Semantic Guided Latent Parts Embedding for Few-Shot Learning

Fengyuan Yang[1,2], Ruiping Wang[1,2,3], Xilin Chen[1,2]
[1]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
[3]Beijing Academy of Artificial Intelligence, Beijing, 100084, China
fengyuan.yang@vipl.ict.ac.cn, {wangruiping, xlchen}@ict.ac.cn

Abstract

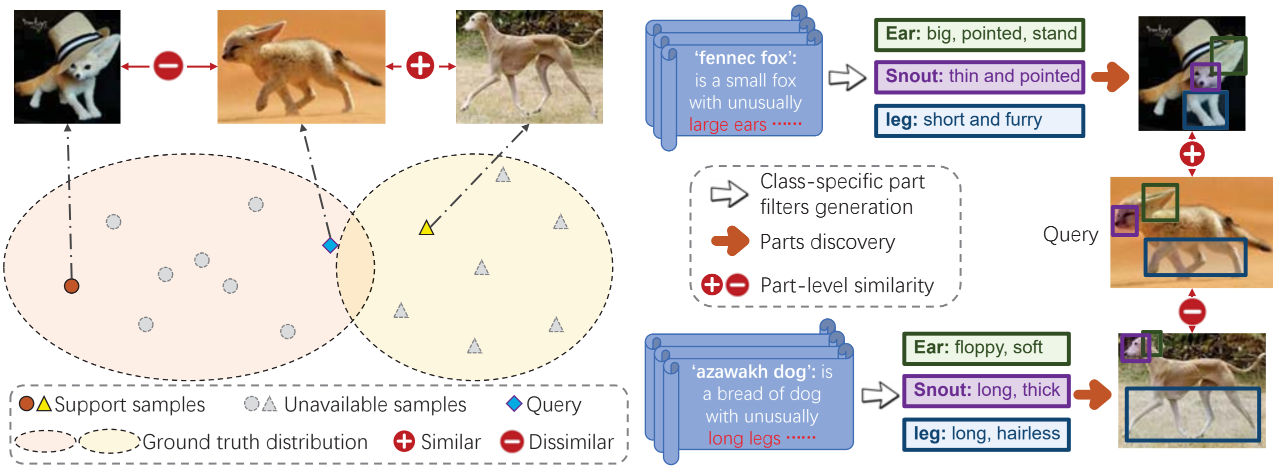
The ability of few-shot learning (FSL) is a basic requirement of intelligent agent learning in the open visual world. However, existing deep learning systems rely too heavily on large numbers of training samples, making it hard to learn new categories efficiently from limited size of training data. Two key challenges of FSL are insufficient comprehension and imperfect modeling of the few-shot novel class. For insufficient visual comprehension, semantic knowledge which is information from other modalities can help replenish the understanding of novel classes. But even so, most works still suffer from the second challenge because the single global class prototype they adopted is extremely unstable and imperfect given the larger intra-class variation and harder inter-class discrimination in FSL scenario. Thus, we propose to represent each class by its several different parts with the help of class semantic knowledge. Since we can never pre-define parts for unknown novel classes, we embed them in a latent manner. Concretely, we train a generator that takes the class semantic knowledge as input and outputs several filters of class-specific semantic latent parts. By applying each part filter, our model can pay attention to corresponding local regions containing each part. At the inference stage, the classification is conducted by comparing the similarities between those parts. Experiments on several FSL benchmarks demonstrate the effectiveness of our proposed method and show its potential to go beyond class recognition to class understanding. Furthermore, we also find when semantic knowledge is more visualized and customized, it will be more helpful in the FSL task.

1. Introduction

It is challenging to learn novel categories well from a limited number of training samples because the success of today's deep learning systems significantly depends on the size of the training set [30]. On the contrary, humans can learn new categories rapidly even with very few training samples [5]. This inspires the possibility of few-shot learning (FSL). In addition, the ability of FSL is essential for intelligent agents to actively learn in the open visual world [2].

There are two key challenges of FSL which are insufficient comprehension and imperfect modeling of the novel class. The first challenge is intuitive since the limited training samples lead to inadequate visual comprehension of the novel class. In this case, semantic knowledge (which is information from other modalities) can be rather helpful in FSL [38, 50, 55]. Furthermore, we argue that semantic knowledge is actually indispensable in FSL because of the ambiguity when representing a class by very few samples. For example, assuming that a novel class has only one support image as shown in the top-left picture of Fig.1(a), even humans tend to be confused about whether this class is 'hat' or 'fennec fox' or 'fox'. Thus, it is necessary to use semantic knowledge to replenish the definition of the novel class.

Besides, existing semantic-using FSL methods still face the second challenge which is imperfect modeling of the novel class. The reason is that most FSL methods represent each class by a single global prototype which is extremely unstable and imprecise caused of its large variance of posture, environment, illumination, occlusion, and so on. As shown in Fig.1(a), both intra-class variation and inter-class confusion are exacerbated in the few-shot scenario. Therefore, a single global representation is not enough for FSL, and more precise local information contained in semantic parts is necessary. As a result, we propose to represent each class by its several different parts with the help of class semantic knowledge. Compared with the large intra-class variations of a single global class representation, each part has fewer dimensions of variation, so often fewer support images are enough to represent each part of the class. Besides smaller intra-class variations, representing class by its parts can also obtain better inter-class discrimination. Fig.1(b) shows the example, by comparing the discriminative semantic parts (such as ears, snout, leg, etc.), two overall similar classes can be better told apart. In summary,



(a) A single global class representation is unstable and easily confused in FSL. (b) Representing a class by its different parts is more stable and precise.

Figure 1: The motivation of our latent parts embedding. (a) Previous works using a single global embedding to represent a class will lead to large intra-class variations (unstable prototype caused by posture variance, occlusion, and scene changing which are sensible in FSL) and poor inter-class discrimination (easily confused between overall-similar classes when using imprecise global embedding as class representation). (b) Our method represents each class by its parts with the help of class semantic knowledge. The variations of each part are much smaller, which means our representation is more stable in FSL scenario. And our part-based classification is more precise, so overall-similar classes can be better told apart.

parts-based representation is more suitable for the FSL task.

In order to represent each class by its parts, the first step is to know what parts this class contains and then obtain parts embeddings. As shown in Fig.1(b), we first use semantic knowledge of each class to generate several class-specific part filters. It is worth noting that we can never pre-define parts for unknown novel classes, so here each filter corresponds to a class-specific latent part (ideally, 'large and pointed ears'). By using those filters to perform convolutional operations, different latent parts based on local regions (like the region of large ears) can be discovered. At last, we conduct the spatial reweight pooling operation to get the embedding of each part. Those latent parts embeddings (LPE) together form the class representation. In addition, we transfer part-level visual prior from base classes to refine these LPEs. This makes sense because for each novel class, different part tends to be similar to different base classes' part (e.g., the merlion's head resembles the lion's head while its tail is similar to the fish's). Therefore, a part-level prior transfer is more reasonable than the classic class-level transfer and we will verify its effectiveness later.

Then in the testing stage, we will compare the query with all novel classes one by one under each LPE representation so as to calculate the part-level similarity between the query and each novel class. The final score will be the weighted average of the part similarity scores. Experiments on several few-shot learning datasets not only demonstrate the effectiveness but also show its potential to go beyond class-level recognition to part-level understanding. Furthermore, by comparing the performance of different semantic sources (e.g., Word2Vec [29], CLIP semantic embeddings [31], and attributes), it can be concluded that more visualized and customized semantic knowledge is more useful in FSL.

2. Related Work

Few-shot learning. The introduction of FSL can be traced back to 2006 in [12]. This work proposed the basic approach to deal with FSL which is to learn the hard way of some base classes so as to facilitate the learning of few-shot novel classes. Different from the above work based on bag of visual words [39, 57], Matching Networks [44] is the first to adopt deep learning in FSL and has many follow-up works. From the perspective of how to transfer prior from base classes to novel classes, current methods can be divided into three main streams [18, 47]. The first one is data-based method whose aim is to generate sufficient training data for novel class [1,17,37]. The second one is optimization-based method where generalized initialization and efficient optimization strategies are designed, like MAML [14] and LSTM-based method [32]. The last one is metric-based method in which classification is performed according to the distance in feature space [16, 40, 41, 44]. Recently, there emerges some rethinking works of FSL like the task is unrealistic and too simple [6, 14], a good embedding is better than complicated meta-learning methods [42]. Similarly, in this paper we rethink that semantic knowledge is indispensable for FSL otherwise class definition will be ambiguous as mentioned above.

Semantic-using few-shot learning. In recent years, there has been a trend of using semantic knowledge to assist FSL. The inspiration for using semantics comes from a closely related topic, i.e., zero-shot learning (ZSL) [10, 21, 22]. The semantic knowledge source can be attributes [22], embeddings from pre-trained language models [27], knowledge bases [7], etc. In this paper, we will explore different semantic sources in our framework to find which semantic is more suitable for FSL. Different previous methods use semantic in different granularity, like task-level [9], class-level [8, 46, 50, 52], and part-level [55].

Part-based object understanding. Since objects are made by parts, part-based disentanglement is of vital importance for object understanding. In object detection, there are some classic part-based models like DPM [13] and its follow-up works [3, 28]. In these methods, all parts are well-defined. However, in FSL scenario, the large variety of categories causes the diversity of parts and we can never pre-define the parts for novel classes. So in our framework, we perform latent parts discovery instead of using explicit pre-defined parts. To that end, we use class semantic knowledge as the guide. Actually, many recent works in FSL are already focusing on the local representation [9, 25, 48, 56]. However, these methods ignore the importance of semantic knowledge that can be really helpful in parts discovery.

3. Approach

Fig.2 shows the framework of our proposed method which will be described in detail in the following subsections. We introduce the class-specific latent parts filters generation module in §3.1 and latent parts discovery module in §3.2. Then, §3.3 describes how to transfer latent parts representation from similar base classes. After that, §3.4 demonstrates part-based classification pipeline of the query image. At last, we describe the training strategy and loss functions of our framework in §3.5.

Problem Formulation. The target of FSL is to learn how to learn novel classes based on *M* base classes (denoted as $\mathcal{Y}^b$). A typical testing protocol is the *N*-Way, *K*-Shot setting, which means there are *N* novel classes (denoted as $\mathcal{Y}^n$) in each few-shot learning task where the base classes and novel classes are disjoint, i.e., $\mathcal{Y}^b \cap \mathcal{Y}^n = \emptyset$. We use the index $\{1, ..., M\}$ to represent the base classes and $\{M+1, ..., M+N\}$ to represent the novel classes. The base classes dataset (denoted as $\mathcal{D}^{base}$) has plenty of samples per class, while the novel class dataset named support set (denoted as $\mathcal{D}^{novel}$) has only *K* labeled samples per class. As we can see, $\mathcal{D}^{novel} = \{(\boldsymbol{x}_i, y_i) \mid \boldsymbol{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}^n\}_{i=1}^{N\times K}$. $\mathcal{X} \subseteq \mathbb{R}^{d_v \times H \times W}$ denotes the d_v dimension visual space which keeps the spatial information of visual feature map. Apart from visual space, we leverage semantic knowledge $\mathcal{S} = \{\boldsymbol{s}^c \in \mathbb{R}^{d_s}\}_{c=1}^{M+N}$ of both base and novel classes like other works, where the d_s is the dimension of semantic space. And in this paper, we try to adopt different semantic knowledge as the source. At last, the goal of FSL is to learn the classifiers for novel classes $f_{fsl} : \mathcal{X} \to \mathcal{Y}^n$.

3.1. Class-specific Latent Parts Filters Generation

Here we introduce the convolutional filter generators, each of which corresponds to a class-specific latent part of this class. As shown in the yellow-background region in Fig.2, this module generates *P* convolutional filters independently based on class semantic knowledge. These filters will be used for latent parts discovery in the next step.

Concretely, the input of this module is the class semantic vector $\boldsymbol{s}^c \in \mathcal{S}$. And the module outputs are *P* convolutional filters. As shown in Fig.2, there are *P* different MLPs: $\{\phi_p : \mathbb{R}^{d_s} \to \mathbb{R}^{d_v \times 1 \times 1}\}_{p=1}^{P}$. Every MLP independently maps the class semantic vector from semantic space to a 1×1 convolutional filter in visual space. Take the *p*-th MLP ϕ_p as the example:

$$\mathbf{filter}_p^c = \phi_p(\boldsymbol{s}^c). \tag{1}$$

Thus $\mathbf{filter}_p^c \in \mathbb{R}^{d_v \times 1 \times 1}$ is the 1×1 convolutional filter corresponding to the *p*-th latent part of the class *c*. Similarly, we can get *P* convolutional filters for each class. In the following subsections, we will conduct the latent parts discovery on the visual feature map based on these generated class-specific latent parts filters.

3.2. Latent Parts Discovery

After class-specific latent parts filters generation, each class has *P* convolutional filters [$\mathbf{filter}_1^c$, ..., $\mathbf{filter}_P^c$]. Now we use these filters to perform latent parts discovery on spatial feature maps of the support set images.

As shown in the green-background region in Fig.2, every filter will be used to perform a convolutional operation on spatial feature map $\boldsymbol{x}^c \in \mathbb{R}^{d_v \times H \times W}$ generated by feature extractor (without the last global pooling layer) and get the spatial activation map:

$$\boldsymbol{a}_p^c(\boldsymbol{x}^c) = \mathrm{sigmoid}(\boldsymbol{x}^c \odot \mathbf{filter}_p^c), \tag{2}$$

where $(\boldsymbol{x}^c, c) \in \mathcal{D}_c^n$ is one support sample of class *c*, $\mathcal{D}_c^n \subset \mathcal{D}^{novel}$ is the subset of the support set which contains the samples belonging to class *c*, $\mathbf{filter}_p^c \in \mathbb{R}^{d_v \times 1 \times 1}$ is the *p*-th 1×1 filter of class *c*, and $\odot$ denotes the convolutional operation. Therefore, each value in the spatial activation map $\boldsymbol{a}_p^c(\boldsymbol{x}^c) \in \mathbb{R}^{H \times W}$ represents how likely this local region contains the corresponding latent part of this class. It is worth noting that the last operation is a sigmoid function, thus the activation value is bounded between [0,1].

After the above convolutional operation, for each support image we get *P* spatial activation maps corresponding to *P* latent parts of this class. Then we use these spatial activation maps to perform region-based attention and weighted average pooling on the original spatial feature map. We use the activation values as the pooling weights. Therefore, we can get *P* latent parts embeddings. Since the weighted average pooling is based on region attention, we call this process

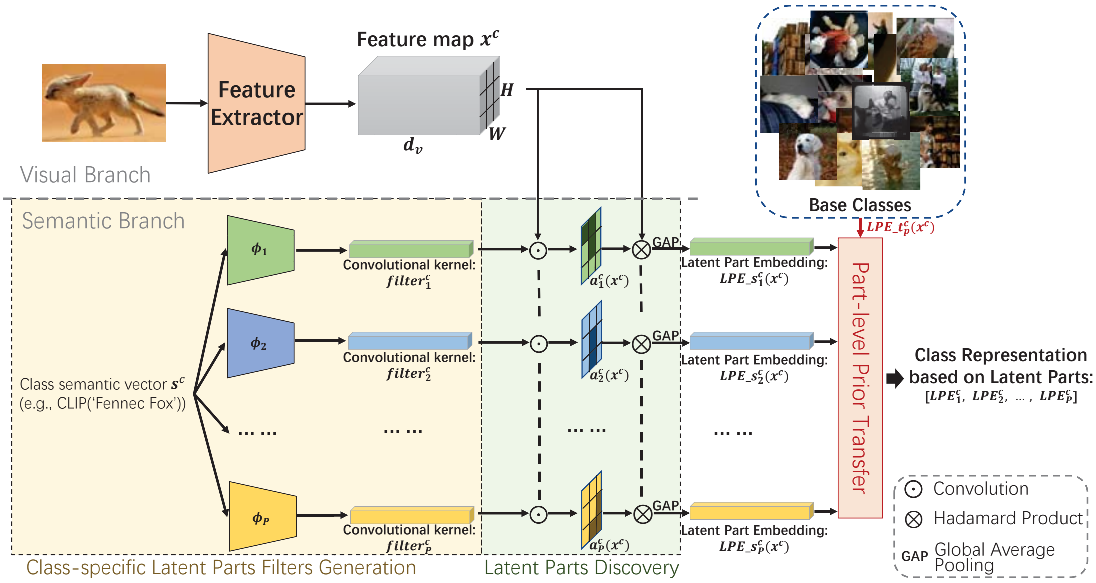

Feature map x^c
Feature Extractor
H
W
d_v
Visual Branch
Semantic Branch
Class semantic vector s^c (e.g., CLIP('Fennec Fox'))
ϕ_1
ϕ_2
ϕ_P
Convolutional kernel: $filter_1^c$
Convolutional kernel: $filter_2^c$
Convolutional kernel: $filter_P^c$
$a_1^c(x^c)$
$a_2^c(x^c)$
$a_P^c(x^c)$
GAP
Latent Part Embedding: $LPE\_s_1^c(x^c)$
Latent Part Embedding: $LPE\_s_2^c(x^c)$
Latent Part Embedding: $LPE\_s_P^c(x^c)$
Base Classes
$LPE\_t_p^c(x^c)$
Part-level Prior Transfer
Class Representation based on Latent Parts: $[LPE_1^c, LPE_2^c, \ldots, LPE_P^c]$
⊙ Convolution
⊗ Hadamard Product
GAP Global Average Pooling
Class-specific Latent Parts Filters Generation
Latent Parts Discovery

Figure 2: The framework of our proposed latent parts embeddings method. It contains three steps to obtain the final class representation. (1) Generate P convolutional filters from class semantic knowledge. Each filter corresponds to one latent part of the class. (2) Utilize these filters to perform latent parts discovery by spatial reweight pooling operation and get P latent parts embeddings. (3) Transfer part-level visual prior from base classes to the novel class so as to refine the final latent parts embeddings. At last, these latent parts embeddings together form the class representation of the current novel class.

latent parts discovery. Take the p-th latent part embedding derived from support image as an example:

$$\mathbf{LPE\_s}_p^c(\boldsymbol{x}^c) = \mathrm{GAP}(\boldsymbol{x}^c \otimes \boldsymbol{a}_p^c(\boldsymbol{x}^c)), \tag{3}$$

where $\otimes$ denotes Hadamard product (i.e., element-wise product), and GAP is the global average pooling. In other word, we perform a spatial reweight pooling operation on $\boldsymbol{x}^c$ to get the latent part embedding $\mathbf{LPE\_s}_p^c(\boldsymbol{x}^c) \in \mathbb{R}^{d_v}$.

3.3. Part-level Prior Transfer from Base Classes

So far, we get P LPEs for each support image $\boldsymbol{x}^c$. However, the representation is still facing the unstable problem caused by too few labeled samples. Therefore, in this subsection, we try to explicitly transfer visual prior from base classes to novel classes, so the LPEs can be more stable and precise. The most interesting thing here is that we perform part-level transfer instead of the classic class-level prototype transfer. Actually, it makes more sense to transfer prior knowledge at the part level since the similarity between two categories is always at the part level (e.g., merlion and lion are similar in the head part while merlion and fish are similar in the tail part).

Unlike previous works using one classification weight for each base class, our framework has P classification weights that correspond to P latent parts of each base class. In other words, we also use latent parts embeddings to represent base classes. Thus, the classification weight for base class j is $\boldsymbol{W}^j = [W_1^j, ..., W_P^j] \in \mathbb{R}^{d_v \times P}$, where $W_p^j \in \mathbb{R}^{d_v}$ is the weight corresponding to the p-th LPE. Now we can transfer visual prior from classification weights of base classes $\boldsymbol{W}_{base} = \{\boldsymbol{W}^j\}_{j=1}^M$ to class c:

$$\mathbf{LPE\_t}_p^c(\boldsymbol{x}^c) = \sum_{j \in \mathcal{Y}^b} \cos\left(\boldsymbol{\psi_p} \cdot \mathbf{LPE\_s}_p^c(\boldsymbol{x}^c), \boldsymbol{k}_p^j\right) \cdot \boldsymbol{W}_p^j, \tag{4}$$

where $\boldsymbol{\psi_p} \in \mathbb{R}^{d_v \times d_v}$ is a learnable matrix corresponding to the p-th latent part, $\{\boldsymbol{k}_p^j \in \mathbb{R}^{d_v}\}_{j=1}^M$ are M learnable keys corresponding to the p-th latent part, and $\boldsymbol{\psi_p}$ transform the p-th latent part embedding $\mathbf{LPE\_s}_p^c(\boldsymbol{x}^c)$ to a query vector, which will be used to perform cosine similarity calculation with $\boldsymbol{k}_p^j$ to determine how much of this base class's LPE $\boldsymbol{W}_p^j$ should be transferred. By transferring visual prior knowledge from base classes, we model the final LPE of $\boldsymbol{x}^c$ as the combination of $\mathbf{LPE\_s}_p^c(\boldsymbol{x}^c)$ and $\mathbf{LPE\_t}_p^c(\boldsymbol{x}^c)$:

$$\mathbf{LPE}_p(\boldsymbol{x}^c) = \lambda_1 \times \mathbf{LPE\_s}_p^c(\boldsymbol{x}^c) + \lambda_2 \times \mathbf{LPE\_t}_p^c(\boldsymbol{x}^c), \tag{5}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}$ are learnable coefficients.

Note that in N-Way K-Shot setting, each novel class has K support samples, here we average all K LPEs along the shot dimension to obtain the final LPEs of the novel class c:

$$\mathbf{LPE}_p^c = \frac{1}{|\mathcal{D}_c^n|} \sum_{(\boldsymbol{x^c}, c) \in \mathcal{D}_c^n} \mathbf{LPE}_p(\boldsymbol{x^c}), \tag{6}$$

where $|\mathcal{D}_c^n| = K$. At last, we obtain the final LPE of novel class c: $\mathbf{LPE}^c = [\mathbf{LPE}_1^c, ..., \mathbf{LPE}_P^c] \in \mathbb{R}^{d_v \times P}$.

3.4. Part-based Query Classification

Based on these final latent parts embeddings of novel classes, now we can perform the few-shot classification.

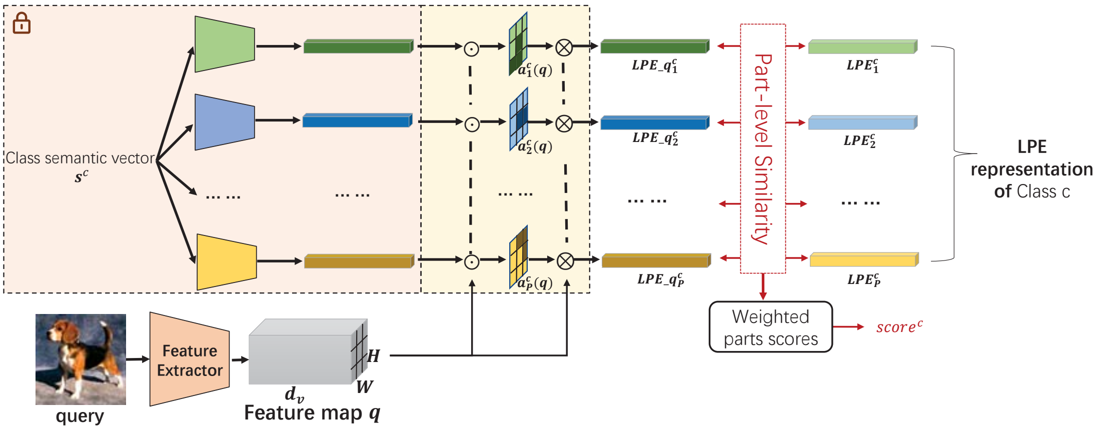


Figure 3: The classification pipeline based on our latent parts embedding. To obtain classification scores, we assume the query to be novel class 1 to N respectively (in this figure we only demonstrate one class c), and calculate the part-level similarity under each novel class's LPE representation. The final score is the weighted average of the part similarity scores.

As shown in Fig.3, for query sample $\boldsymbol{q} \in \mathbb{R}^{d_v \times H \times W}$, our framework will compare the query with N novel classes one by one under each LPE representation so as to calculate the similarity between query $\boldsymbol{q}$ and each novel class. Concretely, to verify whether the query belongs to class c, we first calculate the latent parts activation maps $\{\boldsymbol{a}_p^c(\boldsymbol{q})\}_{p=1}^P$ by performing P convolutional filters of class c to query $\boldsymbol{q}$ just like the process mentioned in 3.2:

$$\boldsymbol{a}_p^c(\boldsymbol{q}) = \text{sigmoid}(\boldsymbol{q} \odot \textbf{filter}_p^c). \tag{7}$$

Based on these spatial activation maps, P latent parts embeddings of query $\mathbf{LPE\_q}^c = [\mathbf{LPE\_q}_1^c, ..., \mathbf{LPE\_q}_P^c]$ can be calculated as:

$$\mathbf{LPE\_q}_p^c = \text{GAP}(\boldsymbol{q} \otimes \boldsymbol{a}_p^c(\boldsymbol{q})). \tag{8}$$

Then based on the query's LPE $\mathbf{LPE\_q}_p^c$ and the LPE of novel class c $\mathbf{LPE}_p^c$, the cosine similarity for p-th latent parts embedding can be calculated as

$$\text{score}_p^c(\boldsymbol{q}) = \cos \left\langle \mathbf{LPE\_q}_p^c, \mathbf{LPE}_p^c \right\rangle. \tag{9}$$

We model the final similarity score as the weighted sum of P LPE similarities:

$$\text{score}^c(\boldsymbol{q}) = \frac{1}{\sum_{p=1}^P \text{weight}_p^c} \sum_{p=1}^P \text{weight}_p^c \cdot \text{score}_p^c(\boldsymbol{q}), \tag{10}$$

where the weight coefficients $\textbf{weight}^c = [\text{weight}_1^c, ..., \text{weight}_P^c] \in \mathbb{R}^P$ are generated by a learnable MLP $g : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^P$ which takes the semantic vector as input and output the weight coefficients:

$$\textbf{weight}^c = g(\boldsymbol{s}^c). \tag{11}$$

It is worth noting that MLP g is designed to leverage the class semantic knowledge to learn the importance of each latent part with respect to each novel class.

3.5. Training Strategy and Loss Functions

Unlike most previous FSL works containing two training stages, we conduct a one-stage end-to-end training by the meta-learning strategy. There are three loss functions. The first is the loss on base classes which contains 2 parts, one is standard cross-entropy loss while the other is LPE-based cross-entropy loss (the objective of the second term here is to learn the part-level base prior $\boldsymbol{W}_{base}$ mentioned above):

$$\begin{aligned}\mathcal{L}_{base} = &- \log \frac{\exp\left(\overline{\mathbf{W}}^{i\top} \boldsymbol{q}' + \overline{\mathrm{b}}^{i}\right)}{\sum_{j=1}^{|\mathcal{Y}^b|} \exp\left(\overline{\mathbf{W}}^{j\top} \boldsymbol{q}' + \overline{\mathrm{b}}^{j}\right)} \\ &- \log \frac{\exp\left(\text{score}^i(\boldsymbol{q}')/\tau\right)}{\sum_{j=1}^{|\mathcal{Y}^b|} \exp\left(\text{score}^j(\boldsymbol{q}')/\tau\right)},\end{aligned} \tag{12}$$

where ($\boldsymbol{q}'$,i) is one of the base query samples, $\overline{\mathbf{W}}^j$ and $\overline{\mathrm{b}}^j$ are standard base classification weight and bias for class j respectively, τ is the scalable coefficient for cosine similarity making it more suitable for cross entropy calculation.

The second loss is few-shot classification loss corresponding to the classification process mentioned above:

$$\mathcal{L}_{fsl} = -\log \frac{\exp\left(\text{score}^c(\boldsymbol{q})/\tau\right)}{\sum_{c'=1}^N \exp\left(\text{score}^{c'}(\boldsymbol{q})/\tau\right)}, \tag{13}$$

where ($\boldsymbol{q}$,c) is the query sample of the fake novel class sampled from base classes to simulate the few-shot scenario (no real novel samples are used since this is in meta-training).

The third loss is a divergent loss which is introduced for learning different P latent parts:

$$\mathcal{L}_{div} = \sum_{c=1}^N \sum_{i=1}^P \sum_{j=1, j\neq i}^P \frac{< \mathbf{LPE}_i^c, \mathbf{LPE}_j^c >}{\left\|\mathbf{LPE}_i^c\right\|_2 \left\|\mathbf{LPE}_j^c\right\|_2}, \tag{14}$$

We model the final loss function as the combination of these three losses by coefficient λ and λ_{div}:

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \lambda \mathcal{L}_{fsl} + \lambda_{div} \mathcal{L}_{div}. \tag{15}$$

4. Experiments

In this section, we first introduce the experiment setting, then verify the effectiveness of our proposed method, and then give some visualization results of our methods, followed by benchmark comparisons.

4.1. Datasets and Settings

Datasets. We conduct our experiments on 4 widely used FSL benchmarks, i.e., miniImageNet [44], tieredImageNet [34], CIFAR-FS [4], and CUB [45]. MiniImageNet and tieredImageNet are derivatives of ImageNet dataset [36], CIFAR-FS is derived from CIFAR-100 dataset [20,43]. The summary can be found in the supplementary material.

Semantic knowledge source. As for benchmarks without semantic knowledge annotations (e.g., class-aware attributes annotations) such as miniImageNet, tieredImageNet, and CIFAR-FS, previous works always leverage pre-trained Word2Vec models such as GloVe [29] as the semantic source. In this paper, we take a further step and try to leverage more visualized and customized semantic knowledge source like the semantic encoder of CLIP [31]. The dimension of GloVe vectors is 300 and the dimension of CLIP semantic embedding is 512. It is worth noting that, to avoid unfair comparison, only the pre-trained semantic encoder of CLIP will be used in this paper, and CLIP visual encoder will not be used. Given that CLIP is trained to align visual and semantic space, the semantic encoder of CLIP is accurately a more visualized semantic knowledge source. As for benchmarks with semantic annotations such as CUB, the customized attributes annotations which have 312 dimensions can be used as the semantic knowledge.

Implementation details. We implement our code using PyTorch framework[1]. Following most previous works [8,16,24,26,35,48], we utilize a ResNet-12 as our backbone for all datasets. We also change the number of filters from [64,128,256,512] to [64,160,320,640] same as most of previous works [19, 23, 33, 42]. The class-specific latent parts filter generators are P MLPs, with 2 fully connected layers and the LeakyReLU nonlinearity layer between them. Network g designed for learning importance for each part is an MLP too, with 2 fully connected layers and the LeakyReLU nonlinearity layer between them, followed by sigmoid nonlinearity. Inspired by [11], we use Z-Score as the normalization of feature representation. Other parameters such as λ_1, λ_2, and temperature t are tuned during end-to-end training. More details can be found in the supplementary material.

4.2. Effectiveness of the Proposed Method

To demonstrate the effectiveness of our proposed method, we verify each part of our framework in the order of the pipeline, including performance comparison when using different semantic sources, the effectiveness of LPE representation, the effectiveness of prior knowledge transfer, and the influence of hyperparameters such as P and λ.

(1) The effectiveness of different semantic knowledge sources. Since our method tries to leverage semantic knowledge to guide the latent parts discovery which is a more difficult task than other semantic using methods, the robustness of semantic knowledge is rather important. Therefore, apart from commonly-used Word2Vec, we explore more visualized and customized knowledge source CLIP semantic which established the alignment between visual space and semantic space. Tab.1 shows the comparison results on miniImageNet when using different semantic sources to guide the LPE representation. As we can see, both the result of using CLIP semantic (the 1st row of Tab.1) and using GloVe (the 2nd row of Tab.1) significantly outperform the result of no semantic baseline (the 3rd row of Tab.1), which shows the effectiveness of semantic using. In addition, by comparing the result of CLIP and GloVe, using CLIP as the semantic source outperform GloVe which means more visualized semantic is more powerful in FSL.

Table 1: Comparison result on miniImageNet when using different semantic knowledge sources.

Semantic source	5-Way 1-Shot	10-Way 1-Shot
CLIP semantic	**71.64**±0.40	**53.20**±0.28
GloVe	68.28±0.43	50.06±0.28
no semantic	65.57±0.44	48.64±0.29

The experiments above demonstrate CLIP semantic embedding is better than GloVe word embedding, now we compare the results of CLIP semantic embedding and customized annotations on CUB where every bird class in this dataset has precise attribute annotations. As shown in Tab.2, the performance when using attribute annotations is better than CLIP. The reason is that coarse-grained CLIP semantic embeddings will not work well in the fine-grained setting. Attribute annotation is more customized semantic knowledge of CUB classes so it outperform the CLIP semantic.

Table 2: Comparison result on CUB when using different semantic knowledge sources.

Semantic source	5-Way 1-Shot	10-Way 1-Shot
CUB attributes annotations	**85.04**±0.34	**77.74**±0.27
CLIP semantic	80.76±0.40	67.70±0.33
no semantic	77.35±0.44	64.91±0.35

To sum up, from the results on miniImageNet and CUB (i.e., annotations $>$ CLIP $>$ GloVe), we can draw the conclusion that when the semantic knowledge is more visualized and customized it will help more in FSL.

(2) The effectiveness of LPE representation. As shown in Tab.3, when we set number of latent parts P to 1, there is a significant decline in FSL performance compared with $P = 5$. Note that when $P = 1$, it degenerates to the global

[1] The codes are available at both http://vipl.ict.ac.cn/zygx/dm/ and https://github.com/MartaYang/LPE

class embedding. This ablation result shows that latent parts embedding is better than global class embedding so as to demonstrates the effectiveness of our key module.

Table 3: Ablation study of our proposed LPE representation on miniImageNet and CUB.

Models	miniImageNet	CUB
	5-Way 1-Shot	5-Way 1-Shot
5 LPEs (i.e., P=5) **(ours)**	**71.64**±0.40	**85.04**±0.34
1 LPE (i.e., P=1) (ablation)	64.03±0.46	76.95±0.44

(3) The effectiveness of visual prior transfer from base classes. As shown in Tab.4, there is a decline in FSL performance if don't perform the transfer, which shows the importance of the visual prior knowledge transfer from base classes. It is also worth noting that the effectiveness comes from the more human-like transfer mechanism. As mentioned above, it makes more sense to transfer visual prior based on part-level instead of class-level.

Table 4: Ablation study of our proposed part-level visual prior transfer from base classes on miniImageNet and CUB.

Models	miniImageNet	CUB
	5-Way 1-Shot	5-Way 1-Shot
w/ transfer **(ours)**	**71.64**±0.40	**85.04**±0.34
w/o transfer (ablation)	64.33±0.46	77.39±0.45

(4) The influence of the latent parts number *P*. Fig.4 gives the 5-Way 1-Shot accuracy of different *P* on validation set of miniImageNet and CIFAR-FS. The best performance is achieved when $P = 5$, so we set $P = 5$ for testing. The accuracy rises with the growth of *P* since more latent parts can offer more precise modeling of novel classes. However, after reaching the peak at $P = 5$, the accuracy presents a declining trend when *P* further increases. The reason is that parts of interest for a class are always limited. Too many parts may cause redundancy and even bring in noise.

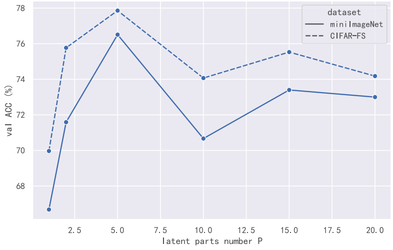


Figure 4: The effect of the number of latent parts *P* on the validation set of miniImageNet and CIFAR-FS.

(5) The influence of the loss weight coefficient λ. Fig.5 shows 5-Way 1-Shot results when setting different weight coefficients λ on the validation set of miniImageNet and CIFAR-FS. From Eq.15, a larger λ means more weight on few-shot classification loss. As we can see, when λ is too small, the few-shot classification loss is suppressed by standard cross-entropy loss on base classes, making the performance remains the same as baseline. And with the growth of λ, the accuracy presents a rising trend, because the LPE representation can be trained more sufficiently. In addition, after reaching the peak at $\lambda = 2.0$, the accuracy slightly drops as λ increases. This is because that cross entropy loss on base classes is also essential for feature space training so we set $\lambda = 2.0$ for testing as the balance of these losses.

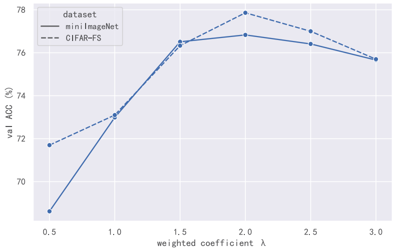


Figure 5: The effect of few-shot loss weight coefficient λ on the validation set of miniImageNet and CIFAR-FS.

4.3. Dive Deep into Latent Parts

In order to explore what exactly novel classes' latent parts are, we visualize the activation map $\boldsymbol{a}_p^c$ as shown in Fig.6. Results in different columns correspond to different latent parts of the corresponding novel class. Firstly, as we can see *P* activation local regions are different for the same support image, which demonstrates different latent parts indeed capture different aspects of the category. Secondly, the visualization results show that similar parts are highlighted in the same columns (e.g., the breast part of different birds are activated in the 1st column of Fig.6, and the head and tail parts are activated in the 3rd column of Fig.6). This phenomenon demonstrates that the same latent parts filter generator tends to discover similar parts or attributes. This shows the potential of our model to align with real semantic parts and the potential of part-based class understanding.

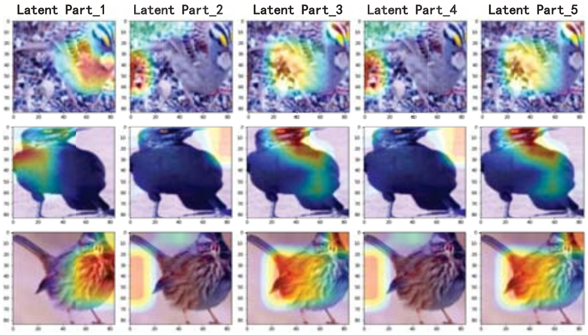


Figure 6: Visualization results of the activation regions of *P* (=5) latent parts on novel classes of CUB. The redder region means higher activation value.

4.4. Benchmark Comparisons and Evaluations

After verifying the effectiveness of the proposed methods, in this subsection we compare our method with other SOTA FSL methods. Tab.5 shows the results on miniImageNet and tieredImageNet dataset. Note that TriNet,

Table 5: Comparisons with popular FSL approaches in average classification accuracies (%) on miniImageNet and tieredImageNet. We report the average classification accuracies with 95% confidence intervals. "Sem." denotes whether to leverage semantic knowledge.

Models	Backbone	Sem.	miniImageNet		tieredImageNet	
			5-Way 1-Shot	5-Way 5-Shot	5-Way 1-Shot	5-Way 5-Shot
Matching Networks (NIPS'16) [44]	4Conv	No	43.56±0.84	55.31±0.73	-	-
MAML (ICML'17) [14]	4Conv	No	48.70±1.84	63.11±0.92	51.67±1.81	70.30±1.75
ProtoNet (NIPS'17) [40]	4Conv	No	49.42±0.78	68.20±0.66	53.31±0.89	72.69±0.74
Dynamic-FSL (CVPR'18) [15]	4Conv	No	56.20±0.86	72.81±0.62	-	-
wDAE-GNN (CVPR'19) [16]	WRN-28-10	No	61.07±0.15	76.75±0.11	68.18±0.16	83.09±0.12
MetaOptNet (CVPR'19) [23]	ResNet-12	No	62.64±0.61	78.63±0.46	65.99±0.72	81.56±0.53
DeepEMD (CVPR'20) [56]	ResNet-12	No	65.91±0.82	82.41±0.56	71.16±0.87	86.03±0.58
RFS (ECCV'20) [42]	ResNet-12	No	64.82±0.60	82.14±0.43	71.52±0.69	86.03±0.49
Neg-Cosine (ECCV'20) [26]	ResNet-12	No	63.85±0.81	81.57±0.56	-	-
ODE (CVPR'21) [51]	ResNet-12	No	67.76±0.46	82.71±0.31	71.89±0.52	85.96±0.35
IEPT+ZN (ICCV'21) [11]	ResNet-12	No	67.35±0.43	83.04±0.29	72.28±0.51	**87.20**±0.34
TPMN (ICCV'21) [48]	ResNet-12	No	67.64±0.63	83.44±0.43	72.24±0.70	86.55±0.63
DeepBDC (CVPR'22) [49]	ResNet-12	No	67.83±0.43	**85.45**±0.29	73.82±0.47	89.00±0.30
TriNet (TIP'19) [8]	ResNet-18	Yes	58.12±1.37	76.92±0.69	-	-
AM3 (NIPS'19) [50]	ResNet-12	Yes	65.30±0.49	78.10±0.36	69.08±0.47	82.58±0.31
LPE-GloVe **(ours)**	ResNet-12	Yes	68.28±0.43	78.88±0.33	72.03±0.49	83.76±0.37
LPE-CLIP semantic **(ours)**	ResNet-12	Yes	**71.64**±0.40	79.67±0.32	**73.88**±0.48	84.88±0.36

Table 6: CIFAR-FS results.Test setting is the same as above.

Models	CIFAR-FS	
	5-Way 1-Shot	5-Way 5-Shot
MAML (ICML'17) [14]	58.9±1.9	71.5±1.0
ProtoNet (NIPS'17) [40]	55.5±0.7	72.0±0.6
MetaOptNet (CVPR'19) [23]	72.0±0.7	84.2±0.5
RFS (ECCV'20) [42]	73.9±0.8	86.9±0.5
TPMN (ICCV'21) [48]	75.5±0.9	**87.2**±0.6
LPE-GloVe **(ours)**	74.88±0.45	85.30±0.35
LPE-CLIP semantic **(ours)**	**80.62**±0.41	86.22±0.33

Table 7: Results on CUB. Test setting is the same as above.

Models	CUB	
	5-Way 1-Shot	5-Way 5-Shot
TriNet (TIP'19) [8]	69.61±0.46	84.10±0.35
MultiSem (CoRR'19) [38]	76.1	82.9
FEAT (CVPR'20) [54]	68.87±0.22	82.90±0.15
DeepEMD (CVPR'20) [56]	75.65±0.83	88.69±0.50
VS-Align (ICMR'21) [52]	77.03±0.85	87.20±0.70
IEPT+ZN (ICCV'21) [11]	73.54±0.48	87.82±0.30
LPE-CLIP semantic **(ours)**	80.76±0.40	88.98±0.26
LPE-attributes **(ours)**	**85.04**±0.34	**89.24**±0.26

AM3, and our method leverage semantic knowledge while other methods do not leverage semantic knowledge. As we can see, our method outperforms other semantic using methods and achieves the highest performance especially in the 5-Way 1-Shot setting. It is also worth noting that by the help of semantic knowledge, our method outperforms TPMN [48] which also adopts part-level representation but in unimodal setting. In addition, as shown in Tab.6, our method also gets competitive results on CIFAR-FS.

Like many other semantic-using FSL methods [38, 50, 53], the performance gain derived from semantic will decline when the number of shots gets larger because the visual embedding itself gets more stable and accurate when there is more visual information. As the saying goes "a picture is worth a thousand words", the assistance from semantic knowledge will drop down in the larger shot scenario. However, as shown in Tab.7 when using more customized semantic knowledge (e.g., attributes annotation in CUB) our methods can still have an advantage in larger shot scenarios.

5. Conclusion

In this work, we propose to represent a class as the combination of several latent parts embeddings (LPE) with the help of class semantic knowledge. Each part has fewer variations and can be more easily represented by fewer samples, and the classification based on parts is more accurate, so LPE is more suitable for the FSL task. In addition, we propose to transfer part-level visual prior from base classes to novel classes which makes more sense since the similarity between the two categories is actually at the part level. From extensive experiences, we find out that (a) semantic knowledge is indispensable for replenishing the definition of the novel class otherwise FSL task will somewhat be ambiguous because of limited training samples, (b) the more visualized and customized semantic source is more useful in FSL, and (c) our method has potential for real semantic parts discovery in FSL which is a vital step from class-level object recognition to part-level object understanding.

Acknowledgements. This work is partially supported by National Key R&D Program of China No. 2021ZD0111901, and Natural Science Foundation of China under contracts Nos. U21B2025, U19B2036, 61922080.

References

[1] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In *European Conference on Computer Vision (ECCV)*, pages 18–35, 2020.

[2] Ali Ayub and Alan R Wagner. Tell me what this is: few-shot incremental object learning by a robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8344–8350. IEEE, 2020.

[3] Hossein Azizpour and Ivan Laptev. Object detection using strongly-supervised deformable part models. In *European Conference on Computer Vision (ECCV)*, pages 836–849. Springer, 2012.

[4] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations (ICLR)*, 2019.

[5] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

[6] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations (ICLR)*, 2019.

[7] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Neil: Extracting visual knowledge from web data. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1409–1416, 2013.

[8] Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. Multi-level semantic feature augmentation for one-shot learning. *IEEE Transactions on Image Processing (TIP)*, 28(9):4594–4605, 2019.

[9] Chuanqi Dong, Wenbin Li, Jing Huo, Zheng Gu, and Yang Gao. Learning task-aware local representations for few-shot learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 716–722, 2021.

[10] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1778–1785, 2009.

[11] Nanyi Fei, Yizhao Gao, Zhiwu Lu, and Tao Xiang. Z-score normalization, hubness, and few-shot learning. In *IEEE International Conference on Computer Vision (ICCV)*, pages 142–151, 2021.

[12] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(4):594–611, 2006.

[13] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(9):1627–1645, 2010.

[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, pages 1126–1135, 2017.

[15] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4367–4375, 2018.

[16] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21–30, 2019.

[17] Jiechao Guan, Zhiwu Lu, Tao Xiang, Aoxue Li, An Zhao, and Ji-Rong Wen. Zero and few shot learning with semantic feature synthesis and competitive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(7):2510–2523, 2020.

[18] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. Collect and select: Semantic alignment metric learning for few-shot learning. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8460–8469, 2019.

[19] Dahyun Kang, Heeseung Kwon, Juhong Min, and Minsu Cho. Relational embedding for few-shot classification. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8822–8833, 2021.

[20] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[21] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 951–958, 2009.

[22] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(3):453–465, 2013.

[23] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10657–10665, 2019.

[24] Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. Boosting few-shot learning with adaptive margin loss. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12576–12584, 2020.

[25] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7260–7268, 2019.

[26] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *European Conference on Computer Vision (ECCV)*, pages 438–455, 2020.

[27] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[28] Patrick Ott and Mark Everingham. Shared parts for deformable part-based models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1513–1520. IEEE, 2011.

[29] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[30] Guo-Jun Qi and Jiebo Luo. Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(4):2168–2187, 2020.

[31] Alec Radford, Jong Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[32] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[33] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *IEEE International Conference on Computer Vision (ICCV)*, pages 331–339, 2019.

[34] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations (ICLR)*, 2018.

[35] Mamshad Nayeem Rizve, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Exploring complementary strengths of invariant and equivariant representations for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10836–10846, 2021.

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[37] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8247–8255, 2019.

[38] Eli Schwartz, Leonid Karlinsky, Rogerio Feris, Raja Giryes, and Alex M Bronstein. Baby steps towards few-shot learning with multiple semantics. *arXiv preprint arXiv:1906.01905*, 2019.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[40] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4077–4087, 2017.

[41] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1199–1208, 2018.

[42] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *European Conference on Computer Vision (ECCV)*, pages 266–282, 2020.

[43] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(11):1958–1970, 2008.

[44] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3630–3638, 2016.

[45] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. 2011.

[46] Xin Wang, Fisher Yu, Ruth Wang, Trevor Darrell, and Joseph E Gonzalez. TAFE-Net: Task-aware feature embeddings for low shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1831–1840, 2019.

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

[48] Jiamin Wu, Tianzhu Zhang, Yongdong Zhang, and Feng Wu. Task-aware part mining network for few-shot learning. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8433–8442, 2021.

[49] Jiangtao Xie, Fei Long, Jiaming Lv, Qilong Wang, and Peihua Li. Joint distribution matters: Deep brownian distance covariance for few-shot classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7972–7981, 2022.

[50] Chen Xing, Negar Rostamzadeh, Boris Oreshkin, and Pedro O O Pinheiro. Adaptive cross-modal few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4847–4857, 2019.

[51] Chengming Xu, Yanwei Fu, Chen Liu, Chengjie Wang, Jilin Li, Feiyue Huang, Li Zhang, and Xiangyang Xue. Learning dynamic alignment via meta-filter for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5182–5191, 2021.

[52] Kun Yan, Zied Bouraoui, Ping Wang, Shoaib Jameel, and Steven Schockaert. Aligning visual prototypes with bert embeddings for few-shot learning. In *Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR)*, pages 367–375, 2021.

[53] Fengyuan Yang, Ruiping Wang, and Xilin Chen. SEGA: Semantic guided attention on visual prototype for few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1056–1066, 2022.

[54] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8808–8817, 2020.

[55] Baoquan Zhang, Xutao Li, Yunming Ye, Zhichao Huang, and Lisai Zhang. Prototype completion with primitive knowledge for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3754–3762, 2021.

[56] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12203–12213, 2020.

[57] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1):43–52, 2010.