60]. However, in the absence of ground-truth class labels, fine-tuning CLIP models with pseudo-labels [29], using FLYP [15], leads to limited improvements[1] This can be caused by two factors. First, the text employed as supervision, resulting from the combination of the template and pseudo-label, lacks expressivity and discriminativity. This is typically the case for classes that are not visually descriptive, such as types of land use (*e.g.*, annual crop, industrial, *etc.*) or names of textures (*e.g.*, paisley, sprinkled, *etc.*). Second, pseudo-labels are inherently noisy, which negatively affects the downstream classification performance due to domain shifts relative to the original training data.

Our method, LATTECLIP, addresses these limitations

[1]Tab. 1 reports the performance of "FLYP + pseudo labels" where we show limited improvements with respect to CLIP across 10 datasets on average. We also observe performance drops on some datasets (*e.g.*, Food101, Flower102).

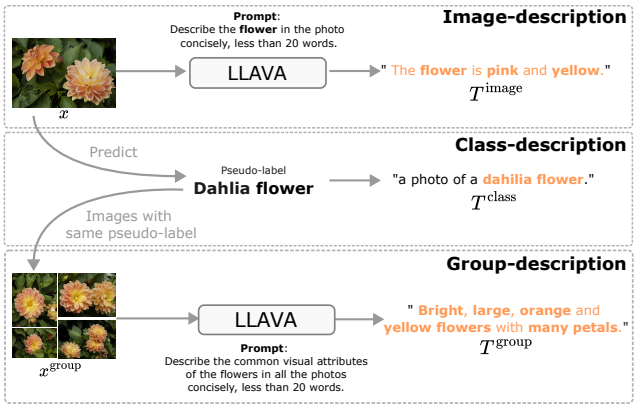


Figure 2. **Text Generation with LMM.** In addition to the usual *class-description* (middle), combining template text and pseudo-label, we leverage LMM [34] to generate *image-description* (top) which provide more expressive visual description of the image. Further, by considering random group of images with the same pseudo-labels, we prompt [34] to capture shared characteristics as *group-description* (bottom).

by proposing an expressive unsupervised text generation (Sec. 3.1) and a prototype-based learning mechanism (Sec. 3.2) to mitigate noisy pseudo labels. To improve expressivity beyond pseudo-labeling, we build upon a recent LMM [34], generating descriptions at multiple levels of contextual granularity, describing the individual image, group of similar images, and entire class. Individual image descriptions offer detailed though possibly extraneous information, which is addressed by group descriptions that capture shared characteristics of similar images, albeit with some noise. This noise is mitigated by class descriptions, which provide stable representations to address inconsistencies. Equipped with such textual description, we additionally introduce a prototype-based learning framework that learns a set of class prototypes from the generated text features. These prototypes are updated in a momentum setting to produce a smooth optimization over the whole training set, reducing the effect of noise from outlier samples and incorrect synthesized texts.

3.1. Expressive Text Generation with LMMs

Without access to ground-truth labels for training CLIP models, we must rely on noisy pseudo-labels. Furthermore, class names alone often lack visual descriptiveness. Consequently, using only cross-entropy loss or solely relying on class names leads to suboptimal performance in our setting. To address this challenge, we introduce a novel approach that leverages generated text to provide additional contextual information. In addition to the more standard *class-description* mentioned above, we propose two additional ways using a recent LMM [34] to generate textual descriptions of im-

ages: *image-description* and *group-description*, as depicted in Fig. 2. These generated texts hold complementary information with increasing semantic abstraction, from class[2], to single image, to group of images, all of which help the model to learn more precise classification boundaries. The *image-description* texts provide detailed descriptions of individual images, capturing their unique characteristics and subtle features. The *group-description* texts offer a comprehensive description representing the entire class, covering shared features and common attributes.

Importantly, we found that the above mentioned descriptions are complementary to the use of template text with pseudo-label class, which we refer as *class-description*. In fact, we later show that preserving this *class-description* in the training process is crucial as our generated texts can be noisy due to missing details or hallucination. The combination of class-/image-/group-description provides a stable and reliable representation corresponding to the classes.

More formally, for each image x we generate three texts, illustrated in Fig. 2, and defined as follows:

Class-description (T^{class}) provides a consistent class representation using template `"a photo of a [class]."` where `[class]` is substituted with the image pseudo-label c obtained from a CLIP zero-shot.

Image-description (T^{image}) captures unique features of image x. We generate T^{image} by prompting LLAVA [34] with: `"Describe the [domain] in the photo concisely, using less than 20 words."` where `[domain]` is replaced with the dataset domain (*e.g.*, flower, product, pet, car, *etc.*). We show *image-description* examples in Fig. 5.

Group-description (T^{group}) captures shared visual characteristics between similar images, to combat known limitation of LMM which may miss or hallucinate visual characteristics [36]. To generate T^{group} from image x, we randomly sample multiple images with the same pseudo-label as x. These are collaged into a single image x^{group} fed to LLAVA which is prompted with: `"Describe the common visual attributes of the [domain] in all the photos concisely, in fewer than 20 words."`. Examples of such group-descriptions are illustrated in Fig. 5.

To generate these descriptions, we use LLAVA 1.6 [34] with a 4-bit quantized Mistral 7B model. This model requires approximately 5GB of GPU memory and takes around 1.2 seconds to generate a single description per image on a Tesla V100 GPU. This makes description generation relatively cost-effective, as we can run five instances of this model in parallel on a Tesla V100 32GB GPU, taking approximately 3.4 hours to generate descriptions for 50k images.

[2]Here, "class" refers to the class-description.

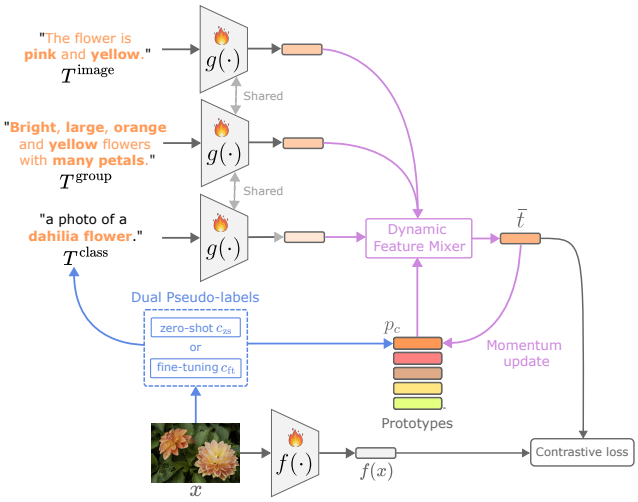


Figure 3. **Training.** For image x, we predict pseudo-label $c \in \{c_{\text{zs}}, c_{\text{ft}}\}$ and create three type of descriptions per pseudo-label as described in Sec. 3.1. Our Dynamic Feature Mixer combines these descriptions with the corresponding prototype p_c to produce a prototype-text embedding $\bar{t}$, which updates the prototype p_c. Lastly, the contrastive loss Eq. (3) is computed between $\bar{t}$ and and the image embedding $f(x)$.

3.2. Prototype-based CLIP fine-tuning

Adopting directly the generated texts from Sec. 3.1 is ineffective, because the text encoder overfits to the distribution of generated texts, which are noisy by construction due to hallucinations of the LMM and missing details. We confirm this experimentally in Tab. 2, rows 4, 5, 6. Therefore, we propose a prototype learning approach that is capable of determining the important synthetic texts and learning better class representations from them. Our approach mixes three key ingredients as shown in Fig. 3: (1) a simple strategy to preserve robustness by leveraging pseudo-labels from both frozen *and* fine-tuning CLIP models; (2) a feature mixer that dynamically balances the importance of each text T^{class}, T^{image} and T^{group}; (3) a module that updates the prototypes during training, stabilizing the learning process.

Dual Pseudo-labels. As in WISE-FT [60], we observe that training only with pseudo-labels from the fine-tuning model improves accuracy but at the cost of overfitting to the training distribution. Hence, to preserve robustness, for each image we employ two pseudo-labels $\{c_{\text{zs}}, c_{\text{ft}}\}$ originating from *both* the zero-shot model (c_{zs}) and the fine-tuning model (c_{ft}). We later show that this simple strategy offers greater generalization and accuracy.

Prototype Learning. From the generated texts and pseudo-labels, we aim to learn a set of prototypes corresponding to all classes, denoted as $\{p_c\}_{c=1}^{C}$. These prototypes are designed to capture class-specific details of the syn-

thesized texts and pseudo-labels within the CLIP embedding space. First, the prototypes are initialized with features derived from the T^{class}, generated based on its associated class name. Then, for an image x, we use our dual pseudo-labels from zero-shot and fine-tuning $\{c_{\text{zs}}, c_{\text{ft}}\}$ to generate two *class-description* texts $\{T^{\text{class}}_{\text{zs}}, T^{\text{class}}_{\text{ft}}\}$ and select the corresponding prototypes p_{zs} and p_{ft}. Our feature mixer strategy, detailed below, then combines the two *class-descriptions* with the *image-description* and the *group-description*, therefore obtaining two prototype-text embeddings $\bar{t}_{\text{zs}}$ and $\bar{t}_{\text{ft}}$, see Fig. 3. We then apply a momentum update to the corresponding prototypes. Finally, we apply two contrastive losses [15] between the image embedding $f(x)$ and each of the prototype-text embeddings $\bar{t}_{\text{zs}}$ and $\bar{t}_{\text{ft}}$.

Dynamic Feature Mixer. To compensate for noisy text descriptions, we propose a mechanism that dynamically reweights the three descriptions as a function of the cosine similarity between each description embedding and corresponding prototype, see Fig. 4. Intuitively, our goal is to assign higher weights to descriptions uniquely describing a class and lower weights to generic descriptions. In the general case, for a text T we first compute the cosine similarities between its CLIP embedding $g(T)$ and each of the prototypes, and obtain its weight w from the difference between the two closest similarities. This writes:

$$w = \text{top}_1\left(\frac{g(T)\cdot p_c}{\|g(T)\|\|p_c\|}\right)_{c=1}^{C} - \text{top}_2\left(\frac{g(T)\cdot p_c}{\|g(T)\|\|p_c\|}\right)_{c=1}^{C} \tag{1}$$

where $\text{top}_1(\cdot)$ and $\text{top}_2(\cdot)$ return the largest and second largest values of the input set, respectively. A high weight indicates a large gap between $\text{top}_1(\cdot)$ and $\text{top}_2(\cdot)$ similarities, ensuring the text feature is uniquely similar to a single prototype while dissimilar from the rest, as $\text{top}_2(\cdot)$ value serving as an upper bound for the similarity of the remaining prototypes. Alternatively, we could use the mean or median, but this might result in a text being very similar to a few prototypes while remaining dissimilar to others. Subsequently, given the set of texts $\{T^{\text{image}}, T^{\text{group}}, T^{\text{class}}\}$ and the weights $\{w^{\text{image}}, w^{\text{group}}, w^{\text{class}}\}$ computed using Eq. (1). The resulting prototype embedding $\bar{t}$ is defined as

$$\bar{t} = (1-\alpha)\frac{\sum_{i\in I} w^i \cdot g(T^i)}{\sum_{i\in I} w^i} + \alpha p_c \tag{2}$$

where $I = \{\text{image}, \text{group}, \text{class}\}$ and α is the prototype weight. We empirically set α to 0.99 in all experiments to stabilize training, as the prototypes are more reliable than the synthetic text embeddings. Thus, this act a strong regularization mechanism against the noise induced by the synthetic texts. Yet, $\bar{t}$ remains tailored for each image as T^{image} and T^{group} differ. With two pseudo-labels per image, this results in two prototype-text embeddings $\{\bar{t}_{\text{zs}}, \bar{t}_{\text{ft}}\}$.

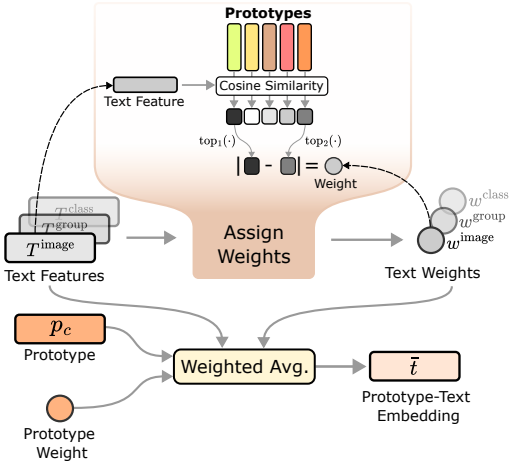


Figure 4. **Dynamic Feature Mixer.** We compute cosine similarities between each text feature and all prototypes. Weights are determined by the difference between the top two similarity scores. We calculate a weighted average of the features and combine it with the prototype (Sec. 3.2), creating a representation relevant to the input prototype yet distinct from others.

Training. Given an image x, we train both the image encoder $f(\cdot)$ and text encoder $g(\cdot)$ using contrastive loss to align the image embedding $f(x)$ with both the prototype-text embeddings $\bar{t}_{\text{zs}}$ and $\bar{t}_{\text{ft}}$, resulting in two losses $\mathcal{L}_{\text{zs}} = \mathcal{L}_{\text{con}}(x, \bar{t}_{\text{zs}})$ and $\mathcal{L}_{\text{ft}} = \mathcal{L}_{\text{con}}(x, \bar{t}_{\text{ft}})$ respectively with $\mathcal{L}_{\text{con}}(\cdot,\cdot)$ defined as:

$$\begin{aligned}\mathcal{L}_{\text{con}}(x, \bar{t}) = &-\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(f(x)\cdot\bar{t}_i/\tau)}{\sum_{j=1}^{N}\exp(f(x)\cdot\bar{t}_j/\tau)} \\ &-\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\bar{t}_i\cdot f(x)/\tau)}{\sum_{j=1}^{N}\exp(\bar{t}_j\cdot f(x_j)/\tau)},\end{aligned} \tag{3}$$

where N is the batch size and τ is the temperature parameter as in [46]. The first term of Eq. (3) normalizes over text embeddings to match the correct text to an image, while the second normalizes over image embeddings to match the correct image to a text. The final loss is $\mathcal{L}_{\text{zs}} + \mathcal{L}_{\text{ft}}$.

Momentum update prototypes. For a pseudo-label c, we derive the corresponding prototype-text embedding $\bar{t}$ for each image. During training, the average prototype-text embedding $\bar{t}_{\text{batch}}$ is computed over the images in the batch. Using the pseudo-label c, we update the respective prototype p_c with a momentum μ, obtaining the updated embedding $\bar{p}_c = (1-\mu)\bar{t}_{\text{batch}} + \mu p_c$, which is then stored back in the prototype bank as the prototype for class c. Momentum update works effectively when $\mu \in \{0.99, 0.999, 0.9999\}$ [17]. As we fine-tune on smaller dataset with fewer iterations, we set μ to 0.99 for faster updates of the prototypes. Intuitively, the prototype can be viewed as the running average

Method	Average	EuroSAT	Sun397	Food101	Flower102	DTD	FGVC	Oxford Pets	Cars	UCF101	Caltech101
Oracle	81.76	94.46	77.45	85.01	87.90	76.65	37.95	92.42	90.21	79.49	96.02
LLAVA zero-shot	27.23	44.78	15.74	29.81	6.58	20.27	3.18	28.92	3.38	44.25	75.38
Pre-trained CLIP	67.49	42.95	68.20	78.65	71.30	55.32	23.79	87.30	**88.25**	64.37	94.73
ReCLIP [20]	68.78	49.25	69.07	77.91	71.13	56.91	**25.92**	88.50	87.84	68.86	92.37
FLYP [15] + pseudo-label [29]	70.01	67.12	70.19	76.83	68.78	**61.82**	17.40	88.96	84.19	69.44	94.69
LATTECLIP (ours)	**72.23**	**80.27**	**70.68**	**79.63**	**71.94**	56.26	22.02	**89.21**	87.40	**70.08**	**94.77**

Table 1. Top-1 accuracy on 10 classification datasets. We report the results for five baselines and our method. The 'Average' column shows the average results across all datasets. **Best**/2nd best.

of the text-prototype embeddings assigned to the class. This process is repeat for each pseudo-label in $\{c_{\text{zs}}, c_{\text{ft}}\}$.

Inference. The predictions for an image x are made by comparing the image embedding $f(x)$, where $f(\cdot)$ is the fine-tuned CLIP image encoder, with $\{p_c\}_{c=1}^{C}$ and taking the prototype with the highest cosine similarity as output.

4. Experiments

We evaluate LATTECLIP on the task of fine-tuning on 10 specialized classification datasets, without using any ground truth labels. We use the training set for unsupervised training and use the test set to compute the top-1 accuracy.

Datasets. We employ a mixture of datasets covering various specialized domains, including satellite imagery, food dishes, airplane models, and others: EuroSAT [18], SUN397 [61], Food101 [5], DTD [6], FGVC [37], Oxford Pets [43], Cars [25], UCF101 [52], Caltech101 [11], Flower102 [40]. These datasets feature specific classes, such as the car model, making the unsupervised fine-tuning setup challenging. We follow the standard train/val/test splits in [68]. We train LATTECLIP using the combined train and val sets and report its performance on the test set.

Baselines. We compare our method to four unsupervised baselines and one fully supervised baseline, which serves as an oracle. First, we perform zero-shot classification with a pre-trained CLIP model. As in CLIP [46], we compute text embeddings for all classes with template `"a photo of a [class]."`. For classification, we compute the cosine similarity between each image and all class text embeddings. Our second baseline, ReCLIP [20], also performs fine-tuning without labels but utilizes improved pseudo labels and self-training. However, ReCLIP primarily focuses on experiments conducted in a *transductive* manner, which involves training and evaluating on the test split of each dataset. To ensure a fair comparison, we retrained ReCLIP using the same CLIP-based model and identical dataset splits as our method. Third, we combined FLYP [15] with pseudo-labeling [29] for unsupervised fine-tuning, as the original method relies on supervised fine-tuning. Note that we use FLYP without weight ensembling to maintain a fair comparison with ReCLIP, which also does not employ weight ensembling. Finally, we add "LLAVA zero-shot" baseline which prompts LLAVA to classify the image from a given list of classes, using the following prompt `"Select the most appropriate category for the image from the following options:[options]. Write only the category name."` , where options is replaced with the list of class names. For the supervised baseline, we train FLYP using ground-truth labels, serving as an oracle. The evaluation is performed in a zero-shot fashion, like zero-shot CLIP. Since our method uses prototypes, no class template embeddings have to be computed, and we directly use the prototype vectors. For all baselines and ours, we use OpenCLIP [22], the open-source implementation of CLIP [46], with a ViT/B-32 architecture, pre-trained on the LAION-2B dataset. Performance is reported based on the last epoch since we have no supervision signal. For additional implementation details, please refer to the supplementary material.

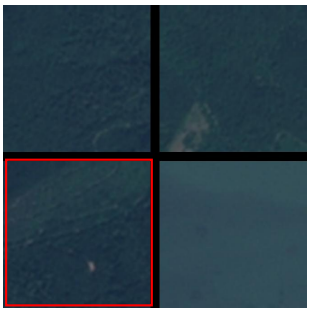
4.1. Results

The main results with top-1 accuracy on the 10 datasets are shown in Tab. 1. Across all datasets, LATTECLIP improves the average top-1 accuracy of CLIP by 4.74 points. Furthermore, it outperforms all unsupervised baselines, including the recently published ReCLIP [20] and our proposed baseline that integrates FLYP [15] with pseudo-labeling [29], by 3.45 and 2.22 points, respectively. Interestingly, FLYP + pseudo-label outperforms ReCLIP, likely due to the robustness and effectiveness of fine-tuning both image and text encoders with contrastive loss, instead of just the image encoder with cross-entropy loss, as demonstrated in FLYP [15]. Notably, LLAVA zero-shot has low overall performance, which could be attributed to LLAVA being trained in generative autogressive manner, thus not optimal for discriminative tasks. Lastly, the oracle is shown on the first line by training FLYP with ground-truth labels. The 9.53-point average performance gap between the fully supervised oracle and unsupervised LATTECLIP highlights room for improvement. Still, LATTECLIP performs competitively, narrowing the gap across multiple datasets, particularly on Oxford Pets, Cars, and Caltech101, to less than 3%.

	T^{class}	T^{image}	T^{group}	Average	EuroSAT	Sun397	Food101	Flower102	DTD	FGVC	Oxford Pets	Cars	UCF101	Caltech101
1	✓	✓	✓	**72.23**	**80.27**	**70.68**	79.63	71.94	56.26	**22.02**	89.21	87.40	**70.08**	94.77
2	✓	✓		70.74	79.98	64.85	75.52	**72.31**	57.03	16.44	89.45	87.09	69.50	**95.21**
3	✓			70.67	78.22	59.79	**81.52**	71.21	56.74	16.50	**90.00**	**87.89**	69.84	94.97
4		✓	✓	55.97	64.75	63.28	76.73	50.95	48.76	9.00	64.51	32.98	60.45	88.24
5		✓		52.37	44.31	62.54	77.00	48.44	43.09	7.89	56.75	33.86	58.58	91.24
6			✓	53.52	59.35	65.00	77.06	31.67	49.05	9.57	64.49	25.99	66.90	86.09
7	✓		✓	71.63	79.68	70.07	80.16	71.99	**57.47**	18.00	89.81	86.15	68.84	94.12

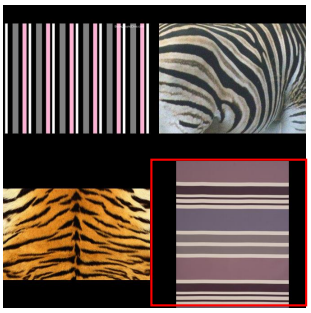
Table 2. **Impact of generated texts.** Best performance is achieved when using all types of descriptions. **Best**/2nd best.

Class: Forest (Eurosat)



T^{group}: Dark blue, green, and brown colors, indicating water, vegetation, and land.
T^{image}: The image shows a large body of water with no visible land use.

Class: Banded (DTD)

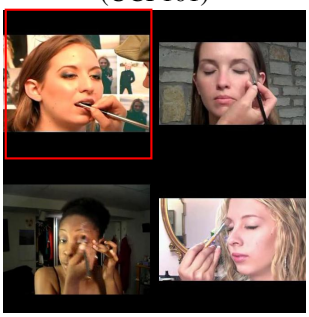


T^{group}: Stripes, zebra, animal print, geometric shapes, lines, and bold colors.
T^{image}: Striped pattern.

Class: Indoor Factory (SUN397)

T^{group}: Industrial factory with white walls, industrial equipment, machinery, clocks, and doors.
T^{image}: Large industrial building with doors.

Class: Apply Eye Makeup (UCF101)



T^{group}: Makeup application, close-up, hands holding tools.
T^{image}: Girl applying makeup.

Figure 5. **Examples of generated captions.** We either generate a caption from the group of 4 images, by inputting them as tiled single image into LLaVA (T^{group}), or we input a single image to LLaVA (T^{image}). For simplicity, in this figure, we only show a single image caption (highlighted by red bounding box).

In Fig. 5, we show examples of generated *image-description* T^{image} and *group-description* T^{group}. Overall, T^{group} offers more comprehensive and contextual information. For instance, in the top-right example, T^{image} is simply "striped pattern," whereas T^{group} provides richer details, including "zebra, animal print, geometric shape, lines". This trend is evident in other examples as well. For example, in the bottom-right example, T^{image} is "Girl applying makeup," while T^{group} elaborates with "Makeup application, close-up, hands holding tools." Additionally, image-description fails to capture "forest" in the top-left example, describing it merely as "The image shows a large body of water with no visible land use." In contrast, T^{group} includes relevant details such as "vegetation, land, green and brown colors."

4.2. Ablations

Different types of synthetic descriptions. Tab. 2 illustrates the impact of different generated texts on overall performance. We observe that all texts are essential for achieving the best performance. Specifically, excluding the *image-description* reduces the average performance across all datasets by 0.6 (row 1 vs. row 7). The impact of removing the *group-description* is even more significant with a 1.49 points reduction (row 1 vs. row 2). Additionally, omitting both the *image-description* and *group-description* results in an even larger loss of 1.56 points (comparing row 1 to row 3). Rows 4, 5, and 6 show that relying solely on synthetic texts causes a drop in performance due to the noise and inaccuracies introduced by the generated descriptions.

Dynamic Feature Mixer. We ablate our Dynamic Feature Mixer in Tab. 3 (row "w/o Dynamic Feature Mixer") by setting all the text weights to 1.0, so that all texts contribute equally. The average performance drops by 2 points, with significant decreases on multiple datasets, such as -14.23 on EuroSAT, -1.41 on DTD, and -2.03 on Cars. This demonstrates that our Dynamic Feature Mixer module effectively assigns relevant weights to the meaningful descriptions.

Dual Pseudo-Labels. Best performance is achieved using both zero-shot and fine-tuning pseudo-labels $\{c_{\text{zs}}, c_{\text{ft}}\}$. This is assessed in Tab. 3 by removing the corresponding losses. Removing the zero-shot pseudo-label (row "w/o $\mathcal{L}_{\text{zs}}$") leads

Method	Average	EuroSAT	Sun397	Food101	Flower102	DTD	FGVC	Oxford Pets	Cars	UCF101	Caltech101
LATTECLIP (ours)	**72.23**	**80.27**	**70.68**	79.63	71.94	56.26	22.02	**89.21**	**87.40**	70.08	**94.77**
w/o Dynamic Feature Mixer	70.23	66.04	69.41	**80.18**	**72.51**	54.85	23.43	87.14	85.37	69.02	94.32
w/o $\mathcal{L}_{\text{ft}}$	68.26	47.06	69.80	79.23	70.48	56.97	**23.82**	87.65	87.19	65.77	94.60
w/o $\mathcal{L}_{\text{zs}}$	70.58	76.96	68.18	70.29	71.01	**61.05**	19.89	87.49	86.13	**70.39**	94.36
w/o Momentum Update	45.72	31.19	56.17	68.08	57.41	31.32	13.41	13.03	43.31	54.69	88.56

Table 3. **Method ablation.** All components contribute to the best performance. **Best**/2nd best.

to a significant drop across multiple datasets: -3.31 on EuroSAT, -2.5 on SUN397, -9.34 on Food101, and an average decline of -1.65 across all datasets. Furthermore, removing the fine-tuned pseudo-labels (row "w/o $\mathcal{L}_{\text{ft}}$") results in an even more substantial average performance drop of -3.97, with particularly notable decreases of -33.21 on EuroSAT and -4.31 on UCF101. We conjecture that this is because the zero-shot pseudo-label is more robust, while the fine-tuned pseudo-label has higher accuracy on the training dataset.

Momentum Update. We ablate the impact of the momentum update by setting μ=0, as shown in row "w/o Momentum Update" in Tab. 3, therefore directly replacing the prototype by the new weighted text features. Without momentum update, performance declines dramatically, with an average decrease of -26.51 across all datasets. Significant declines are observed in many datasets, such as -44.09 on Cars, -14.51 on SUN397, and -14.53 on Flower102. This substantial drop is attributed to the high variance in the prototypes due to the noisy generated texts.

Incorrect images in generating T^{group}. We analyze how incorrect images within a group affect the generated group-description T^{group}. We test groups of 4 images with different number of correct images, selected using ground-truth labels. For 1, 2, 3 and 4 correct images, this results in top-1 accuracy of 72.48, 72.61, 72.72, and 72.64, respectively, averaged across all datasets. Performance improves slightly with more correct images. Notably, our method using pseudo-labels achieves a performance of 72.23, which is competitive with the ground-truth label selection. This demonstrates tolerance to noise and pseudo-label inaccuracies within the image group. Detailed performance on all datasets is provided in the supplementary material.

Number of images per group. We analyze the impact of increasing the number of images per group on performance by testing groups with 2, 4, 8, and 16 images, resulting in average performance scores of 71.55, 72.23, 72.31, and 72.49, respectively, across all datasets. The performance improves with more images per group, likely because the probability of including the correct images increases. The most notable improvement occurs when increasing the number of images per group from 2 to 4, with score rising from 71.55 to 72.23. This is because achieving a majority with only 2 images requires 100% accuracy, whereas larger groups can tolerate

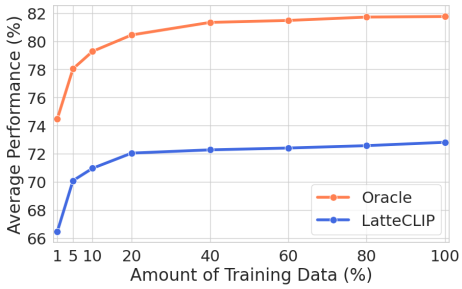


Figure 6. **Impact of amount of training data.** Average top-1 accuracy on 10 datasets while varying the amount of training data.

some errors while still maintaining a correct majority. Performance plateaus after 4 images, possibly due to the fixed input resolution of LLAVA [34], leading to lower per-image resolution as the number of images increases. We provide the performance for all datasets in the supplementary material.

Impact of amount of training data. Sec. 4.2 illustrates the effect of training data size on the average top-1 accuracy of LATTECLIP and oracle across 10 datasets. Despite being unsupervised, LATTECLIP exhibits strong robustness to varying amounts of training data, comparable to an oracle. Specifically, LATTECLIP's performance drops only 0.77/6.36 on 20%/1% of the data, respectively, compared to 7.28/1.30 of the oracle. Increasing data size generally improves performance, though the gains diminish notably for both after 20%.

5. Conclusion

LATTECLIP is a novel method for unsupervised CLIP fine-tuning on specialized datasets where human annotations are costly or require expert knowledge. Leveraging LMMs, LATTECLIP generates rich and expressive synthetic textual descriptions at various levels of contextual granularity, including *image-description*, *group-description*, and *class-description*. To effectively learn from these potentially noisy descriptions, we propose a prototype learning framework with three key elements: (1) dual pseudo-labels from frozen and fine-tuning CLIP models; (2) a Dynamic Feature Mixer for optimal text feature weighting; and (3) momentum update to enhance training stability. LATTECLIP surpasses comparable baselines on average across all datasets.

References

[1] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 1994. 2

[2] 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*, 2023. 3

[3] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024. 1

[4] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. 1

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014. 6

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 6

[7] Mohammad Fahes, Tuan-Hung Vu, Andrei Bursuc, Patrick Pérez, and Raoul de Charette. Poda: Prompt-driven zero-shot domain adaptation. In *ICCV*, 2023. 1

[8] Mohammad Fahes, Tuan-Hung Vu, Andrei Bursuc, Patrick Pérez, and Raoul de Charette. A simple recipe for language-guided domain generalized segmentation. In *CVPR*, 2024. 1

[9] Mohammad Fahes, Tuan-Hung Vu, Andrei Bursuc, Patrick Pérez, and Raoul de Charette. Fine-tuning clip's last visual projector: A few-shot cornucopia. *arXiv*, 2024. 2

[10] Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. In *NeurIPS*, 2023. 2, 3

[11] 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. *CVPRW*, 2004. 6

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 1

[13] 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. *IJCV*, 2024. 2

[14] Muhammad Waleed Gondal, Jochen Gast, Inigo Alonso Ruiz, Richard Droste, Tommaso Macri, Suren Kumar, and Luitpold Staudigl. Domain aligned clip for few-shot classification. In *WACV*, 2024. 2

[15] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *CVPR*, 2023. 1, 2, 3, 5, 6

[16] Hasan Abed Al Kader Hammoud, Hani Itani, Fabio Pizzati, Philip Torr, Adel Bibi, and Bernard Ghanem. Synthclip: Are we ready for a fully synthetic clip training? *arXiv*, 2024. 2, 3

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2, 5

[18] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *JSTAR*, 2019. 6

[19] Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. Mic: Masked image consistency for context-enhanced domain adaptation. In *CVPR*, 2023. 1

[20] Xuefeng Hu, Ke Zhang, Lu Xia, Albert Chen, Jiajia Luo, Yuyin Sun, Ken Wang, Nan Qiao, Xiao Zeng, Min Sun, et al. Reclip: Refine contrastive language image pre-training with source free domain adaptation. In *WACV*, 2024. 2, 3, 6

[21] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In *NeurIPS*, 2021. 2

[22] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, jul 2021. 6

[23] 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 *ICML*, 2021. 1

[24] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. 1

[25] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013. 6

[26] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *ICML*, 2022. 2, 3

[27] Jogendra Nath Kundu, Akshay R Kulkarni, Suvaansh Bhambri, Deepesh Mehta, Shreyas Anand Kulkarni, Varun Jampani, and Venkatesh Babu Radhakrishnan. Balancing discriminability and transferability for source-free domain adaptation. In *ICML*, 2022. 2

[28] Zhengfeng Lai, Haotian Zhang, Bowen Zhang, Wentao Wu, Haoping Bai, Aleksei Timofeev, Xianzhi Du, Zhe Gan, Jiulong Shan, Chen-Nee Chuah, Yinfei Yang, and Meng Cao.

Veclip: Improving clip training via visual-enriched captions. In *ECCV*, 2024. 2, 3

[29] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICMLW*, 2013. 2, 3, 6

[30] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021. 1

[31] Jingjing Li, Zhiqi Yu, Zhekai Du, Lei Zhu, and Heng Tao Shen. A comprehensive survey on source-free domain adaptation. *TPAMI*, 2024. 2

[32] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, 2020. 2

[33] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020. 2

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 3, 4, 8

[35] Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? In *NeurIPS*, 2021. 3

[36] Yanqing Liu, Kai Wang, Wenqi Shao, Ping Luo, Yu Qiao, Mike Zheng Shou, Kaipeng Zhang, and Yang You. Mllms-augmented visual-language representation learning. *arXiv*, 2023. 4

[37] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv*, 2013. 6

[38] Ségolène Martin, Yunshi Huang, Fereshteh Shakeri, Jean-Christophe Pesquet, and Ismail Ben Ayed. Transductive zero-shot and few-shot clip. In *CVPR*, 2024. 2

[39] Allen Newell, Herbert Alexander Simon, et al. *Human problem solving*, volume 104. Prentice-hall Englewood Cliffs, NJ, 1972. 2

[40] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian conference on computer vision, graphics & image processing*, 2008. 6

[41] OpenAI. Gpt-4 technical report, 2024. 1

[42] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. 3

[43] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012. 6

[44] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 1

[45] Qi Qian, Yuanhong Xu, and Juhua Hu. Intra-modal proxy learning for zero-shot visual categorization with clip. In *NeurIPS*, 2023. 2

[46] 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 *ICML*, 2021. 1, 2, 5, 6

[47] Karsten Roth, Jae Myung Kim, A. Sophia Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. Waffling around for performance: Visual classification with random words and broad concepts. In *ICCV*, 2023. 1

[48] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. 1, 3

[49] 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. In *NeurIPS*, 2022. 3

[50] Julio Silva-Rodriguez, Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. A closer look at the few-shot adaptation of large vision-language models. In *CVPR*, 2024. 1, 2

[51] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 2

[52] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv*, 2012. 6

[53] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020. 3

[54] Zhen Tan, Alimohammad Beigi, Song Wang, Ruocheng Guo, Amrita Bhattacharjee, Bohan Jiang, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. Large language models for data annotation: A survey. *arXiv*, 2024. 3

[55] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. 1

[56] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 1

[57] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 3

[58] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. In *CVPR*, 2021. 1

[59] Yixuan Wei, Han Hu, Zhenda Xie, Ze Liu, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Improving clip fine-tuning performance. In *ICCV*, 2023. 2

[60] 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 *CVPR*, 2022. 1, 2, 3, 4

[61] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 6

[62] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. In *ICLR*, 2022. 2

[63] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *TMLR*, 2022. 2

[64] Maxime Zanella and Ismail Ben Ayed. Low-rank few-shot adaptation of vision-language models. In *CVPR*, 2024. 2

[65] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 2

[66] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *ECCV*, 2022. 2

[67] Shuai Zhao, Xiaohan Wang, Linchao Zhu, and Yi Yang. Test-time adaptation with clip reward for zero-shot generalization in vision-language models. In *ICLR*, 2024. 3

[68] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022. 1, 2, 6

[69] Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. In *ICCV*, 2023. 2