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.

Enhancing Vision-Language Few-Shot Adaptation with Negative Learning

Ce Zhang Simon Stepputtis Katia Sycara Yaqi Xie
School of Computer Science, Carnegie Mellon University
{cezhang, sstepput, katia, yaqix}@cs.cmu.edu

Abstract

*Large-scale pre-trained Vision-Language Models (VLMs) have exhibited impressive zero-shot performance and transferability, allowing them to adapt to downstream tasks in a data-efficient manner. However, when only a few labeled samples are available, adapting VLMs to distinguish subtle differences between similar classes in specific downstream tasks remains challenging. In this work, we propose a **Sim**ple yet effective **N**egative **L**earning approach, SimNL, to more efficiently exploit the task-specific knowledge from few-shot labeled samples. Unlike previous methods that focus on identifying a set of representative positive features defining "what is a* {CLASS}*", SimNL discovers a complementary set of negative features that define "what is not a* {CLASS}*", providing additional insights that supplement the positive features to enhance task-specific recognition capability. Further, we identify that current adaptation approaches are particularly vulnerable to potential noise in the few-shot sample set. To mitigate this issue, we introduce a plug-and-play few-shot instance reweighting technique to suppress noisy outliers and amplify clean samples for more stable adaptation. Our extensive experimental results across 15 datasets validate that the proposed SimNL outperforms existing state-of-the-art methods on both few-shot learning and domain generalization tasks while achieving competitive computational efficiency. Code is available at https://github.com/zhangce01/SimNL.*

1. Introduction

Over the past decade, deep learning models have achieved remarkable progress in various vision tasks [41, 42], largely due to training on extensive supervised datasets [6, 27]. However, in real-world scenarios, acquiring such large-scale datasets in specific domains (*e.g.*, satellite and aircraft images) is often impractical due to the time-consuming and costly nature of the collection and annotation process. As a result, there is often not enough data in these domains to capture the variability of the novel classes involved, making it challenging to develop robust models from scratch that can effectively generalize to unseen data [47, 58].

Recent advances in Vision-Language Models (VLMs), such as CLIP [38] and ALIGN [21], provide a promising alternative approach for building robust models in a low-data regime. Specifically, by utilizing large-scale pre-training on web-scale datasets, these VLMs have demonstrated remarkable zero-shot performance and transferability [26, 38, 64, 67], which enables the adaptation of these models to downstream tasks in a data-efficient manner. Currently, researchers have developed two primary few-shot adaptation strategies for VLMs: prompt-based learning [45, 50, 72–74] and adapter-style fine-tuning [10, 25, 65, 70, 71], to enable effective task-specific knowledge extraction. However, due to the very limited number of annotated samples available, these methods still struggle to discern subtle differences between similar classes in specific downstream tasks.

In this work, inspired by the negative learning literature [20, 49, 66], we propose a simple yet effective negative learning approach, SimNL, for more effectively adapting VLMs to downstream tasks. Specifically, while previous adaptation methods [71, 73, 74] typically focus on identifying a set of representative features that define "*what is a* {CLASS}" during few-shot adaptation, we propose to discover a complementary set of negative features that define "*what is not a* {CLASS}" to better exploit the limited task-specific knowledge from the few-shot samples. As shown in Figure 1 (*Left*), this complementary set of negative features guides our model to pay attention to more diverse attributes when classifying an image, evaluating both the presence of class-specific characteristics and the absence of characteristics not associated with the class. Building on these discovered features, our SimNL framework incorporates two coordinated classifiers: one performing similarity matching with the positive features and the other performing dissimilarity matching with the negative features. In Figure 1 (*Middle*), we illustrate that when the positive classifier struggles with distinguishing between similar classes, employing the negative one can provide further insights to improve recognition accuracy. The performance comparisons in Figure 1 (*Right*) further highlight that our SimNL method, by

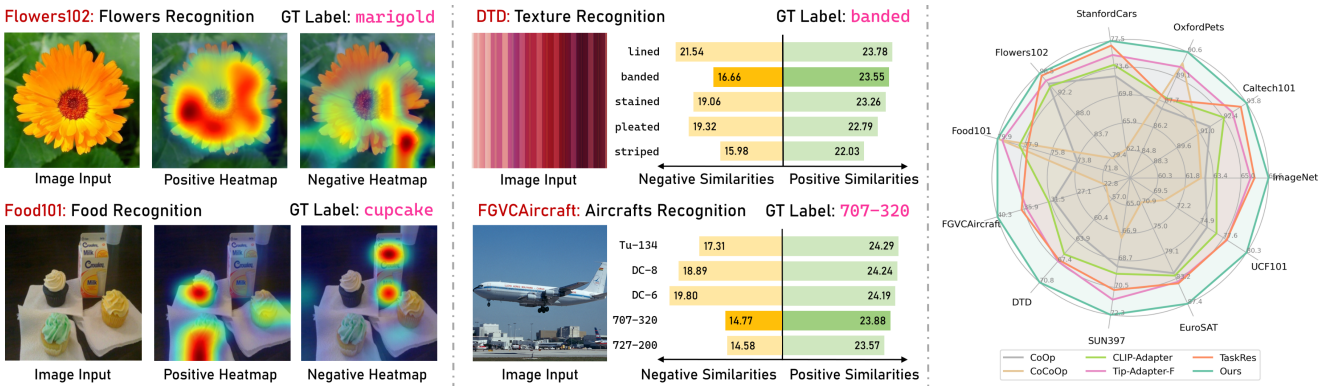

Figure 1. **Negative learning provides complementary information for more accurate recognition**. (*Left*) Grad-CAM [44] visualization of the similarity heatmaps with the learned positive and negative features of the ground-truth class; (*Middle*) Similarities (scaled by 100) to the learned positive and negative features of five similar candidate classes. While the positive branch alone may fail to distinguish among some closely related classes, incorporating the negative classifier, which eliminates certain incorrect classes, enhances the model's ability to accurately identify the true class; (*Right*) Performance comparisons with other state-of-the-art methods in 16-shot scenarios.

incorporating an additional negative classifier, consistently achieves superior few-shot adaptation performance across 11 various datasets when compared to existing state-of-the-art approaches.

Another important challenge often overlooked in existing adaptation methods is the presence of noise in the few-shot sample set. Most existing approaches assume that each few-shot sample is carefully curated to accurately represent its class, but such assumptions rarely holds in practice. Given the extremely small size of the few-shot sample set, the models trained from these samples are particularly vulnerable to noise. To address this important issue, we extend SimNL by reweighting each few-shot instance with a confidence score, which suppresses outliers and amplifies clean samples to ensure stable adaptation. This plug-and-play technique can also be applied to other adapter-style fine-tuning methods, such as Tip-Adapter-F [71], making them more robust to noisy data.

To empirically validate the effectiveness of our SimNL, we conduct comprehensive evaluations on few-shot learning and domain generalization tasks across 15 diverse recognition datasets. These experimental results demonstrate the effectiveness of SimNL in adapting VLMs for downstream tasks and verify its superior robustness to distribution shifts. We also demonstrate that our proposed instance reweighting significantly enhances the model's robustness to label noises in the few-shot sample set. Additionally, SimNL also exhibits competitive efficiency.

Our primary contributions are summarized as follows:

- We propose a simple yet effective negative learning approach, *i.e.*, SimNL, to efficiently adapt CLIP to downstream tasks. Specifically, SimNL introduces an innovative application of negative learning to adapter-style fine-tuning for few-shot adaptation of VLMs.
- To mitigate the impact of noisy samples in few-shot adaptation, we introduce a plug-and-play few-shot instance reweighting technique that assigns non-uniform confidence scores to each sample.
- Through extensive experiments, we demonstrate that our SimNL consistently outperforms other state-of-the-art methods across 15 diverse recognition datasets.

2. Related Work

Efficient Adaptation for VLMs. In recent years, extensive Vision-Language Models (VLMs) have been developed to bridge the vision and language modalities through large-scale pre-training [21, 38]. Given the substantial size of the VLMs, recent research efforts [10, 69, 71, 74] are focusing on the development of lightweight fine-tuning techniques to efficiently adapt VLMs for downstream visual tasks. These methods can generally be divided into two categories: *prompt-based learning* and *adapter-style fine-tuning*. Specifically, prompt-based learning methods [4, 50, 61, 73, 74] aim to optimize the input prompts from downstream data, while adapter-style fine-tuning approaches [10,52,65,71,76] directly tune the extracted visual and textual representations. In this work, we aim to enhance adapter-style fine-tuning by incorporating negative learning into vision-language few-shot adaptation. We also empirically validate that leveraging negative cues from CLIP can effectively improve both few-shot classification performance and generalization capability.

Few-Shot Learning. Few-shot learning aims to quickly adapt a model to new categories using only a few examples. Traditional few-shot learning methods primarily rely on meta-learning and can be roughly separated into two groups: metric-based methods [46,54,68] and optimization-based methods [12,39,43]. However, these methods depend on training with base datasets, limiting their applicability in real-world scenarios. Recent developments in large-scale

pre-trained VLMs [1, 38] present a promising alternative due to their exceptional zero-shot capabilities. Researchers have demonstrated that with efficient adaptation, VLMs can also excel in few-shot learning tasks [17, 71, 73, 74].

Negative Learning. Given that obtaining typical positive labels (which indicate the categories an image belongs to) can be costly and labor-intensive, researchers have proposed an alternative approach known as the indirect negative learning [20, 66] paradigm. This method focuses on learning from negative/complementary labels, which specify the categories to which an image does *not* belong. In recent years, negative learning has been effectively applied to various vision applications, such as image recognition [11, 20, 23], few-shot learning [18, 60], and semantic segmentation [37, 57]. In this work, empowered by the strong negative understanding capabilities of the large-scale pre-trained CLIP model [19, 32, 56], we enable an alternative approach to negative learning, where we explicitly train a negative classifier without the need for complementary labels. We provide more discussions on the differences between our method and traditional negative learning in Section B.1. We also note that recent work on TDA [22] shares a similar approach to ours, as both methods incorporate negative caches to enhance the generalizability of VLMs. However, their method focuses on test-time adaptation, where training data is unavailable, making it unsuitable for direct application in few-shot adaptation scenarios.

3. Method

In this section, we introduce SimNL, a novel approach to vision-language few-shot adaptation, as shown in Figure 2.

3.1. Problem Formulation

We consider the problem of C-way K-shot few-shot classification problem, in which only K labeled examples are provided for each of the C classes. Let the feature space be $\mathcal{X}$ and the label space be $\mathcal{Y} = \{1, ..., C\}$, the few-shot instance $\boldsymbol{x} \in \mathcal{X}$ and its labels $y \in \mathcal{Y}$ are sampled from probability distribution $P(\boldsymbol{x}, y)$. The corresponding one-hot encoded labels are represented by $\boldsymbol{y} \in \{0, 1\}^C$.

Positive Learning. In this study, we employ a CLIP-based classifier with parameters θ, denoted as $\mathcal{F}_\theta : \mathcal{X} \to \mathbb{R}^C$, to project each input into a C-dimensional score space. The classification probabilities are obtained using $\boldsymbol{p} = \texttt{Softmax}(\mathcal{F}_\theta(\boldsymbol{x})) \in \Delta^{C-1}$, where Δ^{C-1} represents the $(C-1)$-dimensional probability simplex. Our goal is to learn a robust classifier $\mathcal{F}_\theta$ that minimizes the expected risk; that is,

$$\min_{\theta} \mathcal{R}(\mathcal{F}_\theta) = \mathbb{E}_{(\boldsymbol{x},y)\sim P(\boldsymbol{x},y)} \left[\mathcal{L}\left(\mathcal{F}_\theta(\boldsymbol{x}), y\right)\right],$$
$$\text{where } \mathcal{L}\left(\mathcal{F}_\theta(\boldsymbol{x}), y\right) = -\sum_{k=1}^{C} \boldsymbol{y}_k \log \boldsymbol{p}_k, \tag{1}$$

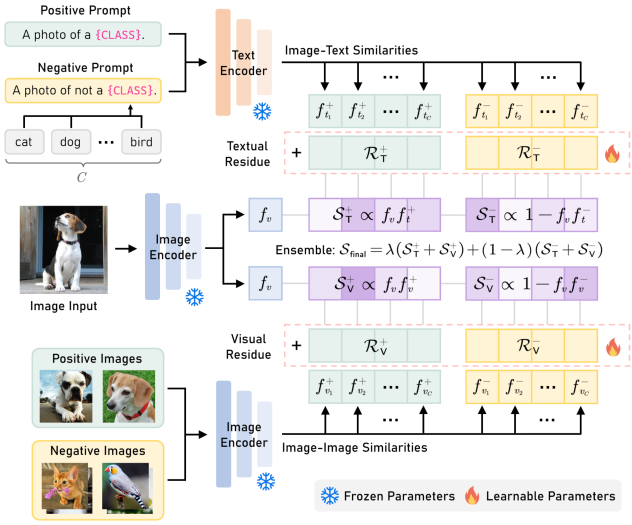

Figure 2. **An overview of our proposed SimNL**. We construct and learn the positive and negative CLIP-based classifiers across visual and textual modalities. Given an image to be classified, the classification logit for a specific class increases when the image feature f_v closely aligns with the corresponding positive features f_t^+, f_v^+ and diverges from negative features f_t^-, f_v^-.

where $\boldsymbol{p}_k$ denotes the probability for the k-th class and $\mathcal{L}\left(\mathcal{F}_\theta(\boldsymbol{x}), y\right)$ computes the cross-entropy loss. We aim to maximize the probability $\boldsymbol{p}_y \to 1$ for the true label.

Negative Learning. In this work, we apply the concept of negative learning to vision-language few-shot adaptation by employing a distinct CLIP-based negative classifier $\mathcal{G}_\varphi : \mathcal{X} \to \mathbb{R}^C$ with parameters φ. This classifier predicts the negative probability $\bar{\boldsymbol{p}} = \texttt{Softmax}(\mathcal{G}_\varphi(\boldsymbol{x}))$ that the image does not belong to specific classes. The expected classification risk for $\mathcal{G}_\varphi$ can be written as

$$\min_{\varphi} \mathcal{R}(\mathcal{G}_\varphi) = \mathbb{E}_{(\boldsymbol{x},y)\sim P(\boldsymbol{x},y)} \left[\mathcal{L}\left(\mathcal{G}_\varphi(\boldsymbol{x}), y\right)\right],$$
$$\text{where } \mathcal{L}\left(\mathcal{G}_\varphi(\boldsymbol{x}), y\right) = -\sum_{k=1}^{C} \boldsymbol{y}_k \log(1 - \bar{\boldsymbol{p}}_k). \tag{2}$$

By optimizing this risk, we aim to reduce the negative probability $\bar{\boldsymbol{p}}_y \to 0$ for the true label.

Finally, we ensemble the optimized classifiers from both positive and negative learning to create an enhanced classifier $\mathcal{F}_\theta \oplus \mathcal{G}_\varphi : \mathcal{X} \to \mathbb{R}^C$ for testing. This is equivalent to combining the positive and negative predicted probabilities as $\hat{\boldsymbol{p}} = \lambda \boldsymbol{p} + (1-\lambda)(1-\bar{\boldsymbol{p}})$, where λ is a balancing hyper-parameter.

Differences with Contrastive Learning. Note that the concept of "negative" differs in our negative learning approach compared to contrastive learning: (1) In our negative learning, "negative" specifically refers to a negative classifier. We explicitly train another negative classifier $\mathcal{G}_\varphi$ to ensemble with the positive classifier; (2) In con-

trastive learning, "negative" refers to the negative sample pairs that are constructed and utilized during training. In Appendix B.2, we show that the training objectives for both the positive and negative CLIP-based classifiers are contrastive in nature.

3.2. Preliminary: A Revisit of CLIP

In this work, our classifiers are based on CLIP's pre-trained visual encoder $\mathcal{F}_{\mathsf{V}}$ and textual encoder $\mathcal{F}_{\mathsf{T}}$. Specifically, CLIP [38] performs zero-shot predictions by assessing the similarity between the image feature and the text feature specific to each class $c \in \{1, ..., C\}$ as follows:

$$f_v = \mathcal{F}_{\mathsf{V}}(\boldsymbol{x}), \quad f_{t_c} = \mathcal{F}_{\mathsf{T}}(\mathcal{T}_c),$$
$$\mathbb{P}(y = c|\boldsymbol{x}) = \frac{\exp\left(\cos\left(f_{t_c}, f_v\right)/t\right)}{\sum_{t'} \exp\left(\cos\left(f_{t'}, f_v\right)/t\right)}, \tag{3}$$

where $\boldsymbol{x}$ is the input image, and $\mathcal{T}_c$ represents the text description for class c (*e.g.*, "`A photo of a` $\{\texttt{CLASS}_c\}$"). The parameter t is the temperature hyperparameter, and $\cos\left(f_t, f_v\right) = f_v^\top f_t$ computes the cosine similarity. To streamline this process, we can precompute a textual cache, which concatenates the textual features associated with each class, denoted as $\mathsf{T}_{\texttt{cache}} = [f_{t_1}\, f_{t_2}\, \cdots\, f_{t_C}]^\top \in \mathbb{R}^{C\times d}$. Subsequently, we can obtain the final prediction $\mathbb{P}(y|\boldsymbol{x})$ via vectorized similarity matching:

$$\mathcal{S} = f_v {\mathsf{T}_{\texttt{cache}}}^\top \in \mathbb{R}^C, \quad \mathbb{P}(y|\boldsymbol{x}) = \texttt{Softmax}\left(\mathcal{S}\right). \tag{4}$$

3.3. Learning the Positive Classifier

In this section, we focus on the *positive* perspective, namely, constructing and optimizing a learnable CLIP-based positive classifier. This similarity-based classifier is designed to identify a set of representative features that enhance the accuracy of predicting the true class of the input image.

Positive Textual Branch. As shown in Eq. (4), CLIP can make zero-shot predictions utilizing a textual cache, which stores the textual features from positive text descriptions (*e.g.*, "`A photo of a` $\{\texttt{CLASS}\}$"). To avoid ambiguity, we denote this cache as $\mathsf{T}^+_{\texttt{cache}} = [f^+_{t_1}\, f^+_{t_2}\, \cdots\, f^+_{t_C}]^\top \in \mathbb{R}^{C\times d}$. With a small set of annotated training images, we can update the textual features by introducing a group of learnable residual parameters $\mathcal{R}^+_{\mathsf{T}} \in \mathbb{R}^{C\times d}$ to integrate task-specific knowledge:

$$\mathsf{T}^+_{\texttt{cache}} \leftarrow \texttt{Normalize}\left(\mathsf{T}^+_{\texttt{cache}} + \mathcal{R}^+_{\mathsf{T}}\right),$$
$$\mathcal{S}^+_{\mathsf{T}} = f_v {\mathsf{T}^+_{\texttt{cache}}}^\top \in \mathbb{R}^C. \tag{5}$$

Here, `Normalize` denotes the L2-normalization applied to each row of the matrix.

Positive Visual Branch. We further extend the recognition capability of the CLIP model by constructing a visual cache-based classifier, which operates by measuring image-image similarities between the input image feature and few-shot image features. Specifically, we store all CK image features in a precomputed visual cache, denoted as $\mathsf{V}^+_{\texttt{cache}} = [f^{(1)}_{v_1}\, f^{(2)}_{v_1}\, \cdots\, f^{(K)}_{v_C}]^\top \in \mathbb{R}^{CK\times d}$. Their one-hot labels are also correspondingly recorded in $L \in \mathbb{R}^{CK\times C}$. Given an image feature f_v to be classified, we calculate its image-image affinities and obtain the logits $\mathcal{S}^+_{\mathsf{V}}$:

$$\mathsf{V}^+_{\texttt{cache}} \leftarrow \texttt{Normalize}\left(\mathsf{V}^+_{\texttt{cache}} + \mathcal{R}^+_{\mathsf{V}}\right),$$
$$\mathcal{S}^+_{\mathsf{V}} = \mathcal{A}\left(f_v {\mathsf{V}^+_{\texttt{cache}}}^\top\right) L \in \mathbb{R}^C, \tag{6}$$

where $\mathcal{A}\left(f_v {\mathsf{V}^+_{\texttt{cache}}}^\top\right) = \alpha \exp\left(-\beta\left(1 - f_v {\mathsf{V}^+_{\texttt{cache}}}^\top\right)\right)$ calculates the affinity, α represents a balance scalar and β denotes a modulating parameter. Learnable residue $\mathcal{R}^+_{\mathsf{V}} \in \mathbb{R}^{C\times d}$ is also introduced, which is broadcast to $\mathbb{R}^{CK\times d}$ and added to the visual cache to refine the visual features.

3.4. Learning the Negative Classifier

Having introduced the positive classifier, we now explore constructing and learning a *negative* dissimilarity-based classifier, which directs CLIP towards more confidently excluding incorrect classes based on the given input image. Conceptually, our goal is to mine a general negative feature for each class c, which is absent in samples from class c but present in samples from all other classes.

Negative Textual Branch. Recall that the positive textual cache, denoted as $\mathsf{T}^+_{\texttt{cache}}$, is constructed based on the positive class descriptor prompt such as "`A photo of a` $\{\texttt{CLASS}\}$." Conversely, we introduce a set of negative prompts (*e.g.*, "`A photo of not a` $\{\texttt{CLASS}\}$," "`A photo without` $\{\texttt{CLASS}\}$") that convey opposite semantics meanings. By leveraging the textual features f^-_t derived from the negative text descriptions linked with these prompts, we again precompute a weight matrix and construct negative textual cache $\mathsf{T}^-_{\texttt{cache}} = [f^-_{t_1}\, f^-_{t_2}\, \cdots\, f^-_{t_C}]^\top \in \mathbb{R}^{C\times d}$. Intuitively, if the feature of an input image f_v closely resembles the negative textual feature for a specific class c, it strongly suggests that the image does not belong to class c, *i.e.*, $\mathbb{P}(y = c|\boldsymbol{x}) \propto 1 - f_v^\top f^-_{t_c}$. Through supervised task-specific training, we optimize a learnable residue $\mathcal{R}^-_{\mathsf{T}} \in \mathbb{R}^{C\times d}$ and obtain the negative predictions based on dissimilarities $1 - f_v \mathsf{T}^-_{\texttt{cache}}$:

$$\mathsf{T}^-_{\texttt{cache}} \leftarrow \texttt{Normalize}\left(\mathsf{T}^-_{\texttt{cache}} + \mathcal{R}^-_{\mathsf{T}}\right),$$
$$\mathcal{S}^-_{\mathsf{T}} = \delta_{\mathsf{T}}\left(1 - f_v {\mathsf{T}^-_{\texttt{cache}}}^\top\right) \in \mathbb{R}^C, \tag{7}$$

where δ_{T} is a fixed scaling parameter that adjusts $\mathcal{S}^-_{\mathsf{T}}$ to match the mean value of $\mathcal{S}^+_{\mathsf{T}}$.

Negative Visual Branch. In the visual domain, we also similarly pursue some negative image features to be non-representative of a specific class, which means that the higher the similarity to them, the lower the probability that the image is classified to that class, *i.e.*, $\mathbb{P}(y = c|\boldsymbol{x}) \propto 1 - f_v^\top f^-_{v_c}$. To achieve this, we randomly

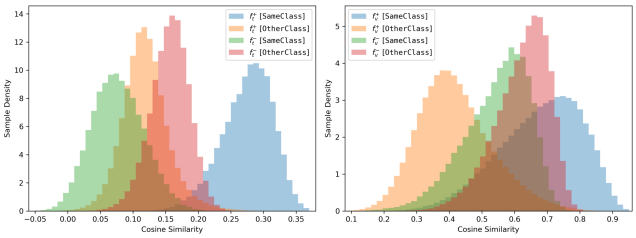

Figure 3. **Visualization of cosine similarities on ImageNet [6] validation set**. We present distributions of pairwise similarities between the input image feature and both the learned positive and negative features from textual (*Left*) and visual (*Right*) modalities.

select one image from each of the $C-1$ classes and compute the average of their extracted features to represent the negative features. In this way, we can get a total of K negative visual features for each of the C classes, thereby constructing a negative visual cache $\mathsf{V}_{\texttt{cache}}^{-} \in \mathbb{R}^{CK\times d}$. Symmetric to Eq. (6), we compute the negative affinities to obtain the classification logits:

$$\begin{aligned} \mathsf{V}_{\texttt{cache}}^{-} &\leftarrow \mathsf{Normalize}\left(\mathsf{V}_{\texttt{cache}}^{-} + \mathcal{R}_{\mathsf{V}}^{-}\right), \\ \mathcal{S}_{\mathsf{V}}^{-} &= \delta_{\mathsf{V}}\mathcal{A}\left(1 - f_v \mathsf{V}_{\texttt{cache}}^{-\top}\right) L \in \mathbb{R}^{C}. \end{aligned} \tag{8}$$

where δ_{V} is another fixed scaling parameter that adjusts $\mathcal{S}_{\mathsf{V}}^{-}$ to match the mean value of $\mathcal{S}_{\mathsf{V}}^{+}$. Here, we also introduce a set of learnable parameters $\mathcal{R}_{\mathsf{V}}^{-} \in \mathbb{R}^{C\times d}$ to refine the negative visual features.

Final Inference. As discussed in Section 3.1, we ensemble the predictions from both classifiers to derive the final classification scores and predictions:

$$\begin{aligned} \mathcal{S}_{\mathsf{final}} &= \lambda\left(\mathcal{S}_{\mathsf{T}}^{+} + \mathcal{S}_{\mathsf{V}}^{+}\right) + (1-\lambda)\left(\mathcal{S}_{\mathsf{T}}^{-} + \mathcal{S}_{\mathsf{V}}^{-}\right), \\ \mathbb{P}(y|\boldsymbol{x}) &= \texttt{Softmax}\left(\mathcal{S}_{\mathsf{final}}\right). \end{aligned} \tag{9}$$

Here, λ serves as a tuning hyper-parameter to balance the contribution of positive and negative logits. Throughout the training process, the collection of learnable parameters $\mathcal{R} = \{\mathcal{R}_{\mathsf{T}}^{+}, \mathcal{R}_{\mathsf{V}}^{+}, \mathcal{R}_{\mathsf{T}}^{-}, \mathcal{R}_{\mathsf{V}}^{-}\}$ is updated via stochastic gradient descent with a cross-entropy loss.

Qualitative Visualizations. Figure 3 qualitatively shows the effectiveness of both the positive and negative classifiers we designed. Specifically, we visualize the distributions of pairwise cosine similarities between the input image feature f_v and both the learned positive and negative features from two modalities $(f_t^+, f_t^-, f_v^+, f_v^-)$ on the validation set of ImageNet [6]. We can observe the following statistical patterns: (1) As we expected, the input image feature is more similar with positive features from the same class (blue > orange) and negative features from other classes (red > green) across both two modalities; (2) Within the visual modality, the similarity distribution of negative features occupies an intermediate position between the positive features of the same and different

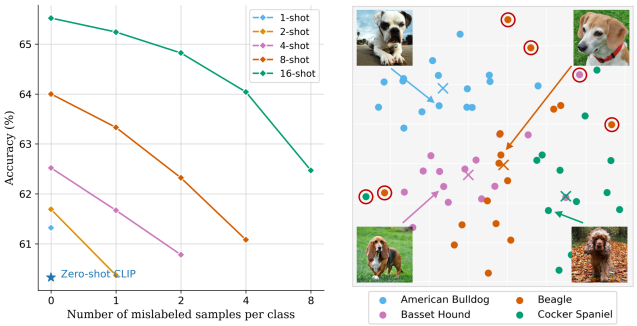

Figure 4. **Few-shot instance reweighting**. (*Left*) The performance of Tip-Adapter-F [71] degrades drastically when label noise exists in the few-shot sample set; (*Right*) t-SNE [53] visualization of visual features for 4 random classes from the OxfordPets [35] dataset, where some outliers are marked with red circles.

classes (orange < green/red < blue). This is because the negative visual features are constructed by averaging the image features across all other classes, inherently leading to a more generic representation that lacks the distinctiveness characteristic of positive features within a single class; (3) Within the textual modality, negative features tend to be less similar to the input image compared to positive features (red < blue, green < orange), since the negative prompts are less common in CLIP [38] training corpus.

3.5. Few-Shot Instance Reweighting

In the few-shot adaptation setting, our classifier is trained using only a limited number of samples, with each sample making a significant contribution to the formation of the final decision boundary. Consequently, our VLMs are particularly vulnerable to potential noise in the few-shot sample set. In Figure 4 (*Left*), we simulate real-world noise scenarios by randomly flipping the labels of a portion of support samples. We demonstrate that the performance of Tip-Adapter-F [71] drastically decreases from 65.52% to 62.47% when the labels of 8 out of the 16 samples per class are randomly flipped. Moreover, even if the labels are all correct, we recognize that not every image is of high quality or equally representative of its respective class, as shown in Figure 4 (*Right*). To address this issue, we have developed a few-shot instance reweighting technique to assign non-uniform confidence scores to each sample, effectively downweighting outliers (or mislabelled samples) and prioritizing more representative samples.

Specifically, our proposed instance reweighting is based on an intuitive assumption: the representative image feature is closer to other image features from the same class than those low-quality outliers. Based on this assumption, given the K-shot image features $\{f_{v_c}^{(i)}\}_{i=1}^{K}$ from a specific class c, we calculate the average cosine similarities of each image

feature to others:

$$\mathsf{d}_c^{(i)} = \frac{1}{K-1}\sum_{j,\,j\neq i}\cos\left(f_{v_c}^{(i)}, f_{v_c}^{(j)}\right) = \frac{1}{K-1}\sum_{j,\,j\neq i} f_{v_c}^{(i)\top} f_{v_c}^{(j)}. \tag{10}$$

We then assign non-uniform weights $w_c^{(i)}$ and compute the reweighted confidences $\ell_c^{(i)}$ for all K-shot image features based on their average similarities to others:

$$w_c^{(i)} = \frac{\exp(\mathsf{d}_c^{(i)}/\tau)}{\sum_{i'}\exp(\mathsf{d}_c^{(i')}/\tau)}, \quad \ell_c^{(i)} = K w_c^{(i)}, \tag{11}$$

where τ is a temperature hyper-parameter to control the intensity of our instance reweighting.

This process is applied across each class, wherein the original one-hot labels L, are reweighted using the new confidence values calculated in Eq. (11) to yield $\mathbb{L}$. By incorporating this reweighting technique into the visual branches of both classifiers, we can reformulate Eqs. (6) and (8) as follows:

$$\mathcal{S}_{\mathsf{V}}^{+} = \mathcal{A}\left(f_v \mathsf{V}_{\mathtt{cache}}^{+\top}\right)\mathbb{L}^{+},\ \mathcal{S}_{\mathsf{V}}^{-} = \delta_{\mathsf{V}}\mathcal{A}\left(1 - f_v \mathsf{V}_{\mathtt{cache}}^{-\top}\right)\mathbb{L}^{-}. \tag{12}$$

4. Experiments

In this section, we conduct extensive experiments on two tasks across 15 datasets. These results demonstrate that our proposed method is simple yet highly effective, surpassing other state-of-the-art methods in both few-shot adaptation and domain generalization capabilities.

4.1. Experimental Settings

Tasks and Datasets. To validate the effectiveness of SimNL, we evaluate our method on two standard benchmarking tasks: few-shot learning and domain generalization, respectively. For few-shot learning task, we comprehensively evaluate our method on 11 well-known image classification benchmarks: ImageNet [6], Caltech101 [9], OxfordPets [35], StandfordCars [24], Flowers102 [33], Food101 [2], FGVCAircraft [30], DTD [5], SUN397 [63], EuroSAT [14], and UCF101 [48]. For domain generalization, we evaluate the generalizability of our SimNL on 4 variants of ImageNet: ImageNet-V2 [40], ImageNet-Sketch [55], ImageNet-A [16], and ImageNet-R [15]. Moreover, we also explore the adaptation of VLMs using a noisy few-shot sample set from ImageNet [6], where labels are randomly flipped to simulate real-world scenarios.

Implementation Details. Following previous works, we adopt ResNet-50 [13] backbone as the visual encoder of CLIP in our experiments by default. We adopt prompt ensembling, leveraging textual prompts from CLIP [38] to enhance model performance. For the negative prompts we used for each dataset, please kindly refer to Appendix C.2. We set the hyper-parameters λ and τ as 0.75 and 1, respectively. Our SimNL is trained using the AdamW [29] optimizer with a cosine scheduler [28]. The batch size is set to 256. For $\mathcal{R}_{\mathsf{T}}^{+}$ and $\mathcal{R}_{\mathsf{V}}^{+}$, the learning rate is set to 0.0001, while for $\mathcal{R}_{\mathsf{T}}^{-}$ and $\mathcal{R}_{\mathsf{V}}^{-}$, the learning rate is set to 0.0005. Our model is trained for 200 epochs on the EuroSAT [14] dataset, and for 20 epochs on all other datasets. To ensure the reliability of our results, we perform each experiment three times using different initialization seeds and report the mean accuracy achieved. All experiments are conducted on a single 48GB NVIDIA RTX 6000 Ada GPU.

Baselines. We compare our proposed method with the following state-of-the-art methods: zero-shot and linear probe CLIP [38], CoOp [74], CoCoOp [73], ProGrad [75], CLIP-Adapter [10], Tip-Adapter-F [71], TPT [45], TaskRes [65], and GraphAdapter [25]. For a fair comparison, we directly report the results of these baselines from their respective original papers.

4.2. Results and Analysis

Few-Shot Learning. In Figure 5, we compare the few-shot learning performance of our proposed method with other state-of-the-art methods on 11 image classification datasets. In the top-left sub-figure, we also present the average classification accuracy across all 11 datasets. The results indicate that our method consistently outperforms other methods across various few-shot learning settings by substantial margins. Moreover, our proposed method demonstrates more pronounced performance improvements in specialized classification tasks, such as satellite image and texture classification on the EuroSAT [14] and DTD [5] datasets. With 16-shot training on these two datasets, our method surpasses Tip-Adapter-F [71] by a notable 3.56% and 3.50%, respectively. For full numerical results, please refer to Table A1 in Appendix A.1. Overall, the consistently superior performance on 11 datasets fully demonstrates the general effectiveness of our proposed approach.

Robustness to Natural Distribution Shifts. In Table 1, we compare the generalizability of our SimNL with other methods in the presence of distribution shifts. Specifically, all the models are trained solely on 16-shot ImageNet [6], and directly tested on 4 out-of-distribution ImageNet variant datasets. As shown in Table 1, our method not only achieves state-of-the-art performance on the source dataset but also attains an average performance gain of 1.18% across 4 out-of-distribution (OOD) target datasets. These experimental results indicate that by enabling adaptation from both positive and negative perspectives, our method enhances robustness against distribution shifts.

Robustness to Label Noise. In Table 2, we report the performance of Tip-Adapter-F [71] and our SimNL on a noisy 16-shot ImageNet [6], where we randomly flip 10% to 50% labels. As shown, applying our instance reweighting enhances both Tip-Adapter-F and SimNL's robustness

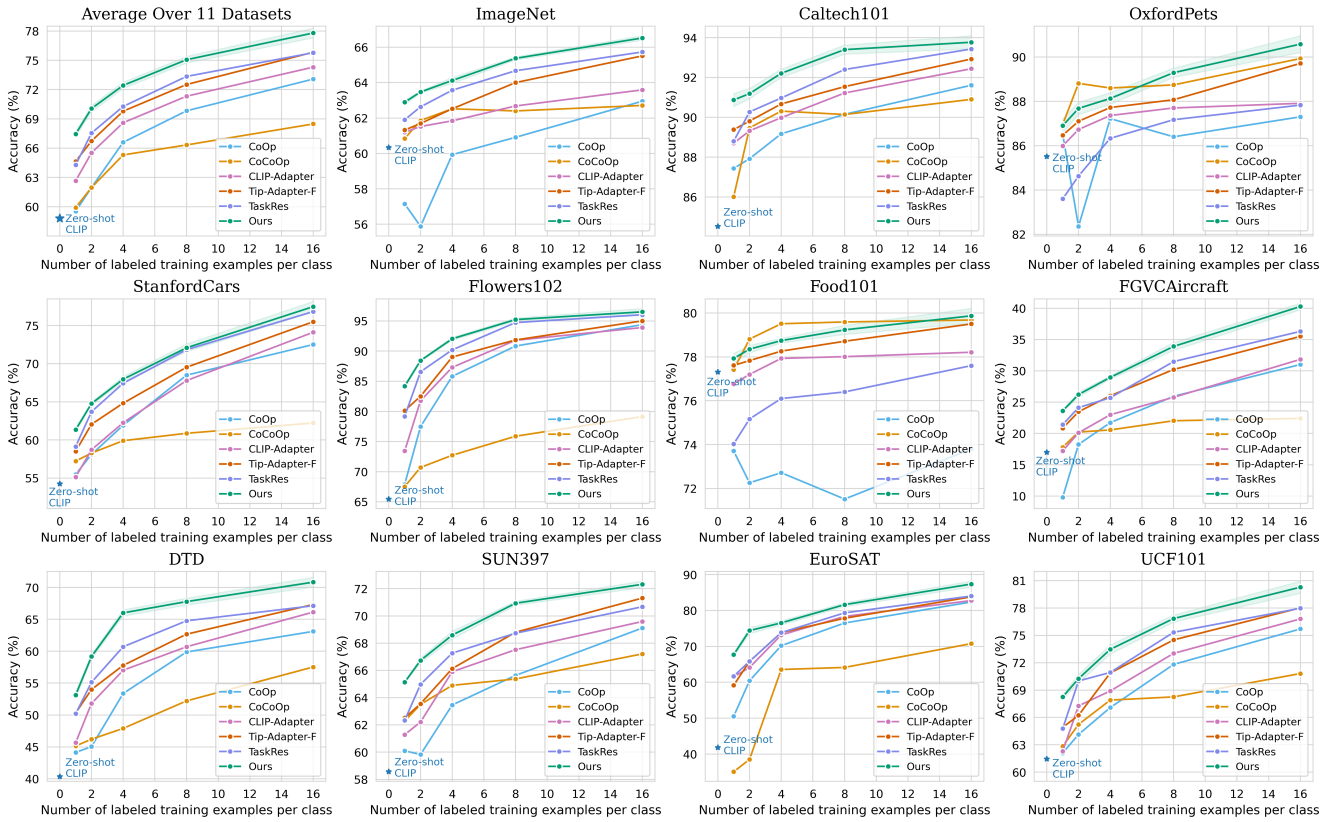

Figure 5. **Performance comparisons on few-shot learning on 11 image classification datasets**. For each dataset, we report the mean accuracy and 95% confidence interval over 3 random seeds of our SimNL on 1-/2-/4-/8-/16-shot settings.

Table 1. **Performance comparison on robustness to distribution shifts**. All the models are trained on 16-shot ImageNet [6] and directed tested on the OOD target datasets. The best results are in **bold** and the second best are underlined.

Method	Source	Target				
	ImageNet	-V2	-Sketch	-A	-R	Avg.
Zero-Shot CLIP [38]	60.33	53.27	35.44	21.65	56.00	41.59
Linear Probe CLIP [38]	56.13	45.61	19.13	12.74	34.86	28.09
CoOp [74]	62.95	55.40	34.67	23.06	56.60	42.43
CoCoOp [73]	62.71	55.72	34.48	23.32	57.74	42.82
ProGrad [75]	62.17	54.70	34.40	23.05	56.77	42.23
TPT [45]	60.74	54.70	35.09	**26.67**	59.11	43.89
TaskRes [65]	64.75	56.47	35.83	22.80	60.70	43.95
GraphAdapter [25]	64.94	56.58	35.89	23.07	60.86	44.10
SimNL (Ours)	**66.52**	**57.87**	**36.38**	25.73	**61.12**	**45.28**

Table 2. **Comparison of robustness to label noise on noisy 16-shot ImageNet [6]**. We apply our instance reweighting technique to both Tip-Adapter-F [71] and our SimNL, and report the performance across four levels of noise.

Method	0%	10%	25%	50%
Tip-Adapter-F [71]	65.52	64.93	64.04	62.47
+ Reweighting	**65.64**	**65.25**	**64.55**	**63.39**
Performance Gain	+0.12	+0.32	+0.51	+0.92
SimNL (Ours)	66.31	65.54	64.77	63.37
+ Reweighting	**66.52**	**65.82**	**65.16**	**64.02**
Performance Gain	+0.21	+0.28	+0.39	+0.65

to label noise. Notably, our reweighting technique also improves performance when no label noise is introduced, as it can identify and downweight outliers during adaptation.

Efficiency Comparison. We also compare the efficiency of SimNL with existing methods in Table 3. Our method achieves the highest accuracy while also exhibiting advantageous computational efficiency: (1) Compared to prompt-based learning methods such as CoOp [74] and ProGrad [75], our proposed method requires approximately 300× less training time and over 1000× fewer FLOPs since we do not need to propagate gradients through the textual encoder; (2) Compared to adapter-style fine-tuning methods, our SimNL requires 10× less training time than CLIP-Adapter [10], and demands 3× fewer FLOPs than Tip-Adapter-F [71] and GraphAdapter [25].

4.3. Ablation Studies

Effectiveness of Different Components. In Table 4, we conduct a systematic analysis of the impacts of various components within our SimNL framework. More specifically, we assess the performance of four distinct SimNL variants, each configured to allow two learnable residues to be updated while keeping the others fixed. We observe

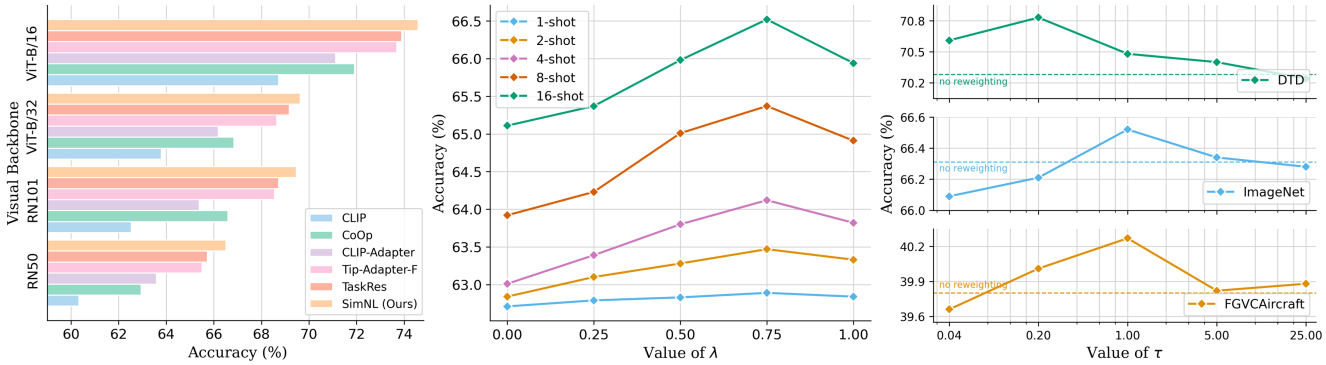

Figure 6. **More ablation results**. (*Left*) Performance comparison of our SimNL with others on few-shot learning using different visual backbones; (*Middle*) Sensitivity analysis of λ from Eq. (9) on ImageNet [6]; (*Right*) Sensitivity analysis of τ from Eq. (11) on 3 datasets.

Table 3. **Efficiency comparison with other existing methods on 16-shot ImageNet [6]**. We report the training time, number of epochs, FLOPs, and the number of parameters for each method.

Method	Training	Epochs	GFLOPs	Param.	Accuracy
CLIP [38]	-	-	-	-	60.33
CoOp [74]	14 hr	200	>10	**0.01M**	62.95
ProGrad [75]	17 hr	200	>10	**0.01M**	63.45
CLIP-Adapter [10]	50 min	200	**0.004**	0.52M	63.59
Tip-Adapter-F [71]	**5 min**	**20**	0.030	16.38M	65.51
GraphAdapter [25]	**5 min**	**20**	0.030	4.15M	65.70
SimNL (Ours)	**5 min**	**20**	0.009	4.15M	**66.52**

Table 4. **Ablation studies for different variants of our method**. We evaluate the few-shot adaptation capabilities of four variants and report their average performance across all 11 datasets.

Method	$\mathcal{R}_{\mathrm{T}}^{+}$	$\mathcal{R}_{\mathrm{T}}^{-}$	$\mathcal{R}_{\mathrm{V}}^{+}$	$\mathcal{R}_{\mathrm{V}}^{-}$	1-shot	2-shot	4-shot	8-shot	16-shot
SimNL-T	✓	✓	✗	✗	67.08	69.70	71.86	74.17	77.08
SimNL-V	✗	✗	✓	✓	64.60	66.03	68.25	71.84	74.41
SimNL-P	✓	✗	✓	✗	66.72	69.30	71.49	73.98	76.76
SimNL-N	✗	✓	✗	✓	66.23	68.10	70.55	72.33	74.17
SimNL	✓	✓	✓	✓	**67.45**	**70.06**	**72.43**	**75.06**	**77.80**

that the textual variant (SimNL-T) and the positive variant (SimNL-P) generally exhibit superior efficiency compared to their visual and negative counterparts. The full SimNL method, which integrates all four branches, surpasses the individual variants by achieving the highest average accuracy of 77.80% in the 16-shot scenario.

Effects of Different Visual Backbones. We also implement our SimNL with various visual encoders, including ResNet [13] and ViT [8], and evaluate their performance against other adaptation methods in Figure 6 (*Left*). We can see that our SimNL consistently exceeds other methods across all visual backbones, indicating the general effectiveness of our negative learning approach.

Sensitivity Analysis of Hyper-Parameters. We provide sensitivity analysis for the hyper-parameters λ and τ in Figure 6. The hyper-parameter λ from Eq. (9) controls the combination of the positive and negative predictions. In Figure 6 (*Middle*), we can observe that setting $\lambda = 0.75$ consistently yields the optimal performance. Moreover, in

Table 5. **Scalability to more shots**. We show performance comparison with Tip-Adapter(-F) on ImageNet at higher-shot settings.

Method	16-shot	32-shot	64-shot	128-shot
CLIP [38]		†0-shot: 60.33		
Tip-Adapter [71]	62.03	62.51	62.88	63.15
Tip-Adapter-F [71]	65.47	66.58	67.96	69.74
SimNL (Ours)	**66.52**	**67.68**	**69.01**	**70.98**

Figure 6 (*Right*), we adjust the temperature hyper-parameter τ in Eq. (11) from 0.04 to 25 and report the 16-shot accuracy of our method. We can see that by reweighting the few-shot samples, we achieve performance improvements ranging from 0.21% to 0.55% across three datasets.

Scalability to More Shots. In Table 5, we demonstrate that our SimNL scales effectively to more-shot settings, consistently outperforming Tip-Adapter-F [71] by 1.05% to 1.24% as the number of shots increases.

5. Conclusion

In this work, we introduce SimNL, a simple and effective approach that applies the concept of negative learning to vision-language few-shot adaptation. Specifically, we transform the open-vocabulary CLIP model into a negative classifier, which is further optimized to exclude incorrect classes based on the image input. During inference, we conduct simultaneous positive classification and negative exclusion to enhance the overall prediction accuracy. Furthermore, we develop an unsupervised few-shot instance reweighting approach to mitigate the adverse effects of noisy image samples during few-shot adaptation . Comprehensive evaluations on 15 diverse datasets demonstrate that our proposed SimNL outperforms the state-of-the-art methods in both few-shot learning and domain generalization tasks, while maintaining competitive efficiency.

Acknowledgements. This work has been funded in part by the Army Research Laboratory (ARL) award W911NF-23-2-0007, DARPA award FA8750-23-2-1015, and ONR award N00014-23-1-2840.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 35:23716–23736, 2022. 3

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, pages 446–461. Springer, 2014. 6, 12, 13, 15

[3] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tiejun Huang, Shiliang Pu, and Yonghong Tian. Learning open set network with discriminative reciprocal points. In *ECCV*, pages 507–522. Springer, 2020. 15

[4] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. PLOT: Prompt learning with optimal transport for vision-language models. In *ICLR*, 2023. 2

[5] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014. 6, 13, 15

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1, 5, 6, 7, 8, 12, 13, 14, 15

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019. 16

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 8

[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. *Computer Vision and Image Understanding*, 106(1):59–70, 2007. 6, 13, 15

[10] 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*, 132:581–595, 2024. 1, 2, 6, 7, 8, 13

[11] Yi Gao and Min-Ling Zhang. Discriminative complementary-label learning with weighted loss. In *ICML*, pages 3587–3597. PMLR, 2021. 3

[12] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *ICLR*, 2018. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6, 8

[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. 6, 13, 15

[15] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pages 8340–8349, 2021. 6, 15

[16] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, pages 15262–15271, 2021. 6, 15

[17] Xueting Hu, Ce Zhang, Yi Zhang, Bowen Hai, Ke Yu, and Zhihai He. Learning to adapt clip for few-shot monocular depth estimation. In *WACV*, pages 5594–5603, 2024. 3

[18] Shiyuan Huang, Jiawei Ma, Guangxing Han, and Shih-Fu Chang. Task-adaptive negative envision for few-shot open-set recognition. In *CVPR*, pages 7171–7180, 2022. 3

[19] Zhenyu Huang, Mouxing Yang, Xinyan Xiao, Peng Hu, and Xi Peng. Noise-robust vision-language pre-training with positive-negative learning. *IEEE TPAMI*, pages 1–13, 2024. 3

[20] Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. Learning from complementary labels. In *NeurIPS*, volume 30, pages 5644–5654, 2017. 1, 3, 12, 14

[21] 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*, pages 4904–4916, 2021. 1, 2, 16

[22] Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *CVPR*, 2024. 3

[23] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *ICCV*, pages 101–110, 2019. 3

[24] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, pages 554–561, 2013. 6, 13, 15

[25] Xin Li, Dongze Lian, Zhihe Lu, Jiawang Bai, Zhibo Chen, and Xinchao Wang. Graphadapter: Tuning vision-language models with dual knowledge graph. In *NeurIPS*, volume 36, pages 13448–13466, 2023. 1, 6, 7, 8, 14

[26] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *ICLR*, 2022. 1

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 1

[28] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2016. 6

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[30] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 6, 13, 15

[31] 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 *ICCVW*, pages 262–271, 2023. 16

[32] Jun Nie, Yonggang Zhang, Zhen Fang, Tongliang Liu, Bo Han, and Xinmei Tian. Out-of-distribution detection with negative prompts. In *ICLR*, 2024. 3, 15

[33] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, pages 722–729. IEEE, 2008. 6, 13, 15

[34] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 14

[35] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505, 2012. 5, 6, 12, 13, 15

[36] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *ICCV*, pages 15691–15701, 2023. 15, 16

[37] Pengchong Qiao, Zhidan Wei, Yu Wang, Zhennan Wang, Guoli Song, Fan Xu, Xiangyang Ji, Chang Liu, and Jie Chen. Fuzzy positive learning for semi-supervised semantic segmentation. In *CVPR*, pages 15465–15474, 2023. 3

[38] 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*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 4, 5, 6, 7, 8, 13, 14, 15

[39] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2

[40] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, pages 5389–5400. PMLR, 2019. 6, 15

[41] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1

[42] 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. *IJCV*, 115:211–252, 2015. 1

[43] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019. 2

[44] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017. 2

[45] 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. *NeurIPS*, 35:14274–14289, 2022. 1, 6, 7

[46] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, volume 30, pages 4080–4090, 2017. 2

[47] Yisheng Song, Ting Wang, Puyu Cai, Subrota K Mondal, and Jyoti Prakash Sahoo. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, 55(13):1–40, 2023. 1

[48] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6, 13, 15

[49] Ximeng Sun, Ping Hu, and Kate Saenko. Dualcoop: Fast adaptation to multi-label recognition with limited annotations. *NeurIPS*, 35:30569–30582, 2022. 1

[50] Xinyu Tian, Shu Zou, Zhaoyuan Yang, and Jing Zhang. Argue: Attribute-guided prompt tuning for vision-language models. In *CVPR*, 2024. 1, 2

[51] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 16

[52] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. In *ICCV*, pages 2725–2736, 2023. 2

[53] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9:2579–2605, 2008. 5

[54] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *NeurIPS*, 29, 2016. 2

[55] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, volume 32, pages 10506–10518, 2019. 6, 15

[56] Hualiang Wang, Yi Li, Huifeng Yao, and Xiaomeng Li. Clipn for zero-shot ood detection: Teaching clip to say no. In *ICCV*, pages 1802–1812, 2023. 3, 15

[57] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *CVPR*, pages 4248–4257, 2022. 3

[58] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, 53(3):1–34, 2020. 1

[59] Zhengbo Wang, Jian Liang, Lijun Sheng, Ran He, Zilei Wang, and Tieniu Tan. A hard-to-beat baseline for training-free CLIP-based adaptation. In *ICLR*, 2024. 16

[60] Xiu-Shen Wei, H-Y Xu, Faen Zhang, Yuxin Peng, and Wei Zhou. An embarrassingly simple approach to semi-supervised few-shot learning. In *NeurIPS*, volume 35, pages 14489–14500, 2022. 3

[61] Cheng-En Wu, Yu Tian, Haichao Yu, Heng Wang, Pedro Morgado, Yu Hen Hu, and Linjie Yang. Why is prompt tuning for vision-language models robust to noisy labels? In *ICCV*, pages 15488–15497, 2023. 2

[62] Wenhao Wu, Huanjin Yao, Mengxi Zhang, Yuxin Song, Wanli Ouyang, and Jingdong Wang. Gpt4vis: What can gpt-4 do for zero-shot visual recognition? *arXiv preprint arXiv:2311.15732*, 2023. 16

[63] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492, 2010. 6, 13, 15

[64] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *TMLR*, 2022. 1, 16

[65] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *CVPR*, pages 10899–10909, 2023. 1, 2, 6, 7, 12, 13, 14

[66] Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *ECCV*, pages 68–83. Springer, 2018. 1, 3, 12, 14

[67] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *CVPR*, pages 18123–18133, 2022. 1

[68] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, pages 12203–12213, 2020. 2

[69] Ce Zhang, Simon Stepputtis, Katia Sycara, and Yaqi Xie. Dual prototype evolving for test-time generalization of vision-language models. *arXiv preprint arXiv:2410.12790*, 2024. 2

[70] Ce Zhang, Simon Stepputtis, Katia P. Sycara, and Yaqi Xie. Few-shot adaptation of vision-language foundation models via dual-path inference. In *ICLRW*, 2024. 1

[71] 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*, pages 493–510. Springer, 2022. 1, 2, 3, 5, 6, 7, 8, 13

[72] Yi Zhang, Ce Zhang, Ke Yu, Yushun Tang, and Zhihai He. Concept-guided prompt learning for generalization in vision-language models. In *AAAI*, volume 38, pages 7377–7386, 2024. 1

[73] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825, 2022. 1, 2, 3, 6, 7, 13

[74] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. 1, 2, 3, 6, 7, 8, 13, 14, 15

[75] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *ICCV*, pages 15659–15669, 2023. 6, 7, 8

[76] 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*, pages 2605–2615, 2023. 2, 15