Compositional Prototypical Networks for Few-Shot Classification

Qiang Lyu, Weiqiang Wang[*]

School of Computer Science and Technology, University of Chinese Academy of Sciences
luqiang20@mails.ucas.ac.cn, wqwang@ucas.ac.cn

Abstract

It is assumed that pre-training provides the feature extractor with strong class transferability and that high novel class generalization can be achieved by simply reusing the transferable feature extractor. In this work, our motivation is to explicitly learn some fine-grained and transferable meta-knowledge so that feature reusability can be further improved. Concretely, inspired by the fact that humans can use learned concepts or components to help them recognize novel classes, we propose Compositional Prototypical Networks (CPN) to learn a transferable prototype for each human-annotated attribute, which we call a component prototype. We empirically demonstrate that the learned component prototypes have good class transferability and can be reused to construct compositional prototypes for novel classes. Then a learnable weight generator is utilized to adaptively fuse the compositional and visual prototypes. Extensive experiments demonstrate that our method can achieve state-of-the-art results on different datasets and settings. The performance gains are especially remarkable in the 5-way 1-shot setting. The code is available at https://github.com/fikry102/CPN.

Introduction

Deep learning models have made tremendous progress in many computer vision tasks such as image classification (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016), object detection (Girshick et al. 2014; He et al. 2017), and semantic segmentation (Long, Shelhamer, and Darrell 2015). However, such models heavily rely on large-scale labeled data, which can be impractical or laborious to collect. Humans, by contrast, can easily learn to recognize novel classes from only one or a few examples by utilizing prior knowledge and experience. To bridge the gap between deep learning models and human intelligence, few-shot learning (Fei-Fei, Fergus, and Perona 2006; Miller, Matsakis, and Viola 2000) (FSL) has recently received much attention from researchers. Inspired by human learning behavior, FSL aims to recognize unlabeled examples from novel classes using only a few labeled examples, with prior knowledge learned from the disjoint base classes dataset containing abundant examples per class.

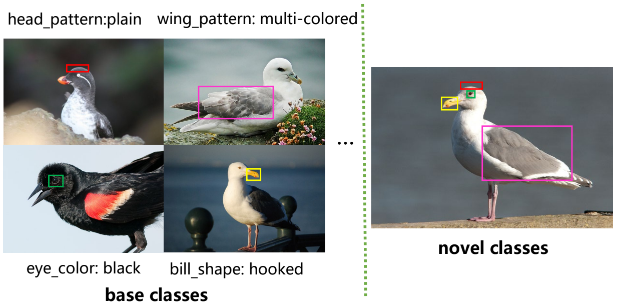


Figure 1: An illustration of the reusable components which can help humans recognize new birds better. We propose to learn component prototypes for human-annotated attributes from base classes and reuse them to construct compositional prototypes for novel classes.

Meta-learning (Schmidhuber 1987), or learning to learn, has become a popular learning paradigm for few-shot learning. Meta-learning based approaches aim to extract transferable meta-knowledge from collections of learning tasks so that the learned model can quickly adapt to new tasks using the learned meta-knowledge. A recent work (Raghu et al. 2020) has demonstrated that the reusable features, i.e., the high-quality features contained in the learned initial condition, are the dominant factor for the effectiveness of Model Agnostic Meta-Learning (Finn, Abbeel, and Levine 2017) (MAML), a well-known meta-learning method. In addition to MAML, a succession of meta-learning based approaches have been proposed to obtain various meta-knowledge, e.g., a good initialization (Ravi and Larochelle 2017; Finn, Abbeel, and Levine 2017), an optimization algorithm (Andrychowicz et al. 2016; Ravi and Larochelle 2017), or a meta-structure (Ye et al. 2020; Hou et al. 2019).

However, many recent works (Chen et al. 2021b, 2019a; Tian et al. 2020; Dhillon et al. 2020) indicate that pre-training a feature extractor on the whole base classes dataset can achieve comparable or even better performance than many existing meta-learning methods. Some researches (Wang, Pontil, and Ciliberto 2021; Chen et al. 2021b) show that global classification in the pre-training stage can provide the feature extractor with strong class transferability and therefore is theoretically beneficial to the subsequent

[*]Corresponding author.

meta-learning stage. Here, we argue that although high novel class generalization can be achieved by reusing the pre-trained feature extractor, there is still a huge gap compared to how humans reuse prior knowledge and experience. More specifically, humans can summarize some reusable concepts or components from past learning tasks and then draw connections between these concepts or components with new learning tasks to help them learn new things better. For instance, if a child has ever seen several birds, one with multi-colored wings, one with black eyes, one with a hooked beak, and so on, he can easily learn to recognize a new kind of bird with these features (see Figure 1).

Based on the above analysis, our motivation is to explicitly learn some fine-grained and transferable meta-knowledge from base classes and then reuse the learned meta-knowledge to recognize novel classes with only a few labeled examples. As some previous works (Huang et al. 2021; Tokmakov, Wang, and Hebert 2019; Ji et al. 2022; Xing et al. 2019; Chen et al. 2022), our work also utilizes **category-level** attribute annotations, *i.e.*, only one attribute score vector for each class. AM3 (Xing et al. 2019) proposes a modality mixture mechanism that can adaptively combine visual and semantic information. AGAM (Huang et al. 2021) proposes an attention alignment mechanism to align the self-guided branch's channel attention and spatial attention with the attributes-guided branch. MAP-Net (Ji et al. 2022) designs a Modal-Alternating Propagation Module (MAP-Module) to alleviate the information asymmetry problem between semantic-guided and nonsemantic-guided examples. However, these methods simply take the attribute score vector as a whole and do not pay enough attention to the more fine-grained attributes. Although the method proposed in (Tokmakov, Wang, and Hebert 2019) has considered fine-grained attributes, it aims to use a regularization technique to improve the compositionality of learned representations. By contrast, our work aims to learn transferable component prototypes for human-annotated attributes and then use these learned component prototypes to construct compositional prototypes for novel classes.

Related Work

Few-Shot Learning. Few-shot learning has been widely studied in recent years. Existing few-shot learning methods can be roughly classified into three branches: optimization-based methods, metric-based methods, and semantic-based methods. Our work belongs to both metric-based and semantic-based methods.

Optimization-based methods (Finn, Abbeel, and Levine 2017; Ravi and Larochelle 2017; Andrychowicz et al. 2016) aim to learn a good initialization or an optimization algorithm so that the model can quickly adapt to new tasks with only a few steps of gradient descent. MAML (Finn, Abbeel, and Levine 2017) combines a second-order optimizing strategy with the meta-learning framework. To overcome difficulties caused by direct optimization in high-dimensional parameter spaces, LEO (Rusu et al. 2019) performs meta-learning in a low-dimensional latent space from which high-dimensional parameters can be generated.

Metric-based methods aim to learn a good embedding function so that the embedded examples from novel classes can be correctly classified using a proper distance metric. For instance, Prototypical Networks (Snell, Swersky, and Zemel 2017) calculate the mean vector of the embedded examples of a given class as the prototype of this class and choose the Euclidean distance as its distance metric. Furthermore, Relation Networks (Sung et al. 2018) use a learnable distance metric, which can be jointly optimized with the embedding function. DeepEMD (Zhang et al. 2020) employs the Earth Mover's Distance (Rubner, Tomasi, and Guibas 2000) (EMD) as its distance metric and calculates the similarity between two images from the perspective of local features.

Unlike the methods that rely solely on visual information, semantic-based methods resort to auxiliary semantic information. For instance, AGAM (Huang et al. 2021) and MAP-Net (Ji et al. 2022) use category-level attribute score vectors. COMET (Cao, Brbic, and Leskovec 2021) uses the human-annotated location coordinates for predefined parts of birds to extract fine-grained features for these parts. By contrast, our method focuses on learning **reusable** component prototypes for predefined attributes. RS-FSL (Afham et al. 2021) replaces numerical class labels with category-level language descriptions. Prototype Completion Network (Zhang et al. 2021) utilizes diverse semantic information to learn to complete prototypes, namely, class parts/attributes extracted from WordNet and word embeddings calculated by GloVe (Pennington, Socher, and Manning 2014) of all categories and attributes. By contrast, our work only utilizes category-level attribute score vectors as auxiliary semantic information following AGAM (Huang et al. 2021) and MAP-Net (Ji et al. 2022),.

Zero-Shot Learning. Zero-shot learning (ZSL) also aims to classify examples from unseen classes. Compared to few-shot learning, there is only semantic information (e.g., attribute annotations or word embeddings) and no labeled images for unseen classes in ZSL. Therefore the key insight of ZSL is to transfer semantic knowledge from seen classes to unseen classes.

Early ZSL methods (Li et al. 2018) learn an embedding function from visual space to semantic space. Then the unlabeled examples for unseen classes can be projected into semantic space in which category-level attribute vectors reside. However, these methods train their models only using examples from seen classes, and the learned models are inevitably biased towards seen classes when it comes to the generalized ZSL setting. To reduce the bias problem, some researches (Verma et al. 2018; Xian et al. 2019) utilize generative models to generate images or visual features for unseen classes based on the category-level attribute vectors. Furthermore, some recent methods (Chen et al. 2021a; Schonfeld et al. 2019) learn to map visual and semantic features into a joint space. Our work is somewhat inspired by such methods since we adaptively fuse the semantic compositional prototype and the visual prototype.

Compositional Representations. It is considered that humans can harness compositionality to rapidly acquire and generalize knowledge to new tasks or situations in cognitive science literature (Hoffman and Richards 1984; Biederman 1987; Lake et al. 2017). Compositional representations allow learning new concepts from a few examples by composing learned primitives, which is a desired property for few-shot learning approaches.

Andreas *et al.* (Andreas 2019) propose a method to evaluate the compositionality of the learned representations by measuring how well they can be approximated by the composition of a group of primitives. Following (Andreas 2019), Tokmakov *et al.* (Tokmakov, Wang, and Hebert 2019) design a regularization technique to improve the compositionality of learned representations. ConstellationNet (Xu, Wang, and Tu 2021) uses self-attention mechanisms to model the relation between cell features and utilize the K-means algorithm to conduct cell feature clustering. The learned cluster centers can be viewed as potential object parts. Similarly, CPDE (Zou et al. 2020) learns primitives related to object parts by self-supervision and uses an Enlarging-Reducing loss (ER loss) to enlarge the activation of important primitives and reduce that of others. These inspiring works motivate us to integrate the idea of compositional representations into our few-shot learning method. Concretely, the compositional prototypes constructed by learned component prototypes in our work can be regarded as a kind of compositional representation.

Method

Preliminaries

Problem Formulation In standard few-shot classification, there are two mutually exclusive class sets, base classes set C_{base} and novel classes set C_{novel}, where $C_{base} \cap C_{novel} = \emptyset$. In N-way K-shot setting (Vinyals et al. 2016), we first sample N categories from C_{novel}, and then the support set $\mathcal{S} = \{(x_i, z_i, y_i) \,|y_i \in C_{novel}\}_{i=1}^{N\times K}$ is constructed by sampling K examples for each of the N categories. Here, x_i is the i-th image, y_i denotes the class label of the image, and z_i denotes the attribute score vector of the image, where each dimension of z_i corresponds to an attribute in a predefined attribute set $\mathcal{A} = \{a_j\}_{j=1}^{M}$. Similarly, the query set $\mathcal{Q} = \{(x_i, y_i) \,|y_i \in C_{novel}\}_{i=1}^{N\times Q}$ contains Q examples for each of the N categories. It is worth noting that the attribute score vector is unavailable for examples in the query set. An episode (task) is comprised of a support set and a query set. Meanwhile, we have a base classes dataset $\mathcal{D}_{base} = \{(x_i, z_i, y_i)|y_i \in C_{base}\}$ which contains abundant examples per base class, and our goal is to learn to classify the examples in the query set $\mathcal{Q}$ with the help of the support set $\mathcal{S}$ and the base classes dataset $\mathcal{D}_{base}$. It should be pointed out that examples from the same class have the same attribute score vector because we only use category-level attribute annotations.

Pre-Training Based Meta-Learning Methods Our work follows the line of pre-training based meta-learning methods (Chen et al. 2021b; Ye et al. 2020; Yang, Wang, and Chen 2022). These methods usually train the model in two stages, pre-training and meta-training. And then the learned model is evaluated in the meta-testing stage. Concretely, in the pre-training stage, a feature extractor f_θ and a classifier are trained on the base classes dataset $\mathcal{D}_{base}$ with standard cross-entropy loss. The pre-trained classifier is removed, and the pre-trained feature extractor f_{θ^*} is preserved. In the meta-training stage, collections of N-way K-shot tasks are sampled from the base classes dataset $\mathcal{D}_{base}$, and each task consists of a support set and a query set. The sampled tasks are used to train a learnable distance metric (Chen et al. 2021b) or module (Ye et al. 2020) with or without fine-tuning the pre-trained feature extractor f_{θ^*}. Finally, in the meta-testing stage, where tasks are sampled from novel classes, the learned model can directly utilize the support set $\mathcal{S}$ to classify the examples on the query set $\mathcal{Q}$ without additional training on the support set $\mathcal{S}$.

Component Prototype Learning

Different from most existing methods (Chen et al. 2021b; Ye et al. 2020) which only reuse the feature extractor learned in the pre-training stage, we propose *Compositional Prototypical Networks* to learn a prototype for each attribute a_j in the predefined attribute set $\mathcal{A} = \{a_j\}_{j=1}^{M}$, which we call a component prototype. These component prototypes can be viewed as a kind of meta-knowledge since they can be shared across different learning tasks. Specifically, in the meta-training and meta-testing stage, they can be reused to construct compositional prototypes for classes from the support set.

As shown in Figure 2, in the pre-training stage, we define a set of learnable component prototypes $\{\mathbf{r}_j\}_{j=1}^{M}$ for the predefined attribute set $\mathcal{A} = \{a_j\}_{j=1}^{M}$, where $\mathbf{r}_j \in \mathbb{R}^d$ denotes the component prototype for the attribute a_j. Then we utilize the learnable component prototypes $\{\mathbf{r}_j\}_{j=1}^{M}$ and category-level attribute score vector z_c to calculate the class prototype $\mathbf{p}_c$ for each class $c \in C_{base}$. Concretely, we first perform L2-normalization on the learnable component prototypes $\{\mathbf{r}_j\}_{j=1}^{M}$ to obtain the L2-normalized component prototypes $\{\hat{\mathbf{r}}_j\}_{j=1}^{M}$, where $\hat{\mathbf{r}}_j = \mathbf{r}_j/||\mathbf{r}_j||_2$. Then we calculate the class prototype $\mathbf{p}_c$ as a weighted sum of the L2-normalized component prototypes $\{\hat{\mathbf{r}}_j\}_{j=1}^{M}$:

$$\mathbf{p}_c = \sum_{j=1}^{M} z_{c,j}\hat{\mathbf{r}}_j, \tag{1}$$

where $z_{c,j}$ denotes the score for class c on attribute a_j.

Given an image x_i from the base classes dataset $\mathcal{D}_{base}$, we use cosine similarity to calculate the probability that x_i belongs to class c as follows:

$$P(\hat{y} = c \mid x_i) = \frac{\exp\left(\tau_1 \cdot \langle f_\theta(x_i), \mathbf{p}_c\rangle\right)}{\sum_{c'} \exp\left(\tau_1 \cdot \langle f_\theta(x_i), \mathbf{p}_{c'}\rangle\right)}, \tag{2}$$

where $f_\theta(x_i) \in \mathbb{R}^d$, and τ_1 is a learnable temperature parameter, and $\langle\cdot,\cdot\rangle$ denotes cosine similarity operator. The classification loss for the whole base classes dataset $\mathcal{D}_{base}$ can be defined as follows using cross-entropy loss:

$$\mathcal{L} = -\frac{1}{|\mathcal{D}_{base}|} \sum_{(x_i,z_i,y_i)\in\mathcal{D}_{base}} \log P\left(\hat{y} = y_i|x_i\right). \tag{3}$$

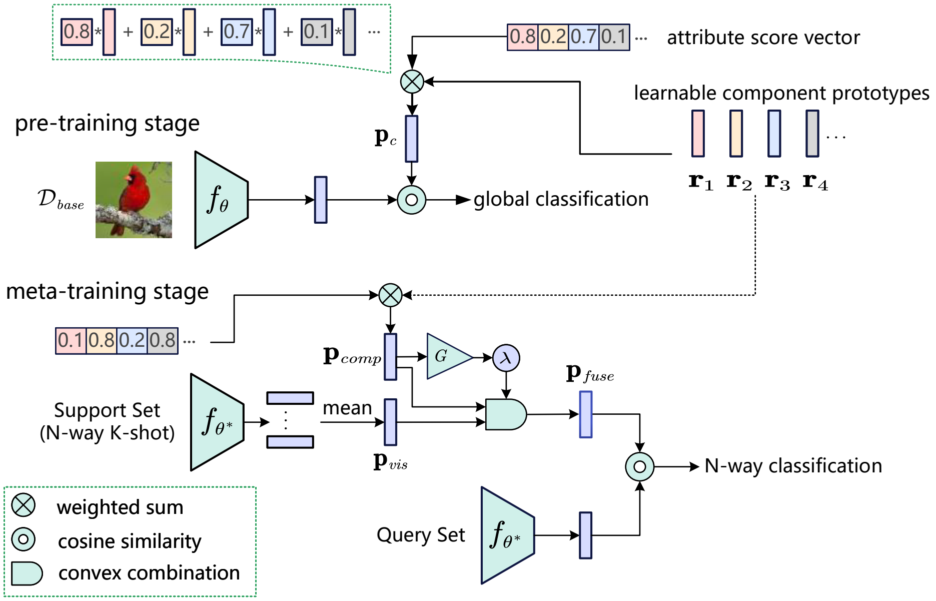


Figure 2: Compositional Prototypical Networks (CPN). In the pre-training stage, CPN trains a feature extractor f_θ and a set of component prototypes $\{r_j\}_{j=1}^M$ for the predefined attribute set $\mathcal{A} = \{a_j\}_{j=1}^M$ by minimizing the global classification loss for examples from the base classes dataset $\mathcal{D}_{base}$. The class prototype p_c is constructed as a weighted sum of the L2-normalized component prototypes. Then in the meta-training stage, the learned component prototypes are reused to construct the compositional prototype p_{comp}. Meanwhile, CPN calculates the mean feature for examples of each class in the support set as the visual prototype p_{vis}. A learnable weight generator G is used to adaptively fuse the compositional and visual prototypes. The fused prototype p_{fuse} is used to classify examples from the query set with cosine similarity as the distance metric.

The model is trained by minimizing Equation 3 and then we can obtain learned component prototypes $\{\mathbf{r}_j^*\}_{j=1}^M$, which can be regarded as transferable meta-knowledge distilled from the base classes dataset $\mathcal{D}_{base}$.

Adaptive Prototype Fusion

In the meta-training (episodic training) stage, a collection of N-way K-shot tasks sampled from the base classes dataset $\mathcal{D}_{base}$ are given to mimic the scenario in the meta-testing stage. In this stage, we focus on learning an adaptive weight generator (Ji et al. 2022; Xing et al. 2019; Ma et al. 2022), which is used to fuse two types of prototypes, namely, a compositional prototype $\mathbf{p}_{comp}$ and a visual prototype $\mathbf{p}_{vis}$. To simplify the notation, we choose one of the N classes from the support set to illustrate the whole fusion process. That is to say, $\mathbf{p}_{comp}$ and $\mathbf{p}_{vis}$ correspond to any class s from the support set.

Similar to Equation 1, we reuse the learned component prototypes to construct the compositional prototype $\mathbf{p}_{comp}$ for class s:

$$\mathbf{p}_{comp} = \sum_{j=1}^{M} z_{s,j} \hat{\mathbf{r}}_j^*, \tag{4}$$

where $\hat{\mathbf{r}}_j^* = \mathbf{r}_j^* / ||\mathbf{r}_j^*||_2$, and $z_{s,j}$ denotes the score for class s on attribute a_j. Following Prototypical Networks (Snell, Swersky, and Zemel 2017), we calculate the visual prototype $\mathbf{p}_{vis}$ for class s as the mean vector of the extracted features for examples labeled with class s in the support set:

$$\mathbf{p}_{vis} = \frac{1}{K} \sum_{i \in \{i | y_i = s\}} f_{\theta^*}(x_i), \tag{5}$$

where f_{θ^*} denotes the pre-trained feature extractor, and K is the number of examples belonging to class s since the support set is sampled using N-way K-shot setting.

Before we fuse the above two prototypes, we need to L2-normalize them. Following (Ji et al. 2022; Xing et al. 2019), we employ a convex combination to fuse the L2-normalized compositional prototype $\hat{\mathbf{p}}_{comp}$ and the L2-normalized visual prototype $\hat{\mathbf{p}}_{vis}$, where $\hat{\mathbf{p}}_{comp} = \mathbf{p}_{comp} / ||\mathbf{p}_{comp}||_2$ and $\hat{\mathbf{p}}_{vis} = \mathbf{p}_{vis} / ||\mathbf{p}_{vis}||_2$. We use a learnable weight generator G followed by a sigmoid function to adaptively generate a weight coefficient λ and use the coefficient λ to calculate the fused prototype $\mathbf{p}_{fuse}$ as follows:

$$\lambda = \frac{1}{1 + \exp\left(-G\left(\hat{\mathbf{p}}_{comp}\right)\right)}, \tag{6}$$

$$\mathbf{p}_{fuse} = \lambda \hat{\mathbf{p}}_{comp} + (1 - \lambda)\, \hat{\mathbf{p}}_{vis}. \tag{7}$$

The sigmoid function used in Equation 6 is to make sure the weight coefficient λ is between 0 and 1 so that the right side in Equation 7 is a convex combination. Given an image q_i from the query set, we still use cosine similarity to predict the probability that q_i belongs to class s as follows:

$$P(\hat{y} = s \mid q_i) = \frac{\exp\left(\tau_2 \cdot \langle f_{\theta^*}(q_i), \mathbf{p}_{fuse}^s \rangle\right)}{\sum_{s'} \exp\left(\tau_2 \cdot \langle f_{\theta^*}(q_i), \mathbf{p}_{fuse}^{s'} \rangle\right)}, \tag{8}$$

where $\mathbf{p}^{s'}_{fuse}$ denotes the fused prototype for class s', and τ_2 is another learnable temperature parameter.

In the meta-training stage, the learnable weight generator G and the learnable temperature parameter τ_2 are optimized by minimizing the classification loss of examples in the query set. It is worth noting that we **further** optimize the learned component prototypes in the meta-training stage. This implementation is based on experimental results. Finally, in the meta-testing stage, the learned model calculates the fused prototype for each novel class from the support set $\mathcal{S}$ following Equation 4-7. The calculated prototypes can be directly used to classify examples in the query set $\mathcal{Q}$ according to Equation 8.

Experiments

Experimental Setup

Datasets. We conduct the experiments on two datasets with human-annotated attribute annotations: Caltech-UCSD-Birds 200-2011 (CUB) (Wah et al. 2011), and SUN Attribute Database (SUN) (Patterson et al. 2014). CUB is a fine-grained bird species dataset containing 11,788 bird images from 200 species and 312 predefined attributes. There is only one attribute vector for each class in CUB, which we call a category-level attribute score vector. SUN is a scene recognition dataset containing 14,340 images from 717 categories and 102 predefined attributes. It should be noted that each image in SUN has an attribute score vector, which we call an image-level attribute score vector. As we have mentioned earlier, we only use category-level attribute score vectors. To this end, we calculate the mean vector of image-level attribute score vectors from the same class as this class's attribute score vector.

Experimental Settings. Our experiments are conducted in 5-way 1-shot and 5-way 5-shot settings. As (Chen et al. 2019a), we divide CUB into 100 training classes, 50 validation classes, and 50 testing classes. As (Huang et al. 2021; Ji et al. 2022), we divide SUN into 580 training classes, 65 validation classes, and 72 testing classes. We sample 15 query examples per class in each task for the meta-training and meta-testing stages. We report the *average accuracy* (%) and the corresponding 95% *confidence intervals* over 5000 test episodes to make a fair comparison. Our work follows the inductive setting, where each example in the query set is classified independently.

Implementation Details. Our experiments are conducted using the convolution neural network ResNet12 (Chen et al. 2021b), a popular feature extractor in recent few-shot learning methods. We also give our results using Conv4 (Vinyals et al. 2016) on SUN for fair comparisons since we find that almost all previous works (Huang et al. 2021; Ji et al. 2022; Chen et al. 2022; Xing et al. 2019) choose Conv4 as their feature extractor on SUN. Moreover, we use a simple fully-connected layer as the learnable weight generator G, and the temperature parameter τ_1 and τ_2 are initialized as 10. We use the SGD optimizer with a momentum of 0.9 and weight decay of 5×10^{-4}. Following (Yang, Wang, and Chen 2022), we adopt the random crop, random horizontal flip and erasing, and color jittering to perform data augmentation. Dropblock (Ghiasi, Lin, and Le 2018) regularization is used to reduce overfitting. In the pre-training stage, we train the feature extractor for 30 epochs. In the meta-training stage, we train our model for 10 epochs. The best model is chosen according to the accuracy on the validation set.

Comparison to the State-of-the-Art

Table 1 and 2 show the results of our method and previous state-of-the-art methods on CUB and SUN, respectively. It can be observed that the proposed CPN achieves the best performance among all approaches in both 5-way 1-shot and 5-way 5-shot settings. We notice that the performance gains are remarkable in the 5-way 1-shot setting. It indicates that the compositional prototype constructed by the learned component prototypes plays a significant role, especially when the visual prototype is inaccurate due to the limited examples in the 5-way 1-shot setting.

Moreover, we notice that better performance can be achieved by using a stronger feature extractor. Concretely, when we replace Conv4 with ResNet12 on SUN, we can obtain 7.62% and 7.65% performance gains in 5-way 1-shot and 5-way 5-shot settings, respectively. It shows that our method can benefit a lot from a more powerful feature extractor.

We attribute the effectiveness of our method to two aspects. The first is that the learned component prototypes have good class transferability. By reusing these transferable component prototypes to construct the compositional prototype for a novel class, we can roughly get the class center for this class. The second is that the adaptive fusion of the compositional and visual prototypes can further improve the classification accuracy by combining visual and semantic information. To illustrate these points, we perform carefully designed ablation experiments in Ablation Study.

Ablation Study

To verify that CPN does learn some meaningful component prototypes, we design several experiments using different prototypes: (1) RICP, the compositional prototype constructed by **R**andomly **I**nitialized **C**omponent **P**rototypes. (2) VP, the **V**isual **P**rototype. (3) LCP, the compositional prototype constructed by **L**earned **C**omponent **P**rototypes. (4) Adaptive fusion of RICP and VP. (5) Adaptive fusion of LCP and VP. (6) Naive fusion using the concatenation of VP and LCP+. We use a fully connected layer to reduce the dimension of the concatenation so that it has the same dimension as the visual feature. (7) Ours, adaptive fusion of VP and LCP+. LCP+ in (6) and (7) means the learned component prototypes will be further optimized in the meta training stage. The results are shown in Table 3.

(i) According to the first and third row of Table 3, we know that the learned component prototypes do have good class transferability since LCP achieves quite good performance. In contrast, RICP seems meaningless since its classification accuracy is almost equivalent to a random classifier (20%). (ii) From the second and fourth row of Table

Method	Backbone	Test Accuracy	
		5-way 1-shot	5-way 5-shot
MatchingNet (Vinyals et al. 2016) ‡	ResNet12	60.96 ± 0.35	77.31 ± 0.25
ProtoNet (Snell, Swersky, and Zemel 2017)	ResNet12	68.8	76.4
FEAT (Ye et al. 2020)	ResNet12	68.87 ± 0.22	82.90 ± 0.15
MAML (Finn, Abbeel, and Levine 2017)	ResNet18	69.96 ± 1.01	82.70 ± 0.65
AFHN (Li et al. 2020)	ResNet18	70.53 ± 1.01	83.95 ± 0.63
CPDE (Zou et al. 2020)	ResNet18	80.11 ± 0.34	89.28 ± 0.33
BlockMix (Tang et al. 2020)	ResNet12	75.31 ± 0.79	88.53 ± 0.49
DeepEMD (Zhang et al. 2020)	ResNet12	75.65 ± 0.83	88.69 ± 0.50
AGPF (Tang et al. 2022)	ResNet12	78.73 ± 0.84	89.77 ± 0.47
MetaNODE (Zhang et al. 2022)	ResNet12	80.82 ± 0.75	91.77 ± 0.49
Comp. (Tokmakov, Wang, and Hebert 2019) *	ResNet10	53.6	74.6
Dual TriNet (Chen et al. 2019b) * $^\circ$	ResNet18	69.61 ± 0.46	84.10 ± 0.35
AM3 (Xing et al. 2019) *	ResNet12	73.6	79.9
Multiple-Semantics (Schwartz et al. 2022) * $^\circ$ $^\bullet$	DenseNet121	76.1	82.9
AGAM (Huang et al. 2021) *	ResNet12	79.58 ± 0.25	87.17 ± 0.23
ASL (Chen et al. 2022) *	ResNet12	82.12 ± 0.14	89.65 ± 0.11
SEGA (Yang, Wang, and Chen 2022) *	ResNet12	84.57±0.22	90.85 ± 0.16
MAP-Net (Ji et al. 2022) *	ResNet12	82.45 ± 0.23	88.30 ± 0.17
CPN (Ours) *	ResNet12	**87.29 ± 0.20**	**92.54 ± 0.14**

Table 1: Comparison with state-of-the-art methods on CUB. We report the average accuracy (%) with 95% confidence intervals over 5000 test episodes. * denotes that it uses attribute annotations. $^\circ$ denotes that it uses word embeddings. $^\bullet$ denotes that it uses natural language descriptions. ‡ denotes that the results are reported in (Huang et al. 2021). Best results are displayed in boldface.

Method	Backbone	Test Accuracy	
		5-way 1-shot	5-way 5-shot
RelationNet (Sung et al. 2018)‡	Conv4	49.58 ± 0.35	76.21 ± 0.19
MatchingNet (Vinyals et al. 2016)‡	Conv4	55.72 ± 0.40	76.59 ± 0.21
ProtoNet (Snell, Swersky, and Zemel 2017)‡	Conv4	57.76 ± 0.29	79.27 ± 0.19
Comp. (Tokmakov, Wang, and Hebert 2019)*	ResNet10	45.9	67.1
AM3 (Xing et al. 2019)* ‡	Conv4	62.79 ± 0.32	79.69 ± 0.23
AGAM (Huang et al. 2021)*	Conv4	65.15 ± 0.31	80.08 ± 0.21
ASL (Chen et al. 2022)*	Conv4	66.17 ± 0.17	80.91 ± 0.15
MAP-Net (Ji et al. 2022) *	Conv4	67.73 ± 0.30	80.30 ± 0.21
CPN-Conv4 (Ours) *	Conv4	**80.45 ± 0.22**	**81.56 ± 0.21**
CPN (Ours) *	ResNet12	**88.07 ± 0.17**	**89.21 ± 0.15**

Table 2: Comparison with state-of-the-art methods on SUN. We report the average accuracy (%) with 95% confidence intervals over 5000 test episodes. * denotes that it uses attribute annotations. ‡ denotes that the results are reported in (Huang et al. 2021). Best results are displayed in boldface.

3, we learn that RICP still does not work even if we adaptively fuse RICP and VP. By contrast, the classification accuracy in the fifth row is generally higher than both the second and third row in Table 3. It suggests the adaptive fusion of LCP and VP can achieve a better performance in most cases. (iii) From the fifth and last row of Table 3, we know that better performance can be achieved if we further optimize the learned component prototypes in the meta-training stage. (iv) Moreover, from the last and penultimate row of Table 3, we learn that the adaptive fusion of VP and LCP+ is better than the naive fusion of VP and LCP+.

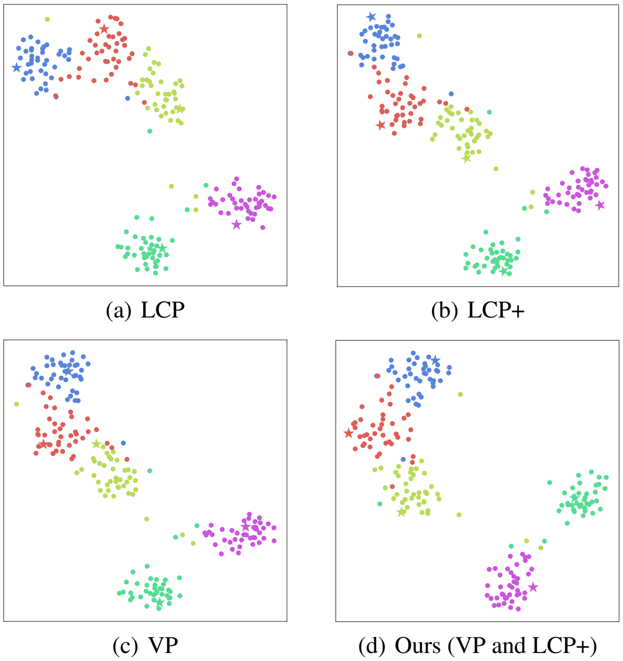
In Equation 6, we use the L2-normalized compositional prototype $\hat{\mathbf{p}}_{comp}$ as the input of the weight generator G following (Xing et al. 2019), where the semantic label embedding is used to generate a weight coefficient. However, there are other available choices for the input of the weight generator. The results of related ablation experiments are shown in Table 4. We can observe that using $\hat{\mathbf{p}}_{comp}$ as the input of G achieves the best performance in the 1-shot setting. By contrast, using the concatenation of $\hat{\mathbf{p}}_{vis}$ and $\hat{\mathbf{p}}_{comp}$ achieves the best performance in the 5-shot setting. It may be because $\hat{\mathbf{p}}_{vis}$ provides meaningless information in the 1-shot setting due to the lim ited examples, and using $\hat{\mathbf{p}}_{comp}$ only is more suitable. By contrast, in the 5-shot setting, $\hat{\mathbf{p}}_{vis}$ summarizes more representative class information from several examples and can provide useful information.

RICP	VP	LCP	CUB		SUN	
			5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
✓			19.43 ± 0.21	17.44 ± 0.19	20.96 ± 0.20	18.06 ± 0.18
	✓		79.62 ± 0.27	92.11 ± 0.14	71.21 ± 0.29	86.61 ± 0.18
		✓	82.04 ± 0.24	82.10 ± 0.24	85.63 ± 0.19	85.53 ± 0.19
✓	✓		79.66 ± 0.27	92.13 ± 0.14	71.21 ± 0.29	86.60 ± 0.18
	✓	✓	84.54 ± 0.22	89.34 ± 0.17	86.54 ± 0.18	88.28 ± 0.16
[VP, LCP+]			85.27 ± 0.21	90.49 ± 0.16	86.61 ± 0.18	88.31 ± 0.16
Ours (VP and LCP+)			**87.29** ± **0.20**	**92.54** ± **0.14**	**88.07** ± **0.17**	**89.21** ± **0.15**

Table 3: Ablation experiments on using different prototypes.VP, the *V*isual *P*rototype. RICP, the compositional prototype constructed by *R*andomly *I*nitialized *C*omponent *P*rototypes. LCP, the compositional prototype constructed by *L*earned *C*omponent *P*rototypes. LCP+ means the learned component prototypes will be further optimized in the meta training stage. [VP, LCP+], naive fusion using the concatenation of VP and LCP+. Ours, adaptive fusion of VP and LCP+.

Input	CUB		SUN	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
$[\hat{\mathbf{p}}_{vis}, \hat{\mathbf{p}}_{comp}]$	87.01 ± 0.20	**92.77** ± **0.13**	87.26 ± 0.17	**89.47** ± **0.15**
$\hat{\mathbf{p}}_{vis}$	86.36 ± 0.21	92.56 ± 0.14	87.48 ± 0.17	89.30 ± 0.15
$\hat{\mathbf{p}}_{comp}$	**87.29** ± **0.20**	92.54 ± 0.14	**88.07** ± **0.11**	89.21 ± 0.15

Table 4: Ablation experiments on the input of the weight generator. $[\hat{p}_{vis}, \hat{p}_{comp}]$ denotes that the input is the concatenation of the the L2-normalized visual prototype $\hat{p}_{vis}$ and the L2-normalized compositional prototype $\hat{p}_{comp}$.


(a) LCP

(b) LCP+

(c) VP

(d) Ours (VP and LCP+)

Figure 3: The t-SNE visualization using different prototypes (See Ablation Study for the meaning of VP, LCP and LCP+). '⋆' denote prototypes for different classes, and '•' denotes features for the query set. We sample 40 examples from each class to form the query set for a better view.

Visualization Analysis

As shown in Figure 3, we use t-SNE (Van der Maaten and Hinton 2008) to visualize the feature distributions using different prototypes in the 5-way 1-shot setting on CUB. From Figure 3(a), we can observe that each class's compositional prototype constructed by the learned component prototypes is well distributed among the examples of that class. This shows that the learned component prototypes have good transferability.

From Figure 3(c), we notice that the visual prototype for the yellow-green class is also very close to the red class, which may result in serious misclassification. This suggests that the visual prototype in the 1-shot setting may not be representative enough due to the limited examples. Figure 3(d) shows the feature distribution using the adaptively fused prototype of VP from Figure 3(c) and LCP+ from Figure 3(b). Compared to Figure 3(c), the adaptively fused prototype for the yellow-green class is pulled away from the red class, which benefits the final classification.

Conclusion

In this work, we propose a novel Compositional Prototypical Network (CPN) to learn component prototypes for predefined attributes in the pre-training stage. The learned component prototypes can be reused to construct a compositional prototype for each class in the support set. And in the meta-training stage, we further optimize the learned component prototypes and learn an adaptive weight generator to fuse the compositional and visual prototypes. We empirically show that the learned component prototypes have good class transferability. Moreover, we show that the adaptive fusion of the compositional and visual prototypes can further improve classification performance. We hope our work can bring more attention and thought to feature reusability and compositional representations in the few-shot learning field.

Acknowledgments

This work is supported by NSFC projects under Grant 61976201, and NSFC Key Projects of International (Regional) Cooperation and Exchanges under Grant 61860206004. We would like to thank Yuzhong Zhao for inspiring and helpful discussions.

References

Afham, M.; Khan, S.; Khan, M. H.; Naseer, M.; and Khan, F. S. 2021. Rich Semantics Improve Few-shot Learning. *32nd British Machine Vision Conference*.

Andreas, J. 2019. Measuring Compositionality in Representation Learning. In *International Conference on Learning Representations*.

Andrychowicz, M.; Denil, M.; Gomez, S.; Hoffman, M. W.; Pfau, D.; Schaul, T.; Shillingford, B.; and De Freitas, N. 2016. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, 3981–3989.

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

Cao, K.; Brbic, M.; and Leskovec, J. 2021. Concept Learners for Few-Shot Learning. In *International Conference on Learning Representations*.

Chen, H.; Li, H.; Li, Y.; and Chen, C. 2022. Shaping Visual Representations with Attributes for Few-Shot Recognition. *IEEE Signal Processing Letters*.

Chen, S.; Xie, G.; Liu, Y.; Peng, Q.; Sun, B.; Li, H.; You, X.; and Shao, L. 2021a. Hsva: Hierarchical semantic-visual adaptation for zero-shot learning. *Advances in Neural Information Processing Systems*, 34.

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C. F.; and Huang, J.-B. 2019a. A Closer Look at Few-shot Classification. In *International Conference on Learning Representations*.

Chen, Y.; Liu, Z.; Xu, H.; Darrell, T.; and Wang, X. 2021b. Meta-Baseline: Exploring Simple Meta-Learning for Few-Shot Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9062–9071.

Chen, Z.; Fu, Y.; Zhang, Y.; Jiang, Y.-G.; Xue, X.; and Sigal, L. 2019b. Multi-level semantic feature augmentation for one-shot learning. *IEEE Transactions on Image Processing*, 28(9): 4594–4605.

Dhillon, G. S.; Chaudhari, P.; Ravichandran, A.; and Soatto, S. 2020. A Baseline for Few-Shot Image Classification. In *International Conference on Learning Representations*.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4): 594–611.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 1126–1135. PMLR. ISBN 2640-3498.

Ghiasi, G.; Lin, T.-Y.; and Le, Q. V. 2018. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, 31.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hoffman, D. D.; and Richards, W. A. 1984. Parts of recognition. *Cognition*, 18(1-3): 65–96.

Hou, R.; Chang, H.; MA, B.; Shan, S.; and Chen, X. 2019. Cross Attention Network for Few-shot Classification. *Advances in Neural Information Processing Systems*, 32: 4003–4014.

Huang, S.; Zhang, M.; Kang, Y.; and Wang, D. 2021. Attributes-Guided and Pure-Visual Attention Alignment for Few-Shot Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7840–7847. ISBN 2374-3468.

Ji, Z.; Hou, Z.; Liu, X.; Pang, Y.; and Han, J. 2022. Information Symmetry Matters: A Modal-Alternating Propagation Network for Few-Shot Learning. *IEEE Transactions on Image Processing*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105.

Lake, B. M.; Ullman, T. D.; Tenenbaum, J. B.; and Gershman, S. J. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

Li, K.; Zhang, Y.; Li, K.; and Fu, Y. 2020. Adversarial feature hallucination networks for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13470–13479.

Li, Y.; Zhang, J.; Zhang, J.; and Huang, K. 2018. Discriminative learning of latent features for zero-shot recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7463–7471.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Ma, R.; Fang, P.; Drummond, T.; and Harandi, M. 2022. Adaptive poincaré point to set distance for few-shot classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 1926–1934.

Miller, E. G.; Matsakis, N. E.; and Viola, P. A. 2000. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, 464–471. IEEE.

Patterson, G.; Xu, C.; Su, H.; and Hays, J. 2014. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1): 59–81.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Raghu, A.; Raghu, M.; Bengio, S.; and Vinyals, O. 2020. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. In *International Conference on Learning Representations*.

Ravi, S.; and Larochelle, H. 2017. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017*.

Rubner, Y.; Tomasi, C.; and Guibas, L. J. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2): 99–121.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2019. Meta-Learning with Latent Embedding Optimization. In *International Conference on Learning Representations*.

Schmidhuber, J. 1987. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. Ph.D. thesis, Technische Universität München.

Schonfeld, E.; Ebrahimi, S.; Sinha, S.; Darrell, T.; and Akata, Z. 2019. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8247–8255.

Schwartz, E.; Karlinsky, L.; Feris, R.; Giryes, R.; and Bronstein, A. 2022. Baby steps towards few-shot learning with multiple semantics. *Pattern Recognition Letters*, 160: 142–147.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical Networks for Few-shot Learning. *Advances in Neural Information Processing Systems*, 30: 4077–4087.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1199–1208.

Tang, H.; Li, Z.; Peng, Z.; and Tang, J. 2020. Blockmix: meta regularization and self-calibrated inference for metric-based meta-learning. In *Proceedings of the 28th ACM international conference on multimedia*, 610–618.

Tang, H.; Yuan, C.; Li, Z.; and Tang, J. 2022. Learning Attention-Guided Pyramidal Features for Few-shot Fine-grained Recognition. *Pattern Recognition*, 108792.

Tian, Y.; Wang, Y.; Krishnan, D.; Tenenbaum, J. B.; and Isola, P. 2020. Rethinking few-shot image classification: a good embedding is all you need? In *European Conference on Computer Vision*, 266–282. Springer.

Tokmakov, P.; Wang, Y.-X.; and Hebert, M. 2019. Learning compositional representations for few-shot recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6372–6381.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Verma, V. K.; Arora, G.; Mishra, A.; and Rai, P. 2018. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4281–4289.

Vinyals, O.; Blundell, C.; Lillicrap, T.; and Wierstra, D. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29: 3630–3638.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset. *Technical Report CNS-TR-2011-001, California Institute of Technology*.

Wang, R.; Pontil, M.; and Ciliberto, C. 2021. The Role of Global Labels in Few-Shot Classification and How to Infer Them. *Advances in Neural Information Processing Systems*, 34.

Xian, Y.; Sharma, S.; Schiele, B.; and Akata, Z. 2019. f-vaegan-d2: A feature generating framework for any-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10275–10284.

Xing, C.; Rostamzadeh, N.; Oreshkin, B.; and O Pinheiro, P. O. 2019. Adaptive cross-modal few-shot learning. *Advances in Neural Information Processing Systems*, 32: 4847–4857.

Xu, W.; Wang, H.; and Tu, Z. 2021. Attentional Constellation Nets for Few-Shot Learning. In *International Conference on Learning Representations*.

Yang, F.; Wang, R.; and Chen, X. 2022. SEGA: Semantic Guided Attention on Visual Prototype for Few-Shot Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1056–1066.

Ye, H.-J.; Hu, H.; Zhan, D.-C.; and Sha, F. 2020. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8808–8817.

Zhang, B.; Li, X.; Feng, S.; Ye, Y.; and Ye, R. 2022. MetaNODE: Prototype optimization as a neural ODE for few-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 9014–9021.

Zhang, B.; Li, X.; Ye, Y.; Huang, Z.; and Zhang, L. 2021. Prototype Completion with Primitive Knowledge for Few-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3754–3762.

Zhang, C.; Cai, Y.; Lin, G.; and Shen, C. 2020. 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*, 12203–12213.

Zou, Y.; Zhang, S.; Chen, K.; Tian, Y.; Wang, Y.; and Moura, J. M. 2020. Compositional few-shot recognition with primitive discovery and enhancing. In *Proceedings of the 28th ACM International Conference on Multimedia*, 156–164.