Multi-Label Few-Shot Image Classification via Pairwise Feature Augmentation and Flexible Prompt Learning

Han Liu[1], Yuanyuan Wang[1], Xiaotong Zhang[1*], Feng Zhang[2], Wei Wang[3], Fenglong Ma[4], Hong Yu[1]

[1] Dalian University of Technology, Dalian, China
[2] Peking University, Beijing, China
[3] Shenzhen MSU-BIT University, Shenzhen, China
[4] The Pennsylvania State University, Pennsylvania, USA
liu.han.dut@gmail.com, wangy9yuan@mail.dlut.edu.cn, zxt.dut@hotmail.com, zfeng.maria@gmail.com, ehomewang@ieee.org, fenglong@psu.edu, hongyu@dlut.edu.cn

Abstract

Multi-label few-shot image classification is a crucial and challenging task due to limited annotated data and elusive category specificity. However, research on this topic is still in the rudimentary stage and few methods are available. Existing methods either leverage data augmentation to alleviate data scarcity or utilize label features as auxiliary knowledge to eliminate the negative effect caused by irrelevant categories, but they ignore the utilization of image region features for data augmentation, and overlook to learn appropriate text feature to better match the image features of specific categories. Moreover, these methods only focus on one side and do not effectively tackle the above two issues simultaneously. In this paper, we introduce a novel prototype-based multi-label few-shot learning framework that seamlessly integrates pairwise feature augmentation and flexible prompt learning. Specifically, by pairwise feature augmentation, we leverage the region features of images in the support set to generate more image features and construct image prototypes, thus alleviating the issue of data scarcity. By flexible prompt learning, we adaptively acquire class-specific prompts to build text prototypes that highly match the image features of specific classes, thereby mitigating the impact of irrelevant classes. Finally, with adaptive learnable parameters, we merge image and text prototypes to obtain the final prototypes, achieving a more powerful classifier for multi-label few-shot image classification. Extensive experimental results demonstrate that our proposed method can push the performance to a higher level.

Introduction

Accurately classifying real-world images with multiple category labels is a fundamental yet prominent problem in the fields of computer vision and multimedia, particularly when compared with single-label image classification (Xu et al. 2022; Li, Zhu, and Wang 2023). Furthermore, in general settings, training an image classifier capable of making accurate predictions requires large amounts of annotated data, which is often impractical for real-world applications. Therefore, researchers have turned their attention to few-shot image classification (Kang et al. 2021; Hiller et al. 2022) to address the problem of data scarcity. In order to simultaneously address the challenges of multiple category labels within an image and limited data availability in practical scenarios, a new research task has emerged, known as multi-label few-shot image classification, aiming to classify multiple class labels in an image with few training samples (Alfassy et al. 2019). However, this task is particularly challenging due to two main reasons. Firstly, there are few annotated data, leading to data scarcity. Secondly, in contrast to single-label image classification, multi-label image classification is more complicated, which is prone to causing the interference from irrelevant classes.

Various approaches have been proposed to address the challenges of data scarcity and mitigate the impact of irrelevant classes in multi-label few-shot image classification. One common strategy involves the utilization of ***data augmentation based techniques*** to tackle the data scarcity issue. For example, Alfassy et al. (2019) introduce a noteworthy approach in this realm, employing label set operation networks to obtain feature vectors corresponding to the union, intersection and subtraction label sets of image pairs. Although existing methods improve the model performance through data augmentation, they only utilize the global features of the images, without fully exploiting the informative region features, limiting the diversity of generated data. Previous works also adopt ***label feature based strategies*** to mitigate the impact of irrelevant classes. Yan et al. (2022) utilize word embeddings as label knowledge and employ an attention mechanism dependent on label vectors to aggregate region features of support images. Sun, Hu, and Saenko (2022) introduce a dual context optimization method based on CLIP (Radford et al. 2021), which encodes positive and negative contexts by incorporating class names as part of the prompts. Li et al. (2024) sequentially learn class-shared positive and negative text prompts to classify images. These methods can partially alleviate the impact of irrelevant classes, but they do not consider how to obtain more class-discriminative text prototypes which highly match with the image features of a specific class, thus affecting their effectiveness to some extent.

Although existing approaches have demonstrated impressive performance in multi-label image classification, they

*Corresponding author.

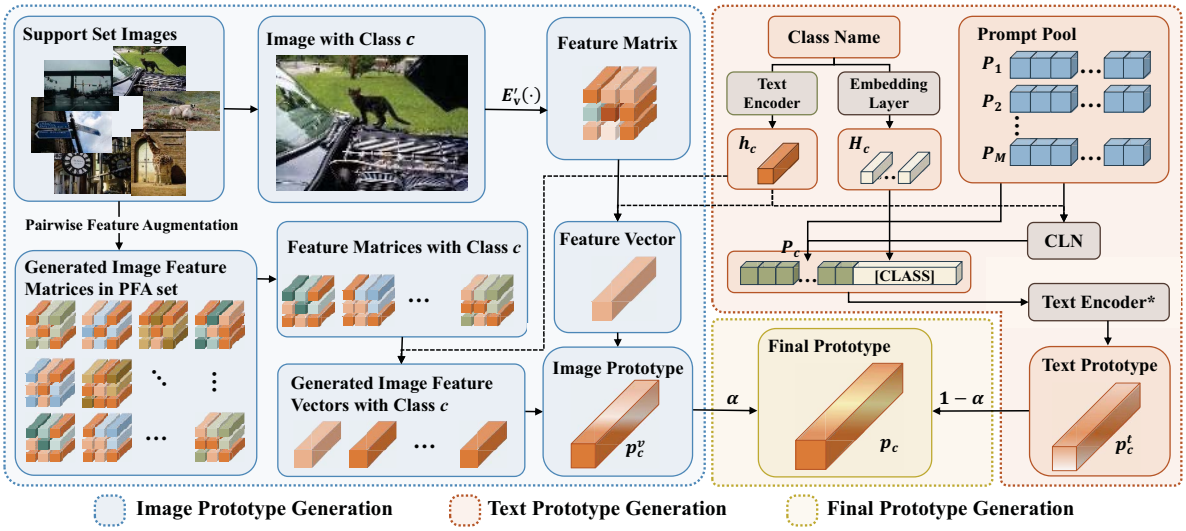

Figure 1: Our method consists of three main components: image, text, and final prototype generation. For image prototype generation, we employ pairwise feature augmentation based on image region features to enrich the support set, and combine the feature vectors from the support set and the PFA set to generate the image prototypes for specific classes. For text prototype generation, we introduce flexible prompt learning to obtain flexible prompts for different classes, and combine the prompts with label words to obtain text prototypes that better match the image features of the specific classes. Finally, we use learnable parameters to weight the image and text prototypes, thus generating the final prototypes for image classification.

usually concentrate solely on addressing either the data scarcity or the class irrelevance issue. However, the solutions to these two problems are complementary, which indicates that solving theses two problems simultaneously seems potential to boost the performance significantly. In this paper, we propose a novel adaptive prototype construction method for multi-label image classification by leveraging pairwise image feature augmentation and flexible prompt learning. As shown in Figure 1, to alleviate the issue of data scarcity, we conduct Pairwise Feature Augmentation (PFA) based on image region features of the support set, aiming to obtain more diverse class features for constructing image prototypes. To alleviate the interference of irrelevant label information, we adopt Flexible Prompt Learning (FPL) to dynamically obtain class-specific prompts and construct more class-discriminative text prototypes that can be highly matched with the image features of specific class. Finally, by combining the image and text prototypes with learnable parameters, we adaptively obtain the final prototypes, achieving a more powerful classifier for multi-label few-shot image classification. Experimental results demonstrate that our method can achieve state-of-the-art performance compared with other strong baselines.

Related Work

Multi-label few-shot image classification is a challenging task due to data scarcity and the difficulty in extracting class-specific features. Currently, there has been some progress in both few-shot image classification (Zhang et al. 2022; Zhou et al. 2022) and multi-label image classification (Lanchantin et al. 2021). Existing methods primarily utilize two strategies to address multi-label few-shot image classification. One is to use data augmentation to alleviate data scarcity, and the other is to use label features to mitigate irrelevant categories.

Data Augmentation Based Methods. Data augmentation is a technique that increases data diversity by transforming and amplifying existing data (Ni et al. 2021; Perez and Wang 2017; Liang, Liang, and Jia 2023). In the field of multi-label few-shot image classification, LaSO (Alfassy et al. 2019) is the only model that addresses this issue by data augmentation. It inputs the feature vectors of image pairs into three label set operation networks respectively to synthesize feature vectors corresponding to image union, intersection and subtraction labels. However, LaSO focuses on extracting the global feature vector of the image for label set operations and does not utilize the information of regional features.

Label Feature Based Methods. Considering the interference of irrelevant classes in multi-label images, some studies utilize label features to assist classifiers in capturing class-specific features. Yan et al. (2022) map text and image embeddings to the shared feature space and introduce an attention mechanism based on label vectors to aggregate region features of support images. Song, Wang, and Zhong (2024) propose a self-prompt method that adaptively adjusts neural networks using intrinsic semantic features. With the development of pre-trained visual-language models, images and label text can be mapped to the same feature space through their text and visual encoders, making it more convenient to utilize label information and obtain class-specific feature. Based on CLIP, Sun, Hu, and Saenko (2022) introduce the Dual Context Optimization (DualCoOp) method, which utilizes the embedding of class names as part of the prompts to encode positive and negative contexts for image classifica-

Symbol	Explanation
y_A	The class label of image A.
$\boldsymbol{X}_A$	The feature matrix of image A.
$\boldsymbol{X}'_{A,B}$	The concatenated feature matrix of images A and B.
$\boldsymbol{R}_{A\cup B}$	The feature matrix generated by PFA.
$\boldsymbol{r}^i_{A\cup B}$	The feature vector of region i in $\boldsymbol{R}_{A\cup B}$.
$\boldsymbol{x}^c_i$	The feature vector of $\boldsymbol{X}_i$ for class c.
$\boldsymbol{r}^c_i$	The feature vector of $\boldsymbol{R}_i$ for class c.
$\boldsymbol{P}_i$	The i-th prompt in the prompt pool.
$\boldsymbol{P}^c$	The specific prompt generated for class c.
$\boldsymbol{p}^v_c$	The image prototype for class c.
$\boldsymbol{p}^t_c$	The text prototype for class c.
$\boldsymbol{p}_c$	The final prototype for class c.

Table 1: Symbol explanation.

tion. Li et al. (2024) propose a lightweight method that sequentially learns class-shared positive and negative prompts based on class label text. However, no work has attempted to adaptively learn a prompt based on the class label that can closely match the image features of a specific class and eliminate the influence of irrelevant categories.

Problem Definition

We employ the meta-learning mechanism for multi-label few-shot image classification. In the multi-label setting, a sample may be assigned to multiple classes, which makes the construction of an episode different. In our study, we follow (Yan et al. 2022) to create episodes. In the N-way K-shot setting, each episode comprises a support set $\mathcal{S} = (x_i, y_i)_{i=1}^{N\times K}$ and a query set $\mathcal{Q} = (x_j, y_j)_{j=1}^{N\times q}$. For the support set, we sample K images without replacement from the training dataset D_{train} for each of the N class labels in $\mathcal{C}_{\text{train}}$ (the train label set), resulting in a total of $N \times K$ images. Since each image may have multiple labels, the number of images corresponding to each label in the support set might exceed K. The query set is constructed similarly, we sample $N \times q$ images without replacement. The test episodes are constructed in the same way from the test dataset D_{test} and its label set $\mathcal{C}_{\text{test}}$, where $\mathcal{C}_{\text{train}} \cap \mathcal{C}_{\text{test}} = \varnothing$. Table 1 summarizes some symbol explanation in details.

The Proposed Method

As shown in Figure 1, we propose a method to simultaneously utilize the image information and class label information in the support set for multi-label few-shot image classification.

Image Prototype Generation

As shown in Figure 1, we construct image prototypes in two steps: (1) We design a method called Pairwise Feature Augmentation (PFA) to generate more image features from the support set and alleviate the data scarcity problem. (2) We utilize the generated image features in conjunction with the support set images to construct image prototypes.

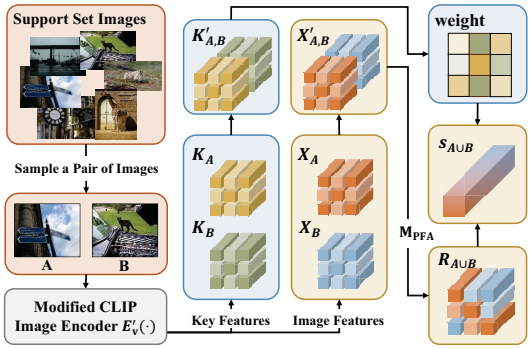

Figure 2: The illustration of pairwise feature augmentation.

In our approach, we utilize the visual encoder of CLIP to extract image features. In the original attention pooling layer of CLIP (Radford et al. 2021), attention pooling is first performed on visual features, and then the global feature vector is projected into the common feature space of image and text, as shown below:

$$\begin{aligned}&\text{AttnPool}(\boldsymbol{V}_A)\\ =&\text{Proj}\left(\sum_{i=1}^{n}\text{softmax}\left(\frac{q(\bar{\boldsymbol{v}}_A)^T k(\boldsymbol{v}_A^i)}{C}\right)\cdot v(\boldsymbol{v}_A^i)\right)\\ =&\sum_{i=1}^{n}\text{softmax}\left(\frac{q(\bar{\boldsymbol{v}}_A)^T k(\boldsymbol{v}_A^i)}{C}\right)\cdot\text{Proj}(v(\boldsymbol{v}_A^i)),\end{aligned}\quad(1)$$

where $\boldsymbol{V}_A \in \mathbb{R}^{n\times d_v}$ is the feature matrix of image A output by the CLIP visual encoder before the attention pooling layer is applied, $q(\cdot)$, $v(\cdot)$ and $k(\cdot)$ are independent linear embedding layers, $\text{Proj}(\cdot)$ is an independent linear embedding layer that projects the visual feature vector into the common feature space of image and text, $\bar{\boldsymbol{v}}_A$ is the average of the region feature vectors, $\boldsymbol{v}_A^i$ is the feature vector at region i in $\boldsymbol{V}_A$, n is the number of region vectors, and d_v is the dimension of each vector in $\boldsymbol{V}_A$.

In multi-label image classification, it is common for objects of different categories to appear in different regions of the image. However, extracting image features using CLIP's original visual encoder will mix features from different regions into one feature vector, resulting in the loss of region-specific information and class features. Therefore, we reconstruct the last attention pooling layer of the CLIP visual encoder. For image A, we drop the pooling operation and project the visual features $\boldsymbol{v}_A^i$ of each region i into the common feature space of image and text:

$$\boldsymbol{X}_A = E'_v(x_A) = [\text{Proj}(v(\boldsymbol{v}_A^i))]_{i=1}^{n} \in \mathbb{R}^{n\times d},\quad(2)$$

where $E'_v(\cdot)$ is the modified CLIP visual encoder, d is the dimension of each vector in $\boldsymbol{X}_A$.

Pairwise Feature Augmentation To alleviate the data scarcity problem, we perform pairwise feature augmentation on the support set. As shown in Figure 2, by sampling images from the support set containing u images, we will obtain $u(u-1)/2$ pairs of images along with their corresponding labels. For each pair of images A and B, we obtain their

image feature matrices $\boldsymbol{X}_A = E'_v(x_A)$ and $\boldsymbol{X}_B = E'_v(x_B)$, along with their respective labels y_A and y_B. Then we can obtain $\boldsymbol{X}'_{A,B} = [\boldsymbol{X}_A; \boldsymbol{X}_B] \in \mathbb{R}^{n\times 2d}$, where ; denotes concatenation. Then, we feed $\boldsymbol{X}'_{A,B}$ into a Multi-Layer Perceptron (MLP) to generate the feature matrix $\boldsymbol{R}_{A\cup B}$:

$$\boldsymbol{R}_{A\cup B} = \mathrm{M}_{\mathrm{PFA}}(\boldsymbol{X}'_{A,B}) \in \mathbb{R}^{n\times d}, \tag{3}$$

where $\mathrm{M}_{\mathrm{PFA}}$ is a linear model designed for image pairwise feature augmentation, and the corresponding labels of $\boldsymbol{R}_{A\cup B}$ is $y_{A\cup B} = y_A \cup y_B$. Then, we perform PFA on each pair of images in the support set, and all generated image features along with their corresponding labels $(\boldsymbol{R}_j, y_j)_{j=1}^{u(u-1)/2}$ collectively constitute the **PFA set** $\mathcal{R}$.

As shown in Eq. (1), for image A, the key $\boldsymbol{K}_A = k(\boldsymbol{V}_A) \in \mathbb{R}^{n\times d_k}$ can be considered as descriptors for the corresponding image blocks (Zhou, Loy, and Dai 2022), where d_k is the dimension of the key in each region. The smaller the differences between the image feature vectors, the smaller the differences between the keys. Therefore, if the differences between the key $\boldsymbol{k}^i_A = k(\boldsymbol{v}_i) \in \mathbb{R}^{d_k}$ at region i and the keys at other regions are small, it indicates that the feature vector at region i may have more global features. Based on this, we use the keys to calculate the feature vector corresponding to the generated feature matrix $\boldsymbol{R}_{A\cup B}$ in the PFA set $\mathcal{R}$:

$$\boldsymbol{s}_{A\cup B} = \sum_{i=1}^{n} \left[\left(\boldsymbol{k}^i_{A,B}{}' \right)^T \bar{\boldsymbol{k}}'_{A,B} \right] \boldsymbol{r}^i_{A\cup B} \in \mathbb{R}^d, \tag{4}$$

where $\boldsymbol{k}^i_{A,B}{}' = [\boldsymbol{k}^i_A; \boldsymbol{k}^i_B] \in \mathbb{R}^{2d_k}$ is a vector formed by concatenating the keys of each pair of images A and B at region i, $\bar{\boldsymbol{k}}'_{A,B}$ is the average of $\boldsymbol{k}^i_{A,B}{}'$ over n regions, $\boldsymbol{r}^i_{A\cup B} \in \mathbb{R}^d$ is the feature vector at region i in $\boldsymbol{R}_{A\cup B}$.

We utilize keys to weight all feature matrices in $\mathcal{R}$, and the weighted feature vectors along with their corresponding labels $(\boldsymbol{s}_j, y_j)_{j=1}^{u(u-1)/2}$ collectively constitute the **PFA vector set** $\mathcal{R}^s$. In our approach, the classifier and the PFA model are trained synchronously. We classify the image feature vectors in $\mathcal{R}^s$, optimize the classifier, and restrict pairwise feature augmentation based on the classification results.

In addition, during the augmentation, we introduce the mean square error (MSE) $\mathcal{L}_{sym}$ to ensure that the generated features is independent of the order of the input images:

$$\mathcal{L}_{sym} = \mathrm{MSE}(\boldsymbol{R}_{A\cup B}, \boldsymbol{R}_{B\cup A}). \tag{5}$$

Image Prototype Learning When constructing image prototypes, we simultaneously utilize the feature matrices obtained from the support set images and the feature matrices with region-specific image features from the PFA set $\mathcal{R}$. This approach allows us to fully leverage the information present in the images of the support set.

To obtain class-specific image feature vectors due to the objects of multiple categories within an image, we design a vector weighting method based on the label vector feature $\boldsymbol{h}_c \in \mathbb{R}^d$ for class c. For the image i in the support set, we use the modified CLIP visual encoder $E'_v(\cdot)$ to obtain their image feature matrix $\boldsymbol{X}_i = E'_v(x_i)$, with each row representing the vector of each region. We compute the dot product between $\boldsymbol{X}_i$ and the label feature $\boldsymbol{h}_c$ of class c, followed by a softmax operation to obtain the weights for the vectors in $\boldsymbol{X}_i$, then use the weights to compute the feature vector of $\boldsymbol{X}_i$ relevant to the c-th class: $\boldsymbol{x}^c_i = (\boldsymbol{X}_i)^T(\mathrm{softmax}(\boldsymbol{X}_i \cdot \boldsymbol{h}_c)) \in \mathbb{R}^d$, where $\cdot$ represents the dot product operation between a matrix and a vector.

Similarly, for the augmented image features in the PFA set $\mathcal{R}$, we compute the dot product of the feature matrix $\boldsymbol{R}_j$ with the label feature $\boldsymbol{h}_c$, followed by a softmax to calculate the weights for each vector $\boldsymbol{r}^i_j$ in $\boldsymbol{R}_j$. With these weights, we can compute the feature vector of $\boldsymbol{R}_j$ relevant to the c-th class: $\boldsymbol{r}^c_j = (\boldsymbol{R}_j)^T(\mathrm{softmax}(\boldsymbol{R}_j \cdot \boldsymbol{h}_c)) \in \mathbb{R}^d$.

Then, as shown in Figure 1, combining the computed feature vectors from the support set and the PFA set, we calculate the image prototype for class c:

$$\boldsymbol{p}^v_c = \frac{1}{|\mathcal{X}_c| + |\mathcal{R}_c|} \left(\sum_{\boldsymbol{x}^c_i \in \mathcal{X}_c} \boldsymbol{x}^c_i + \sum_{\boldsymbol{r}^c_j \in \mathcal{R}_c} \boldsymbol{r}^c_j \right), \tag{6}$$

where $\mathcal{X}_c$ represents the set of weighted feature vectors of images belonging to class c in the support set, and $\mathcal{R}_c$ represents the set of weighted feature vectors of image features in the PFA set belonging to class c.

Text Prototype Generation

In the multi-label setting, the single utilization of image prototypes may introduce noise from other irrelevant classes. Therefore, we introduce textual labels for specific classes to obtain text prototypes which do not contain irrelevant class features. As shown in Figure 1, we extract text prototypes in two steps: (1) We design a method called Flexible Prompt Learning (FPL), utilizing a prompt pool and conditional layer normalization to adaptively obtain prompts for a specific class. (2) We construct text prototypes by combining class-specific prompts and class labels.

Flexible Prompt Learning Since the texts used for training CLIP are mostly sentences or paragraphs rather than label words, it is difficult to match the image features with the single usage of label embeddings as text prototypes. Therefore, during the training process, we infer flexible prompts for different classes from the prompt pool (Wang et al. 2022; Jia et al. 2022), and combine the prompts with label words to obtain text prototypes that better match the image features.

Prompt Pool. Given a prompt pool $\boldsymbol{P} = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3, \ldots, \boldsymbol{P}_M\}$, where $\boldsymbol{P}_i = \{\boldsymbol{p}^i_1, \boldsymbol{p}^i_2, \ldots, \boldsymbol{p}^i_T\} \in \mathbb{R}^{T\times d_e}$, M is the number of prompts in the pool, T is the number of learnable tokens in a prompt, d_e is the dimension of word embedding. Each prompt $\boldsymbol{P}_i$ is designed to capture diverse shared meta-knowledge.

Conditional Layer Normalization. Theoretically, there should exhibit interactive relationships between the learnable prompts and the classes. Inspired by (Li et al. 2022), we introduce Conditional Layer Normalization to generate the most effective prompt for a given class c, and the weights of the prompts in the prompt pool can be learned by:

$$w^c_i = \boldsymbol{\gamma}^c_i \left(\frac{\bar{\boldsymbol{p}}_i - \boldsymbol{\mu}}{\boldsymbol{\epsilon}} \right) + \boldsymbol{\lambda}^c_i, \tag{7}$$

where $\bar{\boldsymbol{p}}_i = \frac{1}{T}\sum_{j=1}^{T} \boldsymbol{p}_j^i \in \mathbb{R}^{d_e}$. The label feature $\boldsymbol{h}_c$ of class c is used as the condition for the gain parameters $\boldsymbol{\gamma}_i^c = \boldsymbol{W}_\alpha \boldsymbol{h}_c + \boldsymbol{b}_\alpha$ and biases $\boldsymbol{\lambda}_i^c = \boldsymbol{w}_\beta \boldsymbol{h}_c + \boldsymbol{b}_\beta$ in layer normalization. Here, $\boldsymbol{\mu}$ and $\boldsymbol{\epsilon}$ denote the mean and standard deviation of each element in $\bar{\boldsymbol{p}}_i$.

By applying softmax and utilizing the weights, we combine prompts from the pool to generate the prompt for class c:

$$\boldsymbol{P}^c = \sum_{i=1}^{M} w_i^c \boldsymbol{P}_i. \tag{8}$$

In addition, to prevent the prompts in the prompt pool from learning highly similar content, we introduce a $\mathcal{L}_{weight}$ to calculate the sum of entropy of each class's weight distribution, encouraging the learning of more diverse prompts by promoting non-uniform probability distributions across different classes:

$$\mathcal{L}_{weight} = -\sum_{c\in\mathcal{C}}\sum_{i=1}^{M} w_i^c \log w_i^c. \tag{9}$$

Text Prototype Learning For class c, we could obtain the feature matrix $\boldsymbol{H}_c \in \mathbb{R}^{n_c \times d_e}$, where n_c is the number of tokens of the class c. With $\boldsymbol{H}_c$ and the weighted prompt $\boldsymbol{P}^c$, we can obtain the textual embedding of class c as follows: $\boldsymbol{t}_c = [\boldsymbol{P}^c; \boldsymbol{H}_c] \in \mathbb{R}^{(T+n_c)\times d_e}$. Then we can obtain the text prototype for class c:

$$\boldsymbol{p}_c^t = E_t(\boldsymbol{t}_c) \in \mathbb{R}^d, \tag{10}$$

where $E_t(\cdot)$ is the text encoder of CLIP.

Adaptive Final Prototype Generation

After obtaining the image prototype $\boldsymbol{p}_c^v$ and the text prototype $\boldsymbol{p}_c^t$ of class c, we design learnable parameters to combine these two prototypes and obtain the final prototype:

$$\boldsymbol{p}_c = \alpha \times \boldsymbol{p}_c^v + (1-\alpha) \times \boldsymbol{p}_c^t, \tag{11}$$

where α is calculated using two learnable parameters $\alpha = \frac{\exp(\alpha_1)}{\exp(\alpha_2)+\exp(\alpha_1)}$, and α_1 and α_2 are both initialized to 1. The α allows our model to adaptively weight text and image prototypes without manually adjusting the weights.

The Loss Function

The Loss of Image Feature Classification For a given image (x_i, y_i) in the query set $\mathcal{Q}$, we can obtain the feature vector $\boldsymbol{s}_i = E_v(x_i)$, where $E_v(\cdot)$ is the original visual encoder of CLIP. Then, using the image feature vectors obtained from $\mathcal{Q}$ and their corresponding labels $(\boldsymbol{s}_i, y_i)_{i=1}^{N\times q}$, we can construct the feature vector set $\mathcal{Q}^s$.

For $(\boldsymbol{s}_i, y_i) \in \mathcal{Q}^s \cup \mathcal{R}^s$, the conditional probability belonging to class c is calculated as follows:

$$p(y=c|\boldsymbol{s}_i, \mathcal{S}) = \frac{\exp(-d(\boldsymbol{s}_i, \boldsymbol{p}_c))}{\sum_{c'\in\mathcal{C}} \exp(-d(\boldsymbol{s}_i, \boldsymbol{p}_{c'}))}, \tag{12}$$

where $d(\boldsymbol{s}_i, \boldsymbol{p}_c) = \|\boldsymbol{s}_i - \boldsymbol{p}_c\|_2^2$ represents the squared Euclidean distance between the sample and the prototype.

Then, we compute the cross-entropy loss for all samples $(\boldsymbol{s}_i, y_i) \in \mathcal{Q}^s \cup \mathcal{R}^s$ as follows:

$$\mathcal{L}_{ce} = \frac{-\sum\limits_{\boldsymbol{s}_i\in\mathcal{Q}^s\cup\mathcal{R}^s}\sum\limits_{c=1}^{|\mathcal{C}|} y_i^c \log p(y=c|\boldsymbol{s}_i, \mathcal{S})}{|\mathcal{Q}^s| + |\mathcal{R}^s|}, \tag{13}$$

where $\mathcal{C}$ represents the set of classes, and $y_i^c \in \{0, 1\}$.

The Loss of Label Count Predictor Since an image may have multiple labels in a multi-label setting, we employ an MLP to directly predict the number of labels (Liu et al. 2022), and optimize the label count predictor via the cross-entropy loss:

$$\mathcal{L}_{count} = \sum_{x\in\mathcal{S}\cup\mathcal{Q}} \text{CE}(\text{MLP}(E_v(x)), g_x), \tag{14}$$

where $\text{CE}(\cdot)$ represents the cross-entropy loss function, and g_x is the true label count of x. Since the data in $\mathcal{R}^s$ is generated through data augmentation, it may introduce potential label noise. Therefore, we only use samples of $\mathcal{S}\cup\mathcal{Q}$ to optimize the label count predictor.

The Final Loss When calculating the final loss, we use uncertainty (Kendall, Gal, and Cipolla 2018) to weight the $\mathcal{L}_{\text{sym}}$ in Eq. (5), $\mathcal{L}_{\text{weight}}$ in Eq. (9), $\mathcal{L}_{\text{ce}}$ in Eq. (13) and $\mathcal{L}_{\text{count}}$ in Eq. (14), eliminating the need for manual weight tuning:

$$\begin{aligned}\mathcal{L} =& \frac{1}{2{\sigma_1}^2}\mathcal{L}_{\text{sym}} + \frac{1}{2{\sigma_2}^2}\mathcal{L}_{\text{weight}} + \frac{1}{2{\sigma_3}^2}\mathcal{L}_{\text{ce}} \\ &+ \frac{1}{2{\sigma_4}^2}\mathcal{L}_{\text{count}} + \sum_{i=1}^{4}\log\sigma_i,\end{aligned} \tag{15}$$

where σ_i represents the learnable uncertainty parameter initialized to 1. Here, we introduce the uncertainty parameter σ_i to automatically adjust the loss weight and reduce the impact of high uncertainty loss, and the regularization term $\sum_{i=1}^{4}\log\sigma_i$ is introduced to prevent σ_i from becoming excessively large.

Experiments

Datasets

We evaluate our method on two datasets, namely MS COCO (Lin et al. 2014) and PASCAL VOC (Everingham et al. 2015), following the dataset split approach outlined in (Yan et al. 2022). For the COCO dataset, which comprises 80 classes, we divide it into training/validation/test sets with a ratio of 52/12/16, respectively. Similarly, For the VOC dataset, which consists of 20 classes, we split it into training/validation/test sets with a ratio of 8/6/6, respectively.

Baselines

We compare our model with the following strong baselines:

- WVAtten (Yan et al. 2022) utilizes GloVe word vectors (Pennington, Socher, and Manning 2014) as prior knowledge for labels and aggregates region features of images with an attention mechanism. Since we use the same experimental setting, the results are directly taken from (Yan et al. 2022).

Dataset	Method	Micro				Macro			
		Precision	**Recall**	**F1**	**AP**	**Precision**	**Recall**	**F1**	**AP**
COCO	WVAtten (2022)	49.72	26.60	34.21	35.30	34.50	25.07	28.91	42.84
	WVAtten* (2022)	53.35	37.18	43.46	46.03	45.48	35.77	35.43	56.12
	DualCoop† (2022)	50.16	67.56	57.06	62.58	55.33	68.47	56.28	68.58
	SPM† (2024)	53.53	32.54	40.18	57.57	47.32	32.97	35.63	60.49
	NegP† (2024)	74.70	53.11	62.06	67.35	68.40	53.04	56.31	73.31
	Ours	**77.41±1.3**	**68.46±0.9**	**68.44±1.1**	**79.36±1.7**	**73.80±2.4**	**68.81±1.2**	**64.82±1.1**	**82.85±1.3**
VOC	WVAtten (2022)	26.78	83.97	40.19	46.28	29.64	85.44	43.35	53.26
	WVAtten* (2022)	57.38	55.86	56.41	62.57	57.25	56.51	53.11	70.12
	DualCoop† (2022)	57.98	86.06	68.32	83.05	58.56	85.29	67.09	86.45
	SPM† (2024)	56.08	32.98	41.04	70.38	53.10	33.15	37.86	75.59
	NegP† (2024)	70.29	80.62	73.57	86.97	74.81	81.29	74.04	89.82
	Ours	**73.33±1.2**	**86.35±1.0**	**77.03±1.1**	**90.95±1.8**	**76.98±1.2**	**86.47±0.8**	**77.77±1.5**	**93.50±0.7**

Table 2: Experimental results on COCO and VOC datasets.

- WVAtten*(Yan et al. 2022) adopts the approach of WVAtten, where we substitute the visual encoder's initial weights with the pre-trained weights of CLIP.
- DualCoop† (Sun, Hu, and Saenko 2022) learns positive and negative prompts for partial-label classification, which is modified for multi-label few-shot image classification by learning a pair of shared prompts.
- SPM† (Song, Wang, and Zhong 2024) is a self-prompt method for few-shot image recognition that adaptively adjusts neural networks using intrinsic semantic features.
- NegP† (Li et al. 2024) is a lightweight OOD detection method that is reformulated as a multi-label few-shot image classification method by sequentially learning class-shared positive and negative prompts.

Implementation Details

Evaluation Metrics. We follow (Yan et al. 2022) to report precision (P), recall (R), F1 score (F1) and average precision (AP) from both micro and macro perspectives.

Parameter Settings. Following the settings of (Yan et al. 2022), during the construction of episodes, we set $K = 1$ and $q = 4$. We use the visual and text encoders with frozen parameters from CLIP, where the visual encoder is ResNet50 (He et al. 2016) and the text encoder is Transformer (Vaswani et al. 2017). In the prompt pool, we set the number of prompts M to 8, and the token number for each prompt T to 16. During the training process, we optimize the model parameters using the SGD optimizer with a learning rate of 0.02. For these parameters, we use the grid searching strategy and validation set to determine them. The reported results are averaged over the experiments using 5 different seeds, and in each run the results are averaged over 200 test episodes.

Result Analysis

Table 2 presents the experimental results on COCO and VOC datasets under the setting of (Yan et al. 2022). The top results are highlighted in bold. As seen in Table 2, our method achieves state-of-the-art performance on COCO and VOC datasets compared with other strong baselines including WVAtten, WVAtten*, SPM†, DualCoop† and NegP†. Compared with the strongest baseline NegP†, on the COCO dataset, our method achieves improvements of 2.71%, 15.35%, 6.38%, 12.01%, 5.40%, 15.77%, 8.51%, and 9.54% in micro P, R, F1, AP and macro P, R, F1, AP, respectively. On the VOC dataset, our method improves by 3.04%, 5.73%, 3.46%, 3.98%, 2.17%, 5.18%, 3.73%, and 3.68% in micro P, R, F1, AP and macro P, R, F1, AP, respectively. The results demonstrate that our model can capture the complex relationships between images and classes, highlighting the effectiveness of our model in multi-label few-shot image classification.

Ablation Study

Ablation Study of Pairwise Feature Augmentation. To investigate the impact of pairwise feature augmentation, we compare the experimental results with and without this augmentation technique, denoted as Ours and w/o PFA, respectively. As shown in Table 3, removing pairwise feature augmentation leads to a decrease of 4.34% in miAP (micro AP) and 4.01% in maAP (macro AP) on the COCO dataset, and a decrease of 1.61% in miAP and 1.40% in maAP on the VOC dataset. This indicates that pairwise feature augmentation contributes to improving the classification performance of the model.

Ablation Study of Flexible Prompt Learning. To investigate the impact of flexible prompt learning, we compare experimental results using flexible prompt learning against those using only class text features as text prototypes, denoted as Ours and w/o FPL, respectively. As shown in Table 3, removing flexible prompt learning results in a decrease of 29.04% in miAP and 2.35% in maAP on the COCO dataset, and a decrease of 10.08% in miAP and 1.17% in maAP on the VOC dataset. It is evident that removing flexible prompt learning leads to reductions in both miAP and maAP, with a

(a) w/o PFA (b) w/o FPL (c) w/o Image Prototype (d) w/o Text Prototype (e) Ours

Figure 3: Visualization of prototype embeddings. Each color represents a class in the test set of COCO dataset.

Method	COCO		VOC	
	miAP	**maAP**	**miAP**	**maAP**
Ours	**79.36**	**82.85**	**90.95**	**93.50**
w/o PFA	75.02	78.84	89.34	92.10
w/o FPL	50.32	80.50	80.87	92.33
w/o Image Prototype	78.30	82.14	87.30	93.08
w/o Text Prototype	49.26	60.01	70.26	78.97
w/o Learnable Alpha	76.38	80.80	87.19	90.77
w/o Uncertain Weight	76.94	81.93	88.91	93.24

Table 3: Ablation study and learnable parameter importance.

more pronounced decrease in miAP. This suggests that omitting prompts impairs classification performance, especially in certain categories.

Ablation Study of Text Prototypes and Image Prototypes. To clarify the contribution of text and image prototypes in our method, we perform ablation experiments by excluding these two types of prototypes respectively when calculating the final prototypes in Eq. (11). As shown in Table 3, when relying solely on image prototypes (w/o Text Prototype), the model's classification performance is not ideal. In contrast, the model achieves better classification results when using only text prototypes (w/o Image Prototype, still utilizing PFA). When using both image and text prototypes (Ours), the model achieves the best performance. It can be seen that the joint utilization of both image and text prototypes enables the acquisition of a superior multi-label few-shot image classifier.

Visualization

To better observe how the prototype embeddings change with the pairwise feature augmentation, flexible prompt learning, image prototypes, and text prototypes, we sample 1000 episodes from the test set of COCO in 16-way-1-shot setting and then use t-SNE (van der Maaten and Hinton 2008) to visualize the prototype embeddings obtained from w/o PFA, w/o FPL, w/o Image Prototype, w/o Text Prototype, and Ours. As shown in Figure 3, the application of pairwise feature augmentation results in clearer boundaries among the distribution regions of different category prototypes. Flexible prompt learning effectively prevents partial prototypes of certain classes from appearing in the regions of other classes, ensuring that prototypes of the same category cluster together. When using text prototypes alone, the category prototypes are well-separated from each other, reducing interference from irrelevant categories. Conversely, when text prototypes are not used, prototypes from different categories are mixed, and the distribution regions lack distinct boundaries. By employing our comprehensive method, prototypes of the same category are clustered together, resulting in the better prototype distribution.

Learnable Parameter Importance

In Eq. (11), we utilize an adaptive learnable parameter α to weight the image prototypes and text prototypes. Simultaneously, in Eq. (15), we employ uncertainty to weigh the losses when calculating the final loss. To elucidate the impact of these adaptive learnable parameters, we conduct the following experiments:

- **w/o Learnable Alpha**: Removing the learnable parameter α and averaging the text and image prototypes.
- **w/o Uncertain Weight**: Removing the uncertainty to weigh losses and averaging the losses.

As shown in Table 3, removing the learnable parameter α or Uncertainty Weight significantly reduces the model performance. It is evident that the weights of text and image prototypes as well as the weights of the losses are crucial parameters for our model. The two types of learnable parameters allow our model to learn adaptive and appropriate weights, leading to enhanced performance.

Conclusion

In this paper, we propose a multi-label few-shot image classification method based on pairwise feature augmentation and flexible prompt learning. To address data scarcity, we perform pairwise feature augmentation on the regional features of each image pair in the support set, and utilize both generated and original image features to construct image prototypes, thus maximizing the utilization of image features. To mitigate the impact of irrelevant classes, we introduce flexible prompt learning to dynamically obtain prompts based on the specific class labels to adaptively obtain text prototypes that match the image features of specific classes. Subsequently, adaptive parameters are utilized to meticulously weigh these prototypes, resulting in the refinement of final prototypes for each class. Extensive experiment results show that our model outperforms other strong baselines significantly. In future work, we plan to explore additional techniques for image feature augmentation and label text utilization to further enhance the performance of multi-label few-shot image classification.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62106035, 62206038), Liaoning Binhai Laboratory Project (No. LBLF-2023-01), and Chunhui Project Foundation of the Education Department of China (No. HZKY20220419). We also would like to thank Dalian Ascend AI Computing Center and Dalian Ascend AI Ecosystem Innovation Center for providing inclusive computing power and technical support.

References

Alfassy, A.; Karlinsky, L.; Aides, A.; Shtok, J.; Harary, S.; Feris, R. S.; Giryes, R.; and Bronstein, A. M. 2019. LaSO: Label-Set Operations Networks for Multi-Label Few-Shot Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 6548–6557.

Everingham, M.; Eslami, S. M. A.; Gool, L. V.; Williams, C. K. I.; Winn, J. M.; and Zisserman, A. 2015. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision (IJCV)*, 111(1): 98–136.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hiller, M.; Ma, R.; Harandi, M.; and Drummond, T. 2022. Rethinking Generalization in Few-Shot Classification. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 3582–3595.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual Prompt Tuning. In *European Conference on Computer Vision (ECCV)*, 709–727.

Kang, D.; Kwon, H.; Min, J.; and Cho, M. 2021. Relational Embedding for Few-Shot Classification. In *IEEE International Conference on Computer Vision (ICCV)*, 8822–8833.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 7482–7491.

Lanchantin, J.; Wang, T.; Ordonez, V.; and Qi, Y. 2021. General Multi-Label Image Classification With Transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 16478–16488.

Li, J.; Fei, H.; Liu, J.; Wu, S.; Zhang, M.; Teng, C.; Ji, D.; and Li, F. 2022. Unified Named Entity Recognition as Word-Word Relation Classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, 10965–10973.

Li, J.; Zhu, X.; and Wang, J. 2023. AdaBoost.C2: Boosting Classifiers Chains for Multi-Label Classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, 8580–8587.

Li, T.; Pang, G.; Bai, X.; Miao, W.; and Zheng, J. 2024. Learning Transferable Negative Prompts for Out-of-Distribution Detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 17584–17594.

Liang, W.; Liang, Y.; and Jia, J. 2023. MiAMix: Enhancing Image Classification through a Multi-Stage Augmented Mixed Sample Data Augmentation Method. *Processes*, 11(12).

Lin, T.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, 740–755.

Liu, H.; Zhang, F.; Zhang, X.; Zhao, S.; Sun, J.; Yu, H.; and Zhang, X. 2022. Label-enhanced Prototypical Network with Contrastive Learning for Multi-label Few-shot Aspect Category Detection. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 1079–1087.

Ni, R.; Goldblum, M.; Sharaf, A.; Kong, K.; and Goldstein, T. 2021. Data Augmentation for Meta-Learning. In *International Conference on Machine Learning (ICML)*, 8152–8161.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global Vectors for Word Representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Perez, L.; and Wang, J. 2017. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. arXiv:1712.04621.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 8748–8763.

Song, M.; Wang, H.; and Zhong, G. 2024. Self-Prompt Mechanism for Few-Shot Image Recognition. In *AAAI Conference on Artificial Intelligence (AAAI)*, 4934–4942.

Sun, X.; Hu, P.; and Saenko, K. 2022. DualCoOp: Fast Adaptation to Multi-Label Recognition with Limited Annotations. In *Conference on Neural Information Processing Systems (NeurIPS)*, 30569–30582.

van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86): 2579–2605.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Conference on Neural Information Processing Systems (NeurIPS)*, 5998–6008.

Wang, Z.; Zhang, Z.; Lee, C.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J. G.; and Pfister, T. 2022. Learning to Prompt for Continual Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 139–149.

Xu, J.; Huang, S.; Zhou, F.; Huangfu, L.; Zeng, D.; and Liu, B. 2022. Boosting Multi-Label Image Classification with Complementary Parallel Self-Distillation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1495–1501.

Yan, K.; Zhang, C.; Hou, J.; Wang, P.; Bouraoui, Z.; Jameel, S.; and Schockaert, S. 2022. Inferring Prototypes for Multi-Label Few-Shot Image Classification with Word Vector

Guided Attention. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2991–2999.

Zhang, R.; Zhang, W.; Fang, R.; Gao, P.; Li, K.; Dai, J.; Qiao, Y.; and Li, H. 2022. Tip-Adapter: Training-Free Adaption of CLIP for Few-Shot Classification. In *European Conference on Computer Vision (ECCV)*, 493–510.

Zhou, C.; Loy, C. C.; and Dai, B. 2022. Extract Free Dense Labels from CLIP. In *European Conference on Computer Vision (ECCV)*, 696–712.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022. Learning to Prompt for Vision-Language Models. *International Journal of Computer Vision (IJCV)*, 130(9): 2337–2348.