



On inferring prototypes for multi-label few-shot learning via partial aggregation

Pengfei Fang^{a,b,1}, Zhihong Chen^{a,b,1}, Hui Xue^{a,b,*}

^a School of Computer Science and Engineering, Southeast University, Nanjing, 210096, China

^b Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China

ARTICLE INFO

Keywords:

Multi-label few-shot learning
Partial aggregation scheme
Partial optimal transport

ABSTRACT

Multi-label few-shot learning (ML-FSL) aims to endow the learning system to recognize multiple objects within an image, trained with insufficient samples. Existing methods have significantly improved ML-FSL and focused on mining the correlation of labels, resulting in a discriminative prototype per class. However, those methods often engage irrelevant information, *i.e.*, the tangled region with other classes, in the phase of constructing prototypes, limiting their performance gain. Following the intuition that only part regions of an image correspond to a target label, this paper addresses this issue by creating prototypes via a partial aggregation scheme. This is realized by first generating aggregation weights via partial optimal transport (POT) between image and label features and producing features per class using relevant regions within an image. Having the refined class features in a support set, one can obtain a better prototype for each class. We evaluate our model on multiple benchmarks and obtain state-of-the-art performance. A thorough study also reveals the superiority of POT as a way of mining important information for generating prototypes.

1. Introduction

This paper investigates the multi-label few-shot learning (ML-FSL) problem by developing the partial aggregation scheme to generate a better class prototype, which is realized by partial optimal transport (POT) as a way of matching the distributions from the textual and visual modalities.

Few-shot learning (FSL) refers to a learning scenario where the machine is required to adapt to new tasks fast, trained with only a limited amount of data. In this regard, a diversity of methods, including the meta-learning [1,2], transfer learning [3,4], generative learning [5,6], *etc.*, have been proposed, and bring significant improvement to the generalization capacity of FSL. However, those pioneer works only focus on the recognition task of a single class, while neglecting the fact that the images in real-world scenarios often include multiple objects (see Fig. 1(a)) and recognizing all possible objects is also necessary. This is referred to as multi-label few-shot learning problems (ML-FSL) [7], and the main challenge of ML-FSL is to disentangle the mixed class² information present with the scarce data for discriminative class representations.

Only a few attempts have been made to address ML-FSL. Existing methods predominantly adopt episodic training paradigms and

generate prototypes for classification to tackle multi-label few-shot learning (ML-FSL) problems. Chen et al. [8] propose leveraging label co-occurrence by constructing a knowledge graph to generate class prototypes, incorporating neighboring labels in the graph. Simon et al. [9] extend the episodic training paradigm, also known as meta-learning, initially proposed for few-shot learning (FSL), to ML-FSL. Their approach generates class prototypes by averaging samples containing the target class. However, when a sample has multiple labels, the averaging operation results in equal contributions from each corresponding prototype, which can diminish the distinguishability of class representations. Yan et al. [10] incorporate word embeddings of class text as prior knowledge and employ a cross-attention mechanism to fuse text and image features, improving prototype generation. While this method differentiates the contributions of different regions of the sample to the corresponding categories, it does not account for noisy regions within the image. This oversight can severely degrade the quality of the generated prototypes, particularly when noisy regions interfere. In such cases, prototypes become less discriminative, reducing the model's ability to distinguish between similar classes. This issue is exacerbated in multi-label classification. Therefore, it is essential to mitigate the influence of such noise to improve the accuracy of prototype generation.

* Corresponding author.

E-mail addresses: fangpengfei@seu.edu.cn (P. Fang), 220232235@seu.edu.cn (Z. Chen), hxue@seu.edu.cn (H. Xue).

¹ Equal contributions.

² In the remainder of this paper, we will use the terms “label”, “class” and “category” interchangeably.

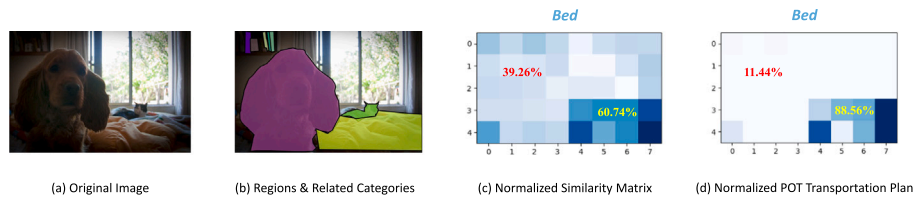


Fig. 1. An example to show local image features and their relationship with categories in a ML-FSL task. (a) shows an original image from the real world containing 3 concerned categories, *i.e.*, *dog*, *cat*, *bed*. (b) roughly shows the image regions of this image w.r.t. its corresponding class(es). The areas related to *dog*, *cat*, and *bed* are distinguished by purple, green, and bright yellow, respectively. (c) is the normalized similarity matrix calculated by the cosine similarity between local features of the origin image and the feature of the category name ‘*bed*’ through CLIP, and (d) is the corresponding POT transportation plan where cosine distances between modalities are set as the cost matrix. The proportion of scores corresponding to concerned and noisy regions are highlighted in yellow and red, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Our work addresses this challenge by focusing on the most informative and relevant parts of the image, ensuring that the prototypes are more representative and robust in distinguishing between classes.

As shown in Fig. 1(a), the real-world image often contains multiple labeled objects, partially distributed in different image regions, *i.e.*, *cat*, *dog* and *bed*, as well as the unlabeled regions, *i.e.*, *window* to the upper right corner in Fig. 1(b). That said, existing methods of producing the class prototypes will be affected by the noise within the image. This is verified by a toy example. Specifically, since the image shown in Fig. 1(a) is labeled with *cat*, *dog* and *bed*, when creating a prototype for a class, *i.e.*, the *bed* class, the region of left down corner, belonging to the *dog* class, and the upper right corner, belonging to the noise, both attain information for the *bed* class, *e.g.*, non-zero similarity value between the word embedding of class *bed* and the local feature of visual feature map in Fig. 1(c). This observation motivates us to develop an adaptive algorithm that partially selects and aggregates the most relevant regions within images to generate robust and representative prototypes. Considering the values as the scores for regions related to the target category, *i.e.*, *bed*, as shown in Fig. 1(c) and Fig. 1(d) (60.74% vs. 88.56%). This demonstrates that the POT coupling directs nearly 30% more attention to the relevant regions, significantly reducing interference from irrelevant parts, *i.e.*, images of *dog*, *cat*, *etc.*. Specifically, we adopt episodic learning methods akin to ProtoNet [11], which generate class prototypes and make predictions by comparing the distance between image features and prototypes. Given the episodic learning paradigm’s nature, where labels and instances vary with each training step, we treat each episode as a unique sampling from the visual and label domains. We leverage the semantic information of the labels, namely, the text of category names, as prior knowledge to assist with the selection and partial aggregation scheme via vision language models (VLMs), benefiting from the unified space for text and image features that VLMs provide. The goal is to align the latent distribution of the two modalities through observed episodes. To this end, optimal transport (OT) is a prominent method in machine learning [12] for effectively measuring the distance between distributions and is extensively applied in areas such as generative models [13, 14], domain adaptation [15,16], *etc.* OT transports probability mass from the source to the target distribution at minimal cost. However, it necessitates the complete transfer of the source’s probability mass in OT and contradicts prior observations, *i.e.*, the noise region should not be considered in the transportation stage.

This is addressed by the partial optimal transport (POT) strategy, which is developed as an extension of the classical OT problem and is gaining traction in computational mathematics and machine learning [17,18]. POT replaces the mass conservation constraint by allowing only a fixed portion of the mass to be transported. Caffarelli et al. [19] have demonstrated that mass from active regions of the source distribution is fully transported to the target, while the remainder stays intact. This leads to a sparse solution to the POT problem, consistent with the required selective transportation attribute. As depicted in Fig. 1(d), the optimized POT plan focuses intensively on the target region while minimizing the transport of noise in image features, aligning with our

prior findings. This strategy helps the model focus on the most pertinent local image features and mitigates the influence of irrelevant ones.

Building on this, we develop the ProPOT method for ML-FSL, which generates improved prototypes using POT to partially aggregate local image features and delineate class features based on their relevance. Initially, we generate distributions for both text and local image features based on the similarity of each image–text pair, where the probability mass assigned to a specific feature reflects its informational value relative to the other modality. Subsequently, we address corresponding POT challenges to match features across modalities, selecting the most pertinent local image features through textual engagement. The resulting optimal transport plans serve as refined attention maps, enabling the formation of distinct class features through a weighted sum of relevant image features. Our contributions in this paper can be summarized as follows:

- (1) We propose a novel modality alignment loss between textual and visual modalities for multi-label tasks applying prompt learning methods, making it suitable for the nature of ML-FSL setting;
- (2) We introduce POT to select related image features to generate representations of classes without extra supervision, obtaining prototypes with reduced redundancy of information beneficial to classification;
- (3) We process ML-FSL tasks through several public multi-label datasets and evaluate our model’s performance. As compared to existing ML-FSL methods, our method achieves a state-of-the-art result.

2. Related work

2.1. Few-shot classification

Few-shot learning has been widely studied in the field of computer vision. The main idea of few-shot learning is to learn a model that can generalize well to unseen classes using only a few examples. Metric-learning-based methods [11,20,21] mainly follow the episodic training method, leveraging the distances between query and support features to make classifications. On the other hand, fine-tuning existing baselines with elaborated transfer learning methods [22–24] also achieves competitive performance without modification on training procedures. Moreover, many recent works [25–27] on FSL pay extensive attention to local features of images for better prototype representation. However, as a challenging and realistic problem, multi-label few-shot learning has not been comprehensively studied.

2.2. Multi-label few-shot classification

In multi-label few-shot classification, the task is to predict multiple labels per image, presenting a greater challenge than single-label classification. Alfassy et al. [7] first addressed this problem, proposing a model with data augmentation strategies to enhance performance. Chen et al. [8] introduced a graph-based approach to capture label dependencies by analyzing label co-occurrence. Simon et al. [9] extended prototypical networks and other few-shot learning methods to

multi-label classification, incorporating a prediction module for estimating the number of labels. However, averaging prototypes across all related images may not fully capture the complexities of multi-label tasks. To address this, Yan et al. [10] employ weak supervision from textual descriptions and a cross-modal attention mechanism to generate prototypes focused on relevant local image features.

2.3. Vision language models

Vision-language models (VLMs), such as CLIP [28], significantly advance the alignment of visual and textual representations. Trained on large-scale image–text pairs, CLIP leverages contrastive learning to link images with their corresponding textual descriptions, using a visual encoder and a text encoder to embed both modalities into a shared semantic space. Based on CLIP’s flexibility for various downstream tasks, numerous works have extended its applications. CoOp [29] and CoCoOp [30] focus on few-shot learning, introducing prompt-tuning methods to generate optimized and dynamic prompts. Sun et al. [31] address multi-label classification with insufficient annotations by developing dual prompts.

3. Preliminary

3.1. Optimal transport

3.1.1. Definition

Optimal transport (OT) is a mathematical framework that aims to find the most efficient way to transport mass distribution from the source to the target. The problem can be formulated as follows: given two probability distributions μ and ν on \mathbb{R}^d , find a transport plan γ that minimizes the cost of transporting μ to ν . The cost of transporting a unit of mass from μ to ν is given by a cost function $C(\mu, \nu)$. The optimal transport problem can be formulated as a linear program:

$$\min_{\gamma \in \Pi(\mu, \nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} C(x, y) d\gamma(x, y), \quad (1)$$

where $\Pi(\mu, \nu)$ is the set of all joint distributions with marginals μ and ν . To be simplified, we mainly consider the discrete case where μ and ν are discrete distributions with n and m points, respectively. In this case, the cost function C is a matrix $C \in \mathbb{R}^{n \times m}$, where C_{ij} is the cost of transporting mass from μ_i to ν_j , usually defined by a distance measure. The transport plan γ is a joint probability distribution $\gamma \in \mathbb{R}^{n \times m}$, where γ_{ij} is the amount of mass transported from source to the target. The optimal transport problem between two discrete distributions can be formulated as below:

$$\begin{aligned} \min_{\gamma \in \Pi(\mu, \nu)} \langle C, \gamma \rangle &= \sum_{i=1}^n \sum_{j=1}^m C_{ij} \gamma_{ij}, \\ \text{s.t.} \quad \sum_{j=1}^m \gamma_{ij} &= \mu_i, \\ \sum_{i=1}^n \gamma_{ij} &= \nu_j, \\ \gamma_{ij} &\geq 0. \end{aligned} \quad (2)$$

3.1.2. Entropic optimal transport regularization

While the OT problem can be solved with specific linear programming algorithms, the computational cost is $O(n^3 m^3)$, which becomes computationally prohibitive for large-scale datasets. To address this issue, Cuturi et al. proposed the entropic optimal transport regularization (EOT) method [32]. The EOT problem is formulated as below:

$$\min_{\gamma \in \Pi(\mu, \nu)} \langle C, \gamma \rangle - \epsilon H(\gamma), \quad (3)$$

where $H(\gamma) = -\sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} \log(\gamma_{ij})$ is the entropy of γ . The entropic regularization parameter ϵ controls the trade-off between the transportation cost and the entropy. The EOT problem can be solved with the Sinkhorn algorithm [32], which iteratively updates the transport plan.

3.2. Partial optimal transport

3.2.1. Definition

In partial optimal transport (POT), the objective is to determine the optimal transport plan between two probability distributions, where only a fixed proportion of the mass is required to be transported. Unlike traditional optimal transport which fully aligns both distributions, POT permits partial matching by allowing a portion of the mass to remain untransported, addressing situations where full alignment is unnecessary. This flexibility is particularly useful in cases such as resource allocation problems or when comparing distributions with outliers.

The modified OT problem to fit this case can be formulated as below:

$$\begin{aligned} \min_{\gamma \in \Pi(\mu, \nu)} \langle C, \gamma \rangle &= \sum_{i=1}^n \sum_{j=1}^m C_{ij} \gamma_{ij}, \\ \text{s.t.} \quad \sum_{j=1}^m \gamma_{ij} &\leq \mu_i, \\ \sum_{i=1}^n \gamma_{ij} &\leq \nu_j, \\ \sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} &= 1 - s, \\ \gamma_{ij} &\geq 0, \quad s \geq 0, \end{aligned} \quad (4)$$

where s is the remaining mass that does not need to be transported. This formulation of POT is particularly useful in scenarios where only part of the data is relevant for the task, such as multi-label classification or few-shot learning, where different regions or features may correspond to different classes, and full alignment between distributions is unnecessary or even undesirable.

3.2.2. Entropic partial optimal transport regularization

The partial optimal transport problem can be solved with the same entropic regularization method as the OT problem. The reformulated POT problem with entropic regularization is:

$$\begin{aligned} \min_{\gamma \in \Pi(\mu, \nu)} \langle C, \gamma \rangle - \epsilon H(\gamma), \\ \text{s.t.} \quad \sum_{j=1}^m \gamma_{ij} &\leq \mu_i, \\ \sum_{i=1}^n \gamma_{ij} &\leq \nu_j, \\ \sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} &= 1 - s, \\ \gamma_{ij} &\geq 0, \quad s \geq 0. \end{aligned} \quad (5)$$

For convenience, we denote this problem by its inputs and the result as:

$$\gamma = \text{POT}(C, \mu, \nu, s). \quad (6)$$

3.2.3. Solution to POT

To address the POT problem defined in Eq. (4), we follow Benamou et al. [33] to apply an iterative algorithm with entropic regularization. Details of the algorithm are shown below:

1. Initialization

Given the cost matrix C and regularization parameter ϵ , the initial transport matrix is defined as:

$$\mathbf{K} = \exp\left(-\frac{C}{\epsilon}\right), \quad (7)$$

which is then scaled to satisfy the total transported mass $1 - s$:

$$\mathbf{K} \leftarrow \mathbf{K} \cdot \frac{1 - s}{\sum \mathbf{K}}. \quad (8)$$

2. Iterative Steps

During each iteration, the transport matrix is alternately scaled along its rows and columns to progressively meet the marginal constraints μ and ν .

Let \mathbf{K}_1 represent the row-normalized matrix:

$$\mathbf{K}_1 = \text{diag} \left(\min \left(\frac{\mu}{\sum_j \mathbf{K}_{ij}}, \mathbf{1} \right) \right) \cdot \mathbf{K}, \quad (9)$$

where $\frac{\mu}{\sum_j \mathbf{K}_{ij}}$ normalizes the sum of each row to ensure that row sums do not exceed the vector μ .

The columns of \mathbf{K}_1 are then normalized as follows:

$$\mathbf{K}_2 = \mathbf{K}_1 \cdot \text{diag} \left(\min \left(\frac{\nu}{\sum_i \mathbf{K}_{ij}}, \mathbf{1} \right) \right), \quad (10)$$

where $\frac{\nu}{\sum_i \mathbf{K}_{ij}}$ normalizes the sum of each column to ensure that column sums do not exceed the vector ν .

Finally, the matrix \mathbf{K}_2 is scaled to ensure the total transported mass equals $1 - m$:

$$\mathbf{K} \leftarrow \mathbf{K}_2 \cdot \frac{1 - m}{\sum \mathbf{K}_2}. \quad (11)$$

3. Convergence

After each iteration, the algorithm computes the difference between the current and previous transport matrix. The error is defined as:

$$\text{error} = \|\gamma_{\text{prev}} - \gamma\|, \quad (12)$$

and the algorithm stops when the error falls below a pre-defined threshold.

This procedure iteratively normalizes the rows and columns of the transport matrix, ensuring that the solution complies with the marginal constraints while maintaining a smooth distribution through entropic regularization.

4. Method

4.1. Problem formulation

In this section, we introduce the notations and problem formulation of ML-FSL. Given a support set S with assigned labels from the label set C and a query set Q , the goal is to predict the labels of samples in Q given the S . We denote an episode as $\mathcal{E} = \{S, Q\}$, where $S = \{(I_i^S, y_i^S)\}_{i=1}^{N_S}$ and $Q = \{(I_i^Q, y_i^Q)\}_{i=1}^{N_Q}$. The symbol y_i is the one-hot label of image I_i in the support or query sets, and $y_i \in \mathbb{R}^{|C|}$ where C is the selected label set for each episode. In the traditional N -way K -shot setting for FSL, the support set S consists of N classes, each with K labeled images where K is small, e.g., 1 or 5. In the ML-FSL setting, each image is labeled with multiple labels, making the number of labels uncertain in each episode. To this end, we fix the number of samples in the support and query sets as N_S and N_Q , guaranteeing each class contains at least one sample, and still use N to denote the number of classes within an episode. To apply the episodic learning paradigm, the label set is divided into two disjoint sets C_{base} and C_{novel} . The images that only contain labels in C_{base} form the training dataset D_{train} , while those with labels in C_{novel} form the test dataset D_{test} . We train and test the model using images sampled from D_{train} and D_{test} , respectively.

4.2. Overview

This section introduces the overview of the proposed ProPOT for ML-FSL, as shown in Fig. 2. In each episode, we extract text features t_c from class names using the pre-trained CLIP text encoder with prompt tuning, and image features X_i^S from support images I_i^S using the CLIP image encoder. We have modified the image encoder to extract local feature maps; the output for image I_i^S is $X_i^S \in \mathbb{R}^{N_L \times d_v}$, comprising N_L local features and d_v feature dimensions. We align text and image features by maximizing the similarity between each correlated text

feature $t_c, c \in [1, 2, \dots, N]$ and the corresponding global image feature $x_i^S = \frac{1}{N_L} \sum_{j=1}^{N_L} x_{ij}^S, i \in [1, 2, \dots, N_S]$, where x_{ij}^S is the j th local feature of image I_i^S . With the well-matched features of the two modalities, we use POT to transport the information carried by image regions to the closely associated labels. We partially aggregate local features x_{ij}^S into class features p_i^c for each relevant class c according to the resulting transportation plans between visual and textual distributions, as illustrated by the POT Module in Fig. 2. Class features p_i^c are averaged across all images labeled c to form the final class prototype p_c . We optimize classification loss by minimizing the distance between the prototype and the global feature x_i^Q of each query image containing the corresponding class. During inference, class prototypes are generated in a similar manner, and query samples are classified based on their similarity to each prototype.

4.3. Modality alignment with prompt-based fine-tuning

To improve alignment between visual and textual modalities, we propose a prompt-based fine-tuning method using class names. The textual descriptions are extracted by the pre-trained CLIP [28] model. Following CoOp [29], we set the input prompts for class c as $\mathbf{P} = [P]_1^c [P]_2^c \dots [P]_{N_{\text{ctx}}}^c [\text{CLASS}]$, where $[\text{CLASS}] \in \mathbb{R}^{d_w}$ represents the word embedding of class c , $[P]_i^c \in \mathbb{R}^{d_w}$ is the i th trainable prompt embedding, and N_{ctx} is the number of prompts. The class-specific text features t_c , generated by the CLIP text encoder, are aligned with the image features. We apply the contrastive loss to maximize the similarity between text features t_c and global image features x_i^S of the same class, and minimize the similarity between text and image features from different classes. The text encoder remains fixed during training, while the prompts are updated through backpropagation. To fit in the ML-FSL problem, the cross-entropy loss is replaced by binary cross-entropy loss to calculate the loss between the predicted labels and the ground truth labels. The loss function is defined as:

$$\mathcal{L}_{\text{align}} = \frac{1}{N_S} \sum_{i=1}^{N_S} \sum_{c=1}^N \left(\frac{y_{ic}^S}{\|y_i^S\|_1} - \frac{\exp(s_{ic})}{\sum_{k=1}^N \exp(s_{ik})} \right)^2, \quad (13)$$

$$s_{ic} = \lambda \cos(x_i^S, t_c), \quad i \in [1, 2, \dots, N_S], \quad c \in [1, 2, \dots, N],$$

where λ is the scaling factor, N is the number of classes in the episode, and y_{ic}^S is the c th element of the one-hot label of support image I_i^S . The aligned features are then used to generate class prototypes for classification.

4.4. Partial optimal transport for prototype generation

To generate class prototypes from the support set, we propose a partial aggregation scheme that selects the most relevant local image features. For each support image I_i^S , we determine the relevance of its local features to each class by solving several POT problems according to its labels. The resulting transportation plans serve as weight matrices, allowing us to compute the weighted average of the local features to generate corresponding class features.

Specifically, we require the probability distributions μ and ν for the textual and visual modalities, the cost matrix C representing the transportation cost between modalities, and the remaining mass s denoting the portion of probability mass that should remain untransported within a single POT problem. For one of the corresponding class c , we define the distribution $\mu_i = (\mu_{ic}) \in \mathbb{R}^N$ of textual modality as

$$\mu_{ic} = \frac{e^{-s_{ic}}}{\sum_{k=1}^N e^{-s_{ik}}}, \quad (14)$$

where s_{ic} is defined the same way as Eq. (13). Here μ_i reveals the overall distribution of information I_i^S carries. The distribution of local

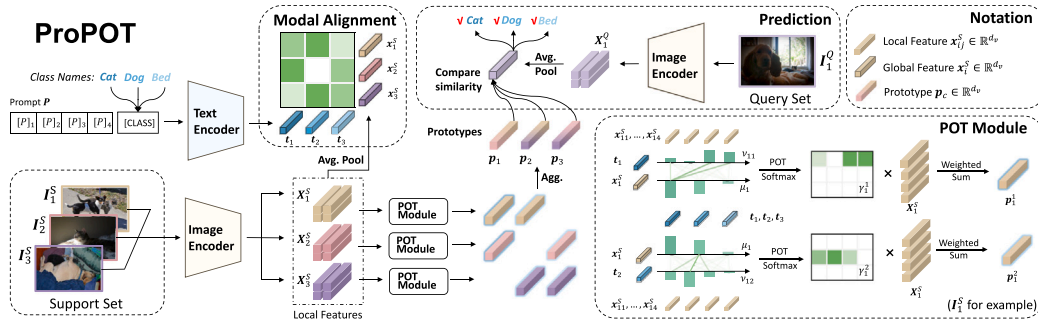


Fig. 2. Overview of ProPOT method using a 3-way 1-shot task as an example.

image features $v_i^c = (v_{ij}^c) \in \mathbb{R}^{N_L}$ is defined as

$$v_{ij}^c = \frac{e^{-s_{ij}^c}}{\sum_{k=1}^{N_L} e^{-s_{ik}^c}}, \quad (15)$$

where s_{ij}^c represents the cosine similarity between the j th local feature \mathbf{x}_{ij}^S and the text feature t_c of the related class c , defined as $s_{ij}^c = \lambda \cos(\mathbf{x}_{ij}^S, t_c)$. Similarly, v_i^c indicates the information of class c that each local feature \mathbf{x}_{ij}^S carries. A specific image I_i^S can carry multiple labels in the multi-label setting. Therefore, we obtain visual distributions in the quantity of $\|\mathbf{y}_i^S\|_1$ corresponding to different classes and there are such a number of POT problems to be solved for I_i^S . It ensures that we engage more related local features in the prototype generation.

Then, the cost matrix C_i is defined by the regularized cosine distance between the local image features and the text features. Specially, we compute it as $C_i = \text{softmax}((1 - S_i)/\tau)$, where S_i is the cosine similarity matrix,

$$S_i = T \times X_i^S, T = [t_1, \dots, t_{N_C}]^T, X_i^S = [x_i^1, \dots, x_i^{N_L}]. \quad (16)$$

As for one POT optimization we aim to obtain the plan transporting local image features to one of its related classes, setting the remaining mass s as the number of the rest classes, $s = 1 - \frac{1}{\|\mathbf{y}_i^S\|_1}$.

We calculate transportation plans γ_i^c for image I_i^S which show the potential of the local features to be the prototype of different classes by solving the following POT problems:

$$\gamma_i^c = \text{POT}(C_i, \mu_i, v_i^c, s), c \in \{k \mid y_{ik}^S = 1, k = 1, 2, \dots, N\}. \quad (17)$$

The resulting transportation plan serves as a weight matrix to aggregate the corresponding class feature p_i^c , which contains the information of concerned regions of I_i^S w.r.t. class c . The class feature is calculated as below:

$$p_i^c = (\gamma_i^c)_{c,:} \times X_i^S, \quad (18)$$

where $(\gamma_i^c)_{c,:}$ denotes the c th row of γ_i^c . Finally the prototypes p_c for each class c are obtained by averaging the class features of support images that contain the specific label,

$$\begin{aligned} p_c &= \frac{1}{|\mathcal{X}_c^S|} \sum_{X_i^S \in \mathcal{X}_c^S} (\gamma_i^c)_{c,:} \times X_i^S \\ &= \frac{1}{|\mathcal{X}_c^S|} \sum_i p_i^c, \end{aligned} \quad (19)$$

where \mathcal{X}_c^S refers to the set of support samples containing class c . The distances between the global feature \mathbf{x}_i^Q of query image I_i^Q and prototypes are then used to predict the labels. The classification loss function for query samples is defined as:

$$\mathcal{L}_{cls} = \frac{1}{N_Q} \sum_{i=1}^{N_Q} \sum_{c=1}^N \left(\frac{y_{ic}^Q}{\|\mathbf{y}_i^Q\|_1} - \frac{\exp(-z_{ic})}{\sum_{k=1}^N \exp(-z_{ik})} \right)^2, \quad (20)$$

where $z_{ic} = \lambda \|\mathbf{p}_c - \mathbf{x}_i^Q\|_2^2$ denotes the squared Euclidean distance between the prototype of class c and the global features of query sample I_i^Q . The final loss function is the combination of the alignment loss and

the classification loss, which is defined as:

$$\mathcal{L} = \alpha \mathcal{L}_{align} + \beta \mathcal{L}_{cls}. \quad (21)$$

5. Experiments

5.1. Datasets

We evaluate the performance of our method on multi-label few-shot tasks using three datasets. An episodic training strategy is employed, where in each episode, a few classes and samples are drawn from the base classes to form the support and query sets. The training images contain only labels from the base classes. During testing, the model is evaluated on novel classes that are unseen during training. Test samples belong exclusively to novel classes and do not carry any labels from the base classes. The details of the datasets are as follows:

MS-COCO [34]: The MS-COCO (Microsoft Common Objects in Context) dataset is a widely used benchmark in computer vision, particularly for tasks such as object detection, segmentation, and captioning. In this study, the MS-COCO dataset is utilized to perform a multi-label classification task. This dataset provides a rich collection of images, each annotated with multiple labels, which correspond to various objects present within the image. MS-COCO contains about 120,000 labeled images, divided into training, validation, and test sets. To evaluate the performance of our model, we use the 2014 version of the dataset, which contains 80 object categories. To perform 10-way few-shot classification tasks, We follow Simon et al. [9] to split the dataset, i.e., 50,10,20 categories are used as base, validation, and novel classes, respectively. To perform 16-way few-shot classification tasks, we follow Alfassy et al. [7] and Yan et al. [10] to split the dataset, using 52, 12, and 16 categories as the base, validation, and novel classes, respectively. The detailed split is shown in Table 1.

NUS-WIDE [35]: The NUS-WIDE (National University of Singapore - Web Image Database) dataset consists of a large collection of images harvested from the web, annotated with multiple labels representing various objects and concepts present within each image. NUS-WIDE contains over 260,000 images, each tagged with multiple labels derived from user-provided tags and further refined through a semi-automated process. These labels span 81 categories, including objects, scenes, and activities, making the dataset suitable for evaluating models in multi-label classification scenarios. Considering there is no current ML-FSL research tackling this dataset, We randomly divided the dataset into 41, 20, and 20 categories for base, validation, and novel classes, respectively.

Pascal-VOC [36]: The Pascal-VOC (Visual Object Classes) dataset consists of over 5000 images annotated with object bounding boxes and class labels. In this study, we use the Pascal-VOC 2012 dataset, which contains 20 object categories. We follow the split of the dataset used in the CMW papers where 14 categories are used as base classes and 6 categories are used as novel classes for testing. The detailed split is shown in Table 2.

Table 1
The dataset split for MS-COCO.

MS-COCO						
Training classes			Validation classes		Test classes	
Backpack	Cell phone	Knife	Sink	Airplane	Apple	Banana
Baseball bat	Chair	Laptop	Spoon	Carrot	Bed	Bird
Baseball glove	Clock	Motorcycle	Suitcase	Frisbee	Boat	Broccoli
Bear	Couch	Mouse	Person	Tennis racket	Donuts	Elephant
Bench	Cow	Oven	Tie	Kite	Fire hydrant	Horse
Bicycle	Cup	Person	Toaster	Orange	Microwave	Parking meter
Book	Dining table	Pizza	Traffic light	Skis	Sandwich	Scissors
Bottle	Dog	Potted plant	Truck	Stop sign	Sheep	Skateboard
Bowl	Fork	Refrigerator	TV	Surfboard	Teddy bear	Toilet
Bus	Giraffe	Remote	Umbrella	Toothbrush	Train	Wine glass
Cake	Hairdryer	Snowboard	Vase			
Car	Handbag	Sports ball	Zebra			
Cat	Keyboard					

Table 2
The dataset split for Pascal-VOC.

Pascal-VOC			
Training classes		Test classes	
Aero-plane	Car	Person	Dog
Bicycle	Chair	Train	Sofa
Bird	Cow		Cat
Boat	Dining table		Potted plant
Bottle	Horse		TV monitor
Bus	Motorbike		Sheep

5.2. Implementation details

In our implementation, we adopt different settings for ML-ProtoNet and CMW to conduct our experiments. The key differences between these two approaches lie in the number of categories and the number of samples per class in each episode. ML-ProtoNet follows the traditional episodic learning paradigm for few-shot learning (FSL), maintaining a consistent number of categories across all episodes. In contrast, CMW adjusts the number of classes in each episode to match the total number of classes in the entire dataset split. Specifically, for ML-ProtoNet, all few-shot tasks across datasets are 10-way, while in CMW, the number of classes varies across datasets, with 16-way, 20-way, and 6-way tasks for MS-COCO, NUS-WIDE, and Pascal-VOC, respectively, during the testing phase. Additionally, ML-ProtoNet ensures that at least one sample is present for each class in the query set. In contrast, CMW samples 4 instances per class, maintaining a fixed query set size of $4 \times n$ for an n -way task.

We use different scales of vision model as feature extractors according to the few-shot tasks. For 10-way 1-shot and 10-way 5-shot tasks, we use the ConvNet-4 [20] architecture to be compared with ML-ProtoNet [9]. The learning rate is initially set to 0.001. For 16-way 1-shot and 16-way 5-shot settings, we use the ResNet-50 [37] architecture which is compared with CMW [10]. To evaluate our model on the LaSO [7] benchmark where episodic testing is not implemented, we use the GoogleNet-v3 [38] and ResNet-101 [37] as our backbone, which allows for comparison with all previous methods. The learning rates for these two settings are initially set to $1e-5$. The feature extractor for the text modality in our model is the default CLIP text encoder, whose weights are frozen during all stages and only the parameterized prompts are used for training. We use the Adam optimizer [39] and a decreasing learning rate scheduler for all experiments. The model is trained for 100,000 episodes for each task and is evaluated on the validation set every 1000 episodes. The best model is selected based on the validation set performance except for experiments on PASCAL-VOC. Considering the limited number of classes in PASCAL-VOC, we use the model trained after the last episode for evaluation to leverage all base classes for training. We set the weight of entropic regularization to 0.001, the number of local features to 49, and the weight α , β for alignment loss and classification loss to 0.01 and 1 respectively for all experiments.

Table 3
Comparison with state-of-the-art methods on MS-COCO dataset using LaSO benchmark on 1-shot and 5-shot scenarios.

Method	Backbone	1-shot	5-shot
LaSO	GoogleNet-v3	45.3	58.1
ML-ProtoNet	ConvNet-4	50.2	60.4
KGGR	GoogleNet-v3	49.4	61.0
	ResNet-101	52.3	63.5
CMW	GoogleNet-v3	53.4	65.1
	ResNet-101	55.7	68.2
ProPOT	ConvNet-4	53.2	62.5
	GoogleNet-v3	56.7	65.8
	ResNet-101	60.3	69.2

Table 4
Comparison with meta-learning methods on MS-COCO dataset following ML-ProtoNet setting on 1-shot and 5-shot scenarios.

Method	MS-COCO		NUS-WIDE		Pascal	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ML-ProtoNet	56.7	66.7	62.1	65.3	54.3	61.5
ML-ProtoNet+NLC	58.5	68.0	62.7	66.9	57.2	63.3
LPN+NLC	60.4	69.1	62.5	67.1	57.9	63.1
ProPOT	61.9	72.3	63.5	68.4	62.4	65.1

5.3. Results

We compare our model with ML-ProtoNet, CMW, and LaSO on the MS-COCO, NUS-WIDE and Pascal-VOC datasets. However, these methods apply different sampling strategies for generating episodes. Therefore, we follow each method separately to ensure fair comparisons. Firstly we evaluate our model on the LaSO benchmark, which applies a traditional test strategy, using all related samples in the MS-COCO dataset to evaluate the model. The results are shown in Table 3, with the best results highlighted in bold. Our method outperforms the state-of-the-art techniques. ProPOT with GoogleNet-v3 as the image encoder gains 3.3% and 0.7% of mAP improvement compared with CMW on 1-shot and 5-shot settings, together with 4.6% and 1.0% if using ResNet-101.

We also test our model using ConvNet-4 as the image encoder in 10-way 1-shot and 10-way 5-shot tasks following the dataset split of [9] and the results are shown in Table 4. To compare with CMW, we test our model using the split provided by [10] with ResNet-50 as the image encoder under the setting of 16-way 1-shot and 16-way-5-shot tasks and the results are shown in Table 5. Both of the latter two settings are based on episodic training and testing. Our model outperforms all other state-of-the-art methods across all settings.

Table 5
Comparison following CMW setting on 1-shot and 5-shot scenarios.

Method	Backbone	MS-COCO		Pascal	
		1-shot	5-shot	1-shot	5-shot
CMW	ResNet-50	57.5	61.2	60.0	60.8
	ResNet-101	58.3	64.3	61.8	62.1
ProPOT	ConvNet-4	54.1	62.8	62.4	67.8
	ResNet-50	61.6	66.5	64.7	68.8
	ResNet-101	62.3	68.7	69.4	72.3

Table 6
Precision, recall and F1 scores on MS-COCO dataset under 1-shot scenarios compared with meta-learning methods.

Method	Backbone	Precision	Recall	F1-score
ML-ProtoNet	ConvNet-4	37.2	70.7	48.8
ProPOT	ConvNet-4	39.4	75.6	51.7
CMW	ResNet-50	34.5	40.1	37.1
ProPOT	ResNet-50	31.5	90.6	39.8

5.4. Ablation study

In this section, we conduct an ablation study to evaluate the effectiveness of each module in our model. The study is carried out using the split provided by Simon et al. [9] under the 10-way 1-shot setting.

5.4.1. Analysis on evaluation metrics

For better method comparison, we provide a more detailed metric table to present the precision, recall, and F1 scores, as shown in Table 6. For ProPOT and ML-ProtoNet, the threshold to determine positive and negative categories is set to 0.1, as the classification logit follows a softmax function. In contrast, the threshold for CMW is set to 0.5, consistent with the original paper. Our method achieves higher F1 scores compared to the other methods, indicating better overall performance. However, the precision score is slightly lower than that of CMW, likely due to the difference in threshold settings.

5.4.2. Analysis on statistical significance

We report the confidence interval as a measure of model performance. Specifically, we calculate the 95% confidence interval for our proposed method across three benchmarks, as shown in Table 7. The results demonstrate that our method clearly outperforms existing state-of-the-art approaches, highlighting its superiority. Note that [9] evaluates methods only on the MS-COCO dataset; therefore, we use the results from [9] for MS-COCO, while the results for NUS-WIDE and Pascal-VOC are re-implemented. We also calculate p-values for ProPOT on NUS-WIDE and Pascal-VOC datasets compared to ML-ProtoNet, yielding 0.027 and 0.001. Considering a common level 0.05 for the p -value, our method realizes a significant improvement over previous methods as the p -values are all smaller than the given level.

5.4.3. Analysis on model design

We perform an ablation study to assess the contribution of each module in our model, with results summarized in Table 8. “Modality Alignment” refers to the implementation of the alignment loss \mathcal{L}_{align} . “Prompt Learning” indicates the presence of trainable prompts, without which text features remain fixed during training. “POT” represents the partial aggregation scheme, compared against a baseline that averages all related image features to generate prototypes. Each component contributes to overall performance, with modality alignment significantly enhancing results, and POT providing further gains.

5.4.4. Analysis on distance metrics

We evaluate the model using various combinations of distance metrics for intra-modal and inter-modal calculations. Intra-modal distance refers to the distance between text and image features, while inter-modal distance refers to the distance between image features and the corresponding prototypes. The results, presented in Table 9, indicate that cosine distance is particularly effective for intra-modal alignment, consistent with CLIP’s design philosophy. In contrast, using Euclidean distance for this task significantly degrades performance. The optimal performance is achieved when cosine distance is employed for intra-modal calculations and Euclidean distance for inter-modal comparisons.

5.4.5. Analysis on partial optimal transport

We perform a detailed ablation study on the core module of our model, the POT mechanism, to assess the impact of key hyper-parameters on performance. The outcomes are presented in Tables 10, 11, 12 13. In each test, the specific hyper-parameter is evaluated while keeping the others at their default settings. The results demonstrate that the entropic regularization weight ϵ plays a critical role, particularly when using larger backbones. ϵ regulates the sparsity of the transportation plan, where lower values lead to sparser solutions—fewer local image features are selected for prototype generation. As seen in Table 10, a smaller ϵ generally yields better performance, highlighting POT’s ability to prioritize relevant features while discarding irrelevant ones. However, as ϵ decreases, training becomes more challenging, resulting in lower model stability and poorer performance due to numerical instabilities that may affect overall accuracy.

The remaining mass s is also a crucial hyper-parameter in POT. s controls the amount of mass that can be transported from the source to the target. A specific POT problem involves matching the information of a single class, as carried by the image, with the corresponding text features. From this perspective, s quantifies the proportion of regions irrelevant to the target class. The results of different remaining masses are demonstrated in Table 13. The results show that among the settings of fixed remaining mass, $s = 0.67$ obtains the best result, which means only one-third of the mass can be transported from the source to the target. The performance drops when s is set to 0.5 and 0.33, and the performance is the worst when s is set to 0.00, which means all the mass can be transported from the source to the target, namely the setting of OT problem. The dynamic setting of s achieves the best performance, indicating that tuning the mass amount according to the number of labels makes sense.

Additionally, calculating the remaining masses for each support image can be time-consuming. A more efficient approach is to fix s for different datasets, which results in only a slight loss of accuracy. In Table 12, we further investigate the effect of different values of s for the datasets used in our manuscript, under 10-way 1-shot tasks with ResNet-50 as the visual encoder. The best result for each dataset is highlighted in bold. For MS-COCO and NUS-WIDE, which have more than 3 labels per image, a smaller remaining mass performs better. In contrast, Pascal-VOC, with an average of 1.4 labels per image, benefits from a larger remaining mass, as smaller masses lead to information loss and worse performance.

In addition, the distributions of text features and image features also affect the performance, but the impact is not as significant as the other two hyper-parameters. It may arise from the strong selective function of POT, which focuses on the most related features according to the cost matrix, even if the possibility masses of all features are the same. Table 11 shows the results, where Uni., GT, Logits mean uniform distribution, ground-truth distribution and output logits defined in Section 4 respectively. Considering that the distribution sums up to 1, we modify the one-hot label y_i^S to the ground-truth distribution $y_i^{S'} = \frac{y_i^S + \epsilon}{\|y_i^S + \epsilon\|_1}$, where ϵ is $1e-5$ to avoid numerical instability.

Table 7
Comparison with meta-learning methods following ML-ProtoNet setting on 1-shot and 5-shot scenarios.

Method	MS-COCO		NUS-WIDE		Pascal-VOC	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ML-ProtoNet	56.7	66.7	62.1 ± 1.02	65.3 ± 1.24	54.3 ± 0.78	61.5 ± 0.99
ML-ProtoNet+NLC	58.5	68.0	62.7 ± 0.97	66.9 ± 0.98	57.2 ± 1.03	63.3 ± 1.12
LPN+NLC	60.4	69.1	62.5 ± 1.35	67.1 ± 1.51	57.9 ± 1.26	63.1 ± 1.19
ProPOT	61.9 ± 0.93	72.3 ± 1.11	63.5 ± 0.82	68.4 ± 1.03	62.4 ± 1.10	65.1 ± 0.99

Table 8
Ablation study on each module in our method with ConvNet-4 as the image encoder.

Modality Alignment	Prompt Learning	POT	mAP
✗	✗	✗	56.7
✓	✗	✗	59.2
✓	✓	✗	59.5
✓	✓	✓	61.9

Table 9
Ablation study on distance used in the method, where Euc. denotes the Euclidean distance and Cos. denotes the cosine distance.

Intra-modal	Inter-modal	Backbone	mAP
Euc.	Euc.	ConvNet-4	57.1
Euc.	Cos.		61.9
Cos.	Euc.		42.3
Cos.	Cos.		59.5
Euc.	Euc.	ResNet-50	67.7
Euc.	Cos.		74.9
Cos.	Euc.		61.3
Cos.	Cos.		70.2

Table 10
Ablation study on the weight of entropic regularization.

Backbone	Entropic weight ϵ	mAP
ConvNet-4	2	61.4
	1	61.0
	0.1	61.5
	0.01	61.9
	0.009	61.7
ResNet-50	2	74.4
	1	74.3
	0.1	74.7
	0.01	74.9
	0.009	74.3

Table 11
Ablation study on the remaining mass of POT.

Backbone	Remaining mass s	mAP
ConvNet-4	0.67	61.7
	0.50	61.7
	0.33	61.4
	0.00	61.5
	Dynamic	61.9
ResNet-50	0.67	74.4
	0.50	74.7
	0.33	74.1
	0.00	73.1
	Dynamic	74.9

Table 12
Ablation study on different remaining mass s across three datasets.

Remaining mass s	MS-COCO	NUS-WIDE	Pascal-VOC
0.00	73.1	76.2	75.1
0.33	74.1	76.4	75.8
0.50	74.7	77.1	74.7
0.67	74.4	76.9	73.6

Table 13
Ablation study on distributions μ of text features and ν of image features, where Uni., GT and Logit denote the uniform distribution, normalized ground-truth label and the predicted logit, respectively.

Backbone	μ	ν	mAP
ConvNet-4	Uni.	Uni.	61.6
	GT	Uni.	58.6
	Logit	Uni.	61.9
	Logit	Logit	61.9
ResNet-50	Uni.	Uni.	74.7
	GT	Uni.	74.2
	Logit	Uni.	74.5
	Logit	Logit	74.9

5.5. Visualization

In this section, we visualize the transportation plans of POT w.r.t. local image features and labels, and contrast them with direct similarity matrices between visual and textual modalities to highlight the efficacy of ProPOT’s partial aggregation scheme.

In Fig. 3, all features are extracted using ProPOT with ResNet-101 after the training stage. It can be observed that POT plans predominantly focus on the informative regions of images to generate prototypes and effectively disregard irrelevant areas. For instance, comparing Fig. 3(a1) and (a2), POT selectively focuses on the location of the class *apple*, while (a1) spreads unnecessary attention across the left part of the image which contains unrelated objects. This phenomenon is also evident in the other visualizations. It is important to note that the quality of POT transportation plans is highly associated with the similarity matrices, which serve as the cost matrices for POT problems. For example, in Fig. 3(c2) and (c4), POT plans also allocate weights to unrelated regions due to the confusion made by corresponding similarity matrices.

6. Conclusion

In this paper, we have introduced a novel approach to multi-label few-shot learning, addressing a challenging area that often limits the performance of existing models. By leveraging weak supervision from the pre-trained VLM, our method generates distinct and informative representations of multiple classes through a proposed regional aggregation scheme that utilizes Partial Optimal Transport (POT). This innovative strategy is specifically tailored for few-shot tasks involving unseen classes, enhancing the model’s ability to generalize in scenarios with limited labeled data. Our empirical evaluations demonstrate that our approach achieves state-of-the-art performance across several benchmark datasets, outperforming the latest methodologies and significantly surpassing the previously best-performing method. These results underscore the effectiveness of integrating optimal transport mechanisms within the context of few-shot learning. Further investigation is needed to address potential scalability issues when applying our method to more complex datasets, as the complexity of optimal transport (OT) algorithms increases rapidly with problem scale. Additionally, the quality of feature representations plays a critical role in the success of the optimal transport process. While our approach demonstrates strong empirical performance, there is still room for



Fig. 3. Visualization of the aggregation scheme for local features to generate prototypes. The normalized similarity matrices are calculated by the cosine similarity between text and local image features, followed by a softmax normalization with a fixed temperature. POT plans are obtained by solving the POT problems defined in Section 4.

improvement in data preprocessing [40,41] and feature extraction. Enhancing the ability to capture more discriminative and informative features, particularly in noisy or less structured environments, remains an area of ongoing research. Future work could focus on integrating more advanced feature extraction methods or exploring alternative forms of regularization to improve the robustness and quality of the learned representations. A more comprehensive exploration of strategies to mitigate these challenges will be essential for the broader applicability of our framework.

CRediT authorship contribution statement

Pengfei Fang: Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization. **Zhihong Chen:** Writing – original draft, Software, Methodology, Investigation. **Hui Xue:** Resources, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62306070 and 62476056) and the Social Development Science and Technology Project of Jiangsu Province (No.

BE2022811). Furthermore, the work was also supported by the Big Data Computing Center of Southeast University.

Data availability

Data will be made available on request.

References

- [1] Z. Zhao, Q. Liu, W. Cao, D. Lian, Z. He, Self-guided information for few-shot classification, *Pattern Recognit.* 131 (2022) 108880.
- [2] H. Xu, J. Wang, H. Li, D. Ouyang, J. Shao, Unsupervised meta-learning for few-shot learning, *Pattern Recognit.* 116 (2021) 107951.
- [3] Z. Yu, L. Chen, Z. Cheng, J. Luo, Transmatch: A transfer-learning scheme for semi-supervised few-shot learning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020*, pp. 12856–12864.
- [4] M. Yazdanpanah, A.A. Rahman, M. Chaudhary, C. Desrosiers, M. Havaei, E. Belilovsky, S.E. Kahou, Revisiting learnable affines for batch norm in few-shot transfer learning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022*, pp. 9109–9118.
- [5] S. Bartunov, D. Vetrov, Few-shot generative modelling with generative matching networks, in: *International Conference on Artificial Intelligence and Statistics, AISTATS, PMLR, 2018*, pp. 670–678.
- [6] S. Sheynin, S. Benaim, L. Wolf, A hierarchical transformation-discriminating generative model for few-shot anomaly detection, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2021*, pp. 8495–8504.
- [7] A. Alfassy, L. Karlinsky, A. Aides, J. Shtok, S. Harary, R. Feris, R. Giryes, A.M. Bronstein, Laso: Label-set operations networks for multi-label few-shot learning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019*, pp. 6548–6557.

- [8] T. Chen, L. Lin, R. Chen, X. Hui, H. Wu, Knowledge-guided multi-label few-shot learning for general image recognition, *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* 44 (3) (2020) 1371–1384.
- [9] C. Simon, P. Koniusz, M. Harandi, Meta-learning for multi-label few-shot classification, in: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, WACV, 2022*, pp. 3951–3960.
- [10] K. Yan, C. Zhang, J. Hou, P. Wang, Z. Bouraoui, S. Jameel, S. Schockaert, Inferring prototypes for multi-label few-shot image classification with word vector guided attention, in: *Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, No. 3, AAAI, 2022*, pp. 2991–2999.
- [11] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: *Proceedings of the Advances in Neural Information Processing Systems, NIPS, 2017*.
- [12] A. Khamis, R. Tsuchida, M. Tarek, V. Rolland, L. Petersson, Scalable optimal transport methods in machine learning: A contemporary survey, *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* (2024).
- [13] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: *Proceedings of the International Conference on Machine Learning, ICML, PMLR, 2017*, pp. 214–223.
- [14] C. Bunne, D. Alvarez-Melis, A. Krause, S. Jegelka, Learning generative models across incomparable spaces, in: *Proceedings of the International Conference on Machine Learning, ICML, PMLR, 2019*, pp. 851–861.
- [15] N. Courty, R. Flamary, A. Habrard, A. Rakotomamonjy, Joint distribution optimal transportation for domain adaptation, in: *Proceedings of the Advances in Neural Information Processing Systems, NIPS, 2017*.
- [16] C. Zhang, Y. Cai, G. Lin, C. Shen, Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020*, pp. 12203–12213.
- [17] K. Nguyen, D. Nguyen, T. Pham, N. Ho, et al., Improving mini-batch optimal transport via partial transportation, in: *Proceeding of the International Conference on Machine Learning, ICML, PMLR, 2022*, pp. 16656–16690.
- [18] B. Riaz, Y. Karahan, A.J. Brockmeier, Partial optimal transport for support subset selection, *IEEE Trans. Mach. Learn. Res. (TMLR)* (2023).
- [19] L.A. Caffarelli, R.J. McCann, Free boundaries in optimal transport and monge-ampere obstacle problems, *Ann. Math.* (2010) 673–730.
- [20] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, in: *Proceedings of the Advances in Neural Information Processing Systems, NIPS, 2016*.
- [21] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: Relation network for few-shot learning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2018*, pp. 1199–1208.
- [22] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C.F. Wang, J.-B. Huang, A closer look at few-shot classification, in: *Proceedings of the International Conference on Learning Representations, ICLR, 2019*.
- [23] M. Goldblum, S. Reich, L. Fowl, R. Ni, V. Cherepanova, T. Goldstein, Unraveling meta-learning: Understanding feature representations for few-shot tasks, in: *Proceedings of the International Conference on Machine Learning, ICML, PMLR, 2020*, pp. 3607–3616.
- [24] G.S. Dhillon, P. Chaudhari, A. Ravichandran, S. Soatto, A baseline for few-shot image classification, in: *Proceedings of the International Conference on Learning Representations, ICLR, 2020*.
- [25] W. Chen, Z. Zhang, W. Wang, L. Wang, Z. Wang, T. Tan, Few-shot learning with unsupervised part discovery and part-aligned similarity, *Pattern Recognit.* 133 (2023) 108986.
- [26] M. Liang, S. Huang, S. Pan, M. Gong, W. Liu, Learning multi-level weight-centric features for few-shot learning, *Pattern Recognit.* 128 (2022) 108662.
- [27] H. Huang, Z. Wu, W. Li, J. Huo, Y. Gao, Local descriptor-based multi-prototype network for few-shot learning, *Pattern Recognit.* 116 (2021) 107935.
- [28] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: *Proceedings of the International Conference on Machine Learning, ICML, PMLR, 2021*, pp. 8748–8763.
- [29] K. Zhou, J. Yang, C.C. Loy, Z. Liu, Learning to prompt for vision-language models, *Int. J. Comput. Vis. (IJCV)* 130 (9) (2022) 2337–2348.
- [30] K. Zhou, J. Yang, C.C. Loy, Z. Liu, Conditional prompt learning for vision-language models, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022*, pp. 16816–16825.
- [31] X. Sun, P. Hu, K. Saenko, Dualcoop: Fast adaptation to multi-label recognition with limited annotations, in: *Proceedings of the Advances in Neural Information Processing Systems, NIPS, 2022*, pp. 30569–30582.
- [32] M. Cuturi, Sinkhorn distances: Lightspeed computation of optimal transport, in: *Proceedings of the Advances in Neural Information Processing Systems, NIPS, 2013*.
- [33] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, G. Peyré, Iterative bregman projections for regularized transportation problems, *SIAM J. Sci. Comput.* 37 (2) (2015) A1111–A1138.
- [34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: *Proceedings of the European Conference on Computer Vision, ECCV, Springer, 2014*, pp. 740–755.
- [35] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, Nus-wide: A real-world web image database from national university of singapore, in: *Proceedings of the ACM International Conference on Image and Video Retrieval, CIVR, 2009*, pp. 1–9.
- [36] M. Everingham, S.A. Eslami, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, *Int. J. Comput. Vis. (IJCV)* 111 (2015) 98–136.
- [37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2016*, pp. 770–778.
- [38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2016*, pp. 2818–2826.
- [39] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: *Proceedings of the International Conference on Learning Representations, ICLR, 2015*.
- [40] A.A.M. Dawood, A.S. Abdulaziz, A.J. Mohammed, Rlc-based image compression using wavelet decomposition with zero-setting of unnecessary sub-bands, *J. Eng. Sci. Technol.* 17 (1) (2022) 0391–0403.
- [41] B. Zhang, J. Sun, F. Sun, F. Wang, B. Zhu, Image deblurring method based on self-attention and residual wavelet transform, *Expert Syst. Appl.* 244 (2024) 123005.

Pengfei Fang is an Associate Professor at the School of Computer Science and Engineering, Southeast University (SEU), China. Before joining SEU, he was a post-doctoral fellow at Monash University in 2022. He received the Ph.D. degree from the Australian National University and DATA61-CSIRO in 2022, and the M.E. degree from the Australian National University in 2017. His research interests include computer vision and machine learning.

Zhihong Chen is a master student at the School of Computer Science and Engineering, Southeast University (SEU), China. His research interests include computer vision and machine learning.

Hui Xue is currently a professor of School of Computer Science and Engineering at Southeast University, China. She received the B.Sc. degree in Mathematics from Nanjing Norm University in 2002. In 2005, she received the M.Sc. degree in Mathematics from Nanjing University of Aeronautics & Astronautics (NUAA). And she also received the Ph.D. degree in Computer Application Technology at NUAA in 2008. Her research interests include pattern recognition and machine learning.