Prototype Completion for Few-Shot Learning

Baoquan Zhang, Xutao Li, Yuming Ye, and Shanshan Feng

Abstract—Few-shot learning (FSL) aims to recognize novel classes with few examples. Pre-training based methods effectively tackle the problem by pre-training a feature extractor and then fine-tuning it through the nearest centroid based meta-learning. However, results show that the fine-tuning step makes marginal improvements. In this paper, 1) we figure out the reason, i.e., in the pre-trained feature space, the base classes already form compact clusters while novel classes spread as groups with large variances, which implies that fine-tuning feature extractor is less meaningful; 2) instead of fine-tuning feature extractor, we focus on estimating more representative prototypes. Consequently, we propose a novel prototype completion based meta-learning framework. This framework first introduces primitive knowledge (i.e., class-level part or attribute annotations) and extracts representative features for seen attributes as priors. Second, a part/attribute transfer network is designed to learn to infer the representative features for unseen attributes as supplementary priors. Finally, a prototype completion network is devised to learn to complete prototypes with these priors. Moreover, to avoid the prototype completion error, we further develop a Gaussian based prototype fusion strategy that fuses the mean-based and completed prototypes by exploiting the unlabeled samples. At last, we also develop an economic prototype completion version for FSL, which does not need to collect primitive knowledge, for a fair comparison with existing FSL methods without external knowledge. Extensive experiments show that our method: i) obtains more accurate prototypes; ii) achieves superior performance on both inductive and transductive FSL settings.

Index Terms—Few-Shot learning, image classification, meta-learning.

I. INTRODUCTION

HUMANS can adapt to a novel task from only a few observations, because our brains have the excellent capability of learning to learn. In contrast, modern artificial intelligence (AI) systems generally require a large amount of annotated samples to make the adaptations, such as image classification [1]. However, preparing sufficient annotated samples is often laborious, expensive, or even unrealistic in some applications such as cold-start recommendation [2] and drug discovery [3]. To equip the AI systems with such human-like ability, few-shot learning (FSL) becomes an important and widely studied problem. Different from conventional machine learning, FSL aims to learn a classifier from a set of base classes with abundant labeled samples, then adapt to a set of novel classes with few examples [4].

Previous studies on FSL roughly fall into four categories, namely the metric-based methods [5], [6], [7], optimization-based methods [8], [9], graph-based methods [10], [11], and semantics-based methods [12], [13]. Though their methodologies are quite different, almost all methods address the FSL problem by a two-phase meta-learning framework, i.e., i) a meta-training phase that learns meta-knowledge from a large number of base class tasks, and ii) a meta-test phase that quickly constructs a model for novel class prediction with the meta-knowledge. Recently, Chen et al. [14] find that introducing an extra pre-training phase can significantly boost the performance. In this method, a feature extractor first is pre-trained by learning a classifier on the entire base classes. Then, the metric-based meta-learning is adopted to fine-tune it. In the meta-test phase, the mean-based prototypes are constructed to classify novel classes via the nearest neighbor classifier with cosine distance.

Though the pre-training based meta-learning method has achieved promising improvements on FSL, Chen et al. find that the fine-tuning step indeed makes very marginal contributions [14] during meta-learning. In other words, the power of the pre-trained model is not effectively explored by the meta-learning methods. However, the reason is not revealed in [14]. To figure out the reason, we visualize the distribution of base and novel class samples of miniImagenet in the pre-trained feature space. \( \sigma^2 \) denotes the averaged variance.
the feature extractor to gather the base class samples into more compact clusters is less meaningful, because this enlarges the probability to overfit the base tasks; and ii) the given few labeled samples may be far away from its ground-truth class centers in the case of large variances for novel classes, which poses a great challenge for estimating representative prototypes. Hence, in this paper, instead of fine-tuning feature extractor, we focus on how to estimate representative prototypes from few labeled samples, especially when these samples are far away from their ground-truth class centers.

Recently, Xue et al. [15] also attempt to address a similar problem by learning a mapping function from noisy samples to their ground-truth class centers. However, learning to recover representative prototypes from noisy samples without any priors is very difficult. Moreover, the method does not leverage the pre-training strategy. Thus, its performance improvement is limited. In this paper, inspired by the visual attribute learning [16], [17], we find that the samples deviated from its ground-truth centers are often incomplete, i.e., missing some representative attribute features. As shown in Fig. 1(b), the meerkat sample nearby the class center contains all the representative features, e.g., the head, body, legs, and tail, while the ones far away may miss some representative features such as legs and tail. This means that the prototypes estimated by the samples deviated from its class centers may be incomplete, which limits the classification performance of FSL.

Based on this fact, we propose a novel prototype completion framework for FSL. Our framework works in a pre-training manner and introduces some primitive knowledge (i.e., class-level attribute or part annotations), e.g., whether a class object should have ears, legs or eyes, as priors to achieve the prototype completion. Specifically, we first extract the visual features for each seen part/attribute, by aggregating the pre-trained feature representations of all the base class samples that have the corresponding attributes in our primitive knowledge. Second, a Part/Attribute Transfer Network (PATNet) is then designed to infer the visual features for each unseen part/attribute. Third, we mimic the setting of few-shot classification task and construct a set of prototype completion tasks. A Prototype Completion Network (ProtoComNet) is then developed to learn to complete representative prototypes with the primitive knowledge and the obtained visual attribute features. To avoid the prototype completion error caused by primitive knowledge noises or base-novel class differences, we further design a Gaussian-based prototype fusion strategy, which effectively combines the mean-based and completed prototypes by exploiting the unlabeled data. Finally, the few-shot classification is achieved via a nearest neighbor classifier. Furthermore, for making a fair comparison with existing FSL methods without introducing external knowledge, we also develop an economic prototype completion version for FSL, which extracts the visual features of all latent parts/attributes in an unsupervised clustering manner and does not need to collect primitive knowledge. Our main contributions of this paper can be summarized as follows:

- We reveal the reason why the feature extractor fine-tuning step contributes very marginally to the pre-training based meta-learning methods, and point out that representative prototype estimation is a more important issue for FSL.

- We propose two novel prototype completion based FSL framework, i.e., with primitive knowledge and without primitive knowledge. In this framework with primitive knowledge, a part/attribute transfer network, a prototype completion network and a Gaussian-based prototype fusion strategy are designed, which offer our framework the excellent ability to construct more representative prototypes, by exploiting the primitive knowledge of both seen and unseen parts/attributes. However, in this framework without primitive knowledge, we propose to discovery latent parts/attributes in an unsupervised clustering manner instead of primitive knowledge.

- In the Gaussian-based prototype fusion strategy, we propose and extend three methods to estimate prototype fusion parameters, i.e., a two-step estimation method, an EM (Expectation Maximization)-based estimation method, and an improved EM-based estimation method, which fully exploit the unlabeled data for more accurate prototypes estimation.

- We conduct comprehensive experiments on five real-world data sets. The experimental results demonstrate that our method achieves superior performance in both inductive and transductive FSL settings over state-of-the-art techniques.

This paper is an extension to our conference version in [18]. Compared to the conference paper, this version additionally presents i) a more powerful prototype completion framework for FSL, which introduces a novel part/attribute transfer network for incorporating unseen parts/attributes and develops two new methods (the EM-based and the improve EM-based methods) to estimate fusion parameters for Gaussian-based prototype fusion strategy, and improves the performance significantly; ii) a unified perspective to understand the mean-based prototype fusion strategy and a theoretical analysis on the Gaussian-based prototype fusion strategy; iii) an economic prototype completion version without primitive knowledge for FSL, which provides a more fair comparison with existing FSL methods without external knowledge; iv) more statistical analysis, ablation results, and visualization on miniImagenet, tieredImageNet, and CUB-200-2011, and comparisons with more state-of-the-art methods in both transductive and inductive FSL settings; v) more performance comparison on CIFAR-FS and FC100.

The rest of this work is organized as follows: In Section II, we briefly review related works on few-shot learning, transfer learning via semantic knowledge, and prototype learning. Section III describes our prototype completion framework with primitive knowledge in details, including the overall framework and its three key components, i.e., the part/attribute transfer network, prototype completion network and prototype fusion strategy. Section IV further extends a more economic prototype completion framework, which does not need to collect primitive knowledge. Section V presents and analyzes the experimental results on miniImagenet, tieredImageNet, and CUB-200-2011 data sets. Finally, the conclusion is summarized in Section VI.
II. RELATED WORK

The key idea of the proposed prototype completion-based meta-learning framework is utilizing primitive knowledge to learn to complete prototypes for FSL. Here, the primitive knowledge refers to class-level part or attribute annotations, which can be regarded as external knowledge. Thus, in this section, many relevant studies, including few-shot learning, transfer learning via semantic knowledge, and prototype learning techniques, are reviewed individually.

A. Few-Shot Learning

Most previous FSL methods focus on only leveraging few labeled samples to learn a classifier for novel classes, which can be divided into two groups in terms of their settings, namely the inductive FSL and transductive FSL techniques.

1) Inductive FSL: Most existing studies primarily address the FSL problem using the idea of inductive learning, which assumes the information of test samples is not available when performing few-shot classification tasks. Specifically, these approaches can be grouped into three categories. 1) Metric-based approaches. The type of methods aims to learn a good metric space, where novel class samples can be nicely categorized via a nearest neighbor classifier with euclidean [19], cosine [20], mahalanobis [21], earth mover’s [22], hyperbolic [23], or learnable distance [24], [25], [26]. For example, Chen et al. [27] proposed a variational method to learn a proper scaling parameter for the euclidean or cosine based metric, aiming to better fit the metric space to a given data distribution. 2) Optimization-based approaches. The methods follow the idea of modeling an optimization process over few labeled samples under the meta-learning framework, aiming to adapt to novel tasks by a few optimization steps, such as [28], [29], [30], [31], [32], [33], [34]. For example, in [34], Zhang et al. propose a continuous-time meta-optimizer for fast adaption of FSL classifier.

Recently, some studies turn to pre-training techniques for the FSL problem and achieve promising performance [35]. Chen et al. [20] first proposed and investigated the pre-training techniques in FSL, by considering linear-based and cosine distance-based classifiers, respectively. In [14], a novel metric-based meta-learning method was developed by incorporating a pre-training phase. These methods, albeit delivering promising performance, do not fully explore the power of pre-training, as results show that the major improvements are made by the pre-training while the meta-learning phase contributes very marginally. According to our analysis, this is because novel classes group loosely in the pre-trained feature space. In such case, estimating more accurate and representative prototypes is more important than fine-tuning the projection spaces. Hence, in this paper, we propose a prototype completion framework to address the issue. Recently, there are also other latest pre-training FSL methods such as [36], [37], [38], [39], which focus on developing either a better pre-training strategy or a more powerful parametric classifier. Their strategies are different from our prototype completion framework.

2) Transductive FSL: Different from inductive FSL, transductive FSL assumes that all information from test samples can be used for recognizing novel classes. Such approaches have been proved to be more effective than inductive FSL approaches in data-scarce scenario [40], [41], [42]. These approaches can be divided into two groups. 1) Graph-based approaches. The type of methods learn how to construct a good graph structure and an effective propagation mechanism from base classes as meta-knowledge, and then apply the meta-knowledge on novel classes [10], [11], [43], [44], [45], [46]. For instance, Yang et al. [46] proposed a distribution propagation graph network for transductive FSL, aiming to propagate labels from labeled samples to unlabeled samples with the graph. 2) Pre-training based approaches. The methods also focus on the pre-training feature extractor and attempt to learn a classifier (e.g., SVM) [47], [48], [49], [50], [51], [52] or enhance prototypes by leveraging unlabeled samples [53]. For example, Liu et al. [53] developed a label propagation and feature shifting strategy to diminish the intra-class and cross-class prototypes bias in the pre-trained feature space. Different from these studies, we leverage the unlabeled samples to estimate prototype distribution and then leverage it to fuse prototypes. As far as we know, this is the first work to explore unlabeled samples for prototype fusion in FSL.

B. Transfer Learning via Semantic Knowledge

Semantic knowledge refers to the prior knowledge summarized from human past experiences, which includes class name [54], [55], class description [56], [57], class attributes [17], [58], and class hierarchy [59], [60], [61]. These semantic knowledge is a high-level description of categories and can be shared among different classes, which provides a clear relationship between classes in the form of text description (e.g., Wikipedia articles) or knowledge graph (e.g., WordNet [62] or ConceptNet [63]). Semantic knowledge plays an important role in transfer learning domains, especially on data-limited transfer learning such as zero-shot learning [17], [56], [64], [65] and few-shot learning with semantic knowledge [54], [57], [61], which resorts to the prior of class relationships and enables the system to recognize novel classes. Next, we review them in detail.

1) Zero-Shot Learning: Zero-shot learning (ZSL) [64], [66] is a challenging task, which aims to recognize novel classes without any labeled samples by resorting to the base classes with abundant labeled data. The main idea is to learn a general mapping function between the semantic and the visual prototype on base classes, then apply the mapping to novel classes for class prediction. The semantic spaces in ZSL are typically attribute-based [17], [67], text description-based [55], [56], [64], and class hierarchy-based [60], [68]. For example, Lampert et al. [64], [67] first define the problem of unseen object class detection and propose a between-class attribute transfer strategy to address it. After that, in [17], a structure constraint on visual centers is further proposed for enhancing the learning of mapping function between class attributes and visual prototypes. Our method differs from these models in two key points: i) our method is for addressing the FSL problem, where few labeled samples should be effectively utilized; ii) based on semantic attributes, we propose a novel prototype completion based meta-learning
framework for FSL, instead of directly learning the mapping function.

2) **Semantic-Based Few-Shot Learning:** Recently, some studies also attempt to employ the semantic knowledge to further enhance the performance of meta-learning on FSL [54, 69, 70], called to semantic-based FSL approaches. The idea of such semantic-based FSL is in line with some real-world application scenarios. For example, for the application scenario that the number of object classes is very large. In such scenario, existing some object classes (especially some rare object classes, e.g., elapharus davidianus) only can collect few labeled data because collecting sufficient labelled instances for such a large number of classes is challenging; but the semantic knowledge of such classes is easy to obtain from Wikipedia [12], [54], [57] or knowledge graph (e.g., WordNet [62] or ConceptNet [63]).

To address the semantic-based FSL problem, Chen et al. [12] first introduce word2vec [71] of class names as priors and then propose to fuse visual prototypes and semantic prototype in a convex combination manner for enhancing the performance of inductive FSL. After that, in [57], [59], [72], authors further replace class text description, knowledge graph, class attribution as semantic knowledge, respectively, to explore the potential of different semantic knowledge. Among them, semantic-based FSL with class attributes as semantic knowledge are most related to our method. Their main idea is resorting to the semantic knowledge of class attributes to learn better representations. For example, in [58], an attribute decoupling regularizer was developed based on visual attributes to obtain good representations for images. Hu et al. [72] proposed a compositional feature aggregation module to explore both spatial and semantic visual attributes for FSL. Zou et al. [73] explored compositional few-shot recognition by learning a feature representation composed of important visual attributes. The above existing studies all focus on inductive FSL. Recently, a novel graph-based transductive FSL method with class name as semantic knowledge is proposed to obtain more accurate propagation strategy [74]. Our method belongs to semantic-based FSL, which also explores class attributes as semantic knowledge. However, different from these existing studies, 1) we leverage the semantic knowledge of class attributes to learn a prototype completion strategy. As a result, more accurate prototypes can be obtained for FSL; 2) our method can work on both inductive FSL and transductive FSL.

**C. Prototype Learning**

Prototype learning is first applied to address the FSL problem in [19]. Its main idea is to estimate the prototype of each class by averaging the feature of its all labeled samples and then perform class prediction via a prototype-nearest manner. However, such mean-based prototype is usually not representatitive in data-limited scenario. To address this issue, some recent studies attempt to improve prototype estimation by introducing prototype restore [15], label propagation [53], [74], feature shift [53], and various semantic-based prototype fusion strategies including mean-based fusion [15], hyperparameter-based fusion [59], and learning-based fusion strategies [12].

Besides, due to the simplicity and superiority of prototype learning, the idea of prototype learning also has been extensively explored in long tail learning [75] and zero-shot learning [17], [65], [76], [77]. For example, Wang et al. [75] propose a residual network to model the restore process of prototypes from few-shot to many-shot classes. Fu et al. [76] identify a domain shift problem of ZSL and attempt to address it in a transductive multi-view embedding and prototype manner; Wan et al. [17] propose learning a mapping function from class attributes to prototypes (i.e. visual centers) and a test data-based visual structure constraint to alleviate the hubness [78], [79] and domain shift [76] problem for ZSL. Moreover, some studies also propose to improve ZSL from the perspective of representative prototype estimation to improve ZSL recently. For example, in [80], Zhong et al. propose a hyperparameter-based fusion strategy to blend class visual prototypes and semantic prototypes for improving ZSL; Soravit et al. [81] propose to synthesize novel class prototypes by fusing phantom prototypes in a weighted manner for ZSL. Jiang et al. [82] identify an issue of incomplete or noise semantic knowledge (i.e., the incomplete or noise class descriptions or class attributes), which easily suffers from the domain shift and hubness problem [78], [79] and leads to an issue that the learned class prototypes are less discriminative for ZSL. To address this issue, they propose to improve the less discriminative semantic space for ZSL by using a coupled dictionary learning approach to align the visual-semantic structures [82]; Zhao et al. [83] and Li et al. [84] propose to alleviate the issue in a semantic augmentation manner; Ali et al. [85] attempt to exploit attribute correlation to address the attribution missing issue; Geng et al. [86] propose an ontology-based knowledge representation and semantic embedding method to explore richer and more competitive prior knowledge for ZSL.

Our work also focuses on more representative prototype estimation for FSL, which differs from these existing studies in two aspects: 1) we focus on the incomplete issue of visual prototypes (instead of semantic prototypes) and attempt to address it in a prototype completion manner with the estimated feature distribution of class attributes; and 2) a Gaussian-based prototype fusion strategy is further designed to achieve more representative prototype, i.e., fusing mean-based and completed prototypes from the perspective of distribution estimation, which is different from existing prototype fusion methods [12], [15], [59], [80], [81].

**III. PROTOTYPE COMPLETION WITH PRIMITIVE KNOWLEDGE**

In this section, we first present a formal definition of FSL setting. Second, the proposed prototype completion framework with primitive knowledge (called PCWPK) is introduced. Finally, the three key components in the framework, namely the parts/attribute transfer network, the prototype completion network, and the prototype fusion strategy are elaborated in the last three subsections, respectively.

**A. Problem Definition**

For N-way K-shot FSL problems, we are given two sets: a training set $\mathcal{S} = \{(x_i, y_i)\}_{i=0}^{N \times K}$ with a few of labeled samples
(called support set) and a test set \(Q = \{(x_i, y_i)\}_{i=0}^M\) consisting of unlabeled samples (called query set). Here \(x_i\) denotes the image sampled from the set of novel classes \(C_{novel}, y_i \in C_{novel}\) is the label of \(x_i\), \(N\) indicates the number of classes in \(S\), \(K\) denotes the number of images of each class in \(S\), and \(M\) denotes the number of images in \(Q\). Meanwhile, we also have an auxiliary data set with abundant labeled images \(D_{base} = \{(x_i, y_i)\}_{i=0}^B\), where \(B\) is the number of images in \(D_{base}\), the image \(x_i\) is sampled from the set of base classes \(C_{base}\), i.e., \(y_i \in C_{base}\), and the sets of class \(C_{base}\) and \(C_{novel}\) are disjoint. Our goal is to learn a classifier for the query set \(Q\) on the support set \(S\) and the auxiliary data set \(D_{base}\). We note that the query set \(Q\) is available by regarding it as a set of unlabeled samples to transductive FSL. However, it is not accessible for inductive FSL.

### B. Overall Framework

As shown in Fig. 2, the proposed prototype completion-based meta-learning framework consists of four phases, including pre-training, learning to complete prototypes, meta-training, and meta-test. Next, we detail them respectively.

1) **Pre-Training**: In this phase, following [11], [14], [20], we build and train a convolution neural network (CNN) classifier with the base classes. Then, the last softmax layer is removed and the classifier turns into a feature extractor \(f_{θ_j}(\cdot)\). This produces a good embedding representation for each image.

2) **Learning to Complete Prototypes**: We propose a Prototype Completion Network (ProtoComNet) as a meta-learner. It accounts for complementing the missing attributes for incomplete prototypes. The main details of the ProtoComNet will be elaborated in Section III-D. Here we first give an overview of its workflow depicted in Fig. 2, which includes four steps:

   **Step 1**: We construct primitive knowledge for all classes. The knowledge is what kinds of attributes feature the class should have, e.g., the leopard has four feet and round spot, and zebra has long face and four feet. We note that such kinds of knowledge is very cheap to obtain, e.g., from WordNet [62]. Let \(A = \{a_i\}_{i=0}^{F-1}\) denotes the set of class parts/attributes where \(F\) is the number of attributes, and \(R\) denotes the association matrix between the attributes and the classes, where \(R_{ka} = 1\) if the attribute \(a_i\) is associated with the class \(k\); otherwise \(R_{ka} = 0\). Meanwhile, the semantic embeddings of all classes and attributes are calculated by Glove [87] in an average manner of word embeddings, denoted by \(H = \{h_k\}_{k=0}^{C_{base}} + \{h_{ka}\}_{k=0}^{C_{base}}\). In particular, we split the set of class parts/attributes \(A\) into two subset: \(A_{seen}\) and \(A_{unseen}\) (i.e., \(F = |A_{seen}| + |A_{unseen}|\)). The former \(A_{seen}\) denotes the set of parts/attributes that base classes contains. On the other hand, the latter \(A_{unseen}\) refers to the set of parts/attributes that the novel classes contain but does not appear in base classes.

   **Step 2**: Based on the pre-trained feature extractor \(f_{θ_j}(\cdot)\) and the above primitive knowledge, we extract two types of information as priors, namely base class prototypes and seen part/attribute features. Specifically, the base class prototypes \(p_k^{real}\) can be calculated by averaging the extracted features of all samples in the base class \(k\), that is

   \[
   p_k^{real} = \frac{1}{|D_k^{base}|} \sum_{(x,y) \in D_k^{base}} f_{θ_j}(x),
   \]

   \[\text{(1)}\]

   where \(D_k^{base}\) denotes the set of samples from the base class \(k\).

   As for the feature \(z_{a_i}\) of each seen part/attribute \(a_i \in A_{seen}\), we denote all base class samples that have the corresponding part/attribute \(a_i \in A_{seen}\) in the primitive knowledge as a set \(D_{base}^{a_i}\). Then, we calculate its mean \(μ_{a_i}\), and diagonal covariance \(\text{diag}(σ_{a_i}^2)\) as

   \[
   μ_{a_i} = \frac{1}{|D_{base}^{a_i}|} \sum_{(x,y) \in D_{base}^{a_i}} f_{θ_j}(x),
   \]

   \[\text{(2)}\]

   \[
   σ_{a_i} = \sqrt{\frac{1}{|D_{base}^{a_i}|} \sum_{(x,y) \in D_{base}^{a_i}} (f_{θ_j}(x) - μ_{a_i})^2}.
   \]

   \[\text{(3)}\]
Here, the mean $\mu_{a_i}$ and the diagonal covariance $\text{diag}(\sigma_{a_i}^2)$ characterize the part/attribute feature distribution of each seen part/attribute $a_i \in A^{seen}$, i.e., $z_{a_i} \sim N(\mu_{a_i}, \text{diag}(\sigma_{a_i}^2))$, which will be used in Sections III-C and III-D.

**Step 3:** According to (2) and (3), we can estimate the feature distribution of the seen parts/attributes $a_i \in A^{seen}$. However, the method fails to model the unseen parts/attributes $a_i \in A^{unseen}$ since it does not appear in base classes. To address the drawback, we design a Part/Attribute Transfer Network (PATNet) $f_{\theta}(.)$ with parameters $\theta$, which accounts for inferring the feature distribution of unseen parts/attributes by exploring the semantics relationship between unseen and seen parts/attributes. The intuition behind it is that the similar parts/attributes in semantics should have a similar feature distribution. Its design details will be introduced in Section III-C. Here, we focus on introducing the overall workflow of the PATNet. Specifically, we take the semantic embedding $\{h_{a_i}\}_{i=0}^{A^{seen}-1}$ of all seen parts/attributes $\{a_i\} \in A^{seen}$ as inputs, and treat the feature distribution $N(\mu_{a_i}, \text{diag}(\sigma_{a_i}^2))$ of the seen parts/attributes $a_i \in A^{seen}$ estimated by (2) and (3) as prediction targets, to train the proposed PATNet $f_{\theta}(.)$ by using the Kullback-Leibler (KL) divergence loss. That is

$$\hat{\mu}_{a_i}, \hat{\sigma}_{a_i} = f_{\theta}(h_{a_i}), i = 0,1,\ldots, |A^{seen}| - 1$$

$$\min_{\theta} \mathbb{E}_{a_i \in A^{seen}} KL(N(\hat{\mu}_{a_i}, \text{diag}(\hat{\sigma}_{a_i}^2)), N(\mu_{a_i}, \text{diag}(\sigma_{a_i}^2))))$$  

(4)

where $KL(.)$ denotes the Kullback-Leibler (KL) divergence loss. Then, we train the parts/attributes transfer network $f_{\theta}(.)$ until it converges. The well trained PATNet can infer the feature distribution of each seen and unseen part/attribute through its semantics. As a result, we obtain a new feature distribution $\hat{z}_{a_i} \sim N(\hat{\mu}_{a_i}, \text{diag}(\hat{\sigma}_{a_i}^2))$ for each seen and unseen parts/attribute by utilizing its semantics as input of PATNet, which will be used in Section III-D.

**Step 4:** Upon the results of the previous steps, we mimic the setting of K-shot tasks and construct a set of prototype completion tasks to train our meta-learner $f_{\theta}(.)$ (i.e., ProtoComNet) in an episodic manner [88]. Specifically, in each episode, we first randomly select one class $k$ from base classes $C_{base}$ and $K$ images for the class $k$ from $D_{base}$ as support set $S$. Then, we average the features of all samples in $S$ as the incomplete prototypes $p_k$. Here, we consider it as incomplete because some representative features may be missing. Even though in some cases this may not be true, regarding them as incomplete ones does no harms to our meta-learner. Finally, we take the incomplete prototypes $p_k$, the primitive knowledge (the class-attribute association matrix $R$ and word embedding $H$), and the parts/attributes features $Z = \{z_{a_i}\}_{i=0}^{A^{seen}-1}$ and $\hat{Z} = \{\hat{z}_{a_i}\}_{i=0}^{A^{seen}+A^{unseen}-1}$ as inputs, and treat the base class prototypes $p_k^{\text{real}}$ as outputs, to train our meta-learner by using the Mean-Square Error (MSE) loss. That is

$$\min_{\theta_c} \mathbb{E}_{(p_k, p_k^{\text{real}}) \in T} \text{MSE} \left( f_{\theta_c}(p_k, R, H, Z, \hat{Z}), p_k^{\text{real}} \right)$$  

(5)

where $\theta_c$ denotes the parameters of our meta-learner and $T$ denotes the set of prototype completion tasks.

3) **Meta-Training:** To jointly fine-tune the feature extractor $f_{\theta_f}(.)$ and the meta-learner $f_{\theta_c}(.)$, we construct a number of N-way K-shot tasks from $D_{base}$ following the episodic training manner [88]. Specifically, in each episode, we sample $N$ classes from the base classes $C_{base}$, $K$ images in each class as the support set $S$, and $M$ images as the query set $Q$. Then, $f_{\theta_f}(.)$ and $f_{\theta_c}(.)$ can be further fine-tuned by maximizing the likelihood estimation on query set $Q$. That is

$$\max_{\theta} \mathbb{E}_{(S,Q) \in T'} \sum_{(x,y) \in Q} \log(P(y|x, S, R, H, Z, \hat{Z}, \theta))$$  

(6)

where $\theta = \{\theta_f, \theta_c\}$ and $T'$ denotes the set of N-way K-shot tasks. Specifically, for each episode, we first estimate its class prototype $p_k$ by averaging the features of the labeled samples. That is

$$p_k = \frac{1}{|S_k|} \sum_{x \in S_k} f_{\theta_f}(x)$$  

(7)

Moreover, to obtain more reliable prototypes, we further explore unlabeled samples and combine $p_k$ and $p_k'$ by introducing a Gaussian-based prototype fusion strategy (which will be introduced in Section III-E). As a result, the fused prototype $p_k'$ is obtained. Finally, the probability of each sample $x \in Q$ to be class $k$ is estimated based on the proximity between its feature $f_{\theta_f}(x)$ and $p_k'$. That is

$$P(y = k|x, S, R, H, Z, \theta) = \frac{e^{d(f_{\theta_f}(x), p_k')} \cdot \gamma}{\sum_c e^{d(f_{\theta_f}(x), p_k')} \cdot \gamma}$$  

(9)

where $d(.)$ denotes the cosine similarity of two vectors and $\gamma$ is a learnable scale parameter.

4) **Meta-Test:** Following (7)–(9), we directly perform few-shot classification for novel class prediction.

C. **Part/attribute Transfer Network**

In this subsection, we introduce the first key component of learning to complete prototypes (Step 3 in Section III-B2), namely the PATNet $f_{\theta}(.)$. Our intuition is that the similar parts/attributes in semantics should have a similar feature distribution. Thus, we directly treat the semantic embeddings $\{h_{a_i}\}_{i=0}^{F-1}$ of part/attribute as input and the parts/attributes distribution $N(\mu_{a_i}, \text{diag}(\sigma_{a_i}^2))$ as output to build the PATNet.

As shown in Fig. 3, the network consists of an embedding layer $f_{\theta_{pe}}(.)$ and an inference layer $f_{\theta_{pi}}(.)$, where $\theta_{pe}$ and $\theta_{pi}$ denote their parameters, respectively. Here, the former aims to map each semantic embeddings to a new embedding space, and then the latter accounts for estimating the feature distribution of each part/attribute. Next, we detail them, respectively.

**Embedding Layer:** We take the semantic embedding $h_{a_i}$ of each part/attribute as input of the embedding layer $f_{\theta_{pe}}(.)$, and then project the semantic embedding $h_{a_i}$ to a new embedding
...space. As a result, the new embedding $h'_a$ can be obtained. That is,

$$h'_a = f_{\theta_{pa}}(h_a).$$

**Inference Layer:** Based on the new embedding $h'_a$, we employ an inference layer consisting of a mean module and a diagonal covariance module to predict the distribution of each seen and unseen part/attribute, which is characterized by a multivariate normal distribution parameterized with its mean $\mu_a$ and diagonal covariance $\sigma_a$.

That is,

$$\mu_a, \sigma_a = f_{\theta_{in}}(h'_a),$$

$$z_a \sim N(\mu_a, \sigma_a).$$

(11)

Note that $\theta_{in}$ contains the two parameters $\theta_{pe}$ and $\theta_{pt}$.

**D. Prototype Completion Network**

In this subsection, we introduce how the ProtoComNet $f_{\theta_{c}}()$ is designed, which is the second key component for learning to complete prototypes (Step 4 in Section III-B2). Our intuition is that the parts/attributes feature can be transferred from base classes to novel classes for prototype completion. For example, even if human haven’t seen “zebra”, they can also imagine its visual features of “long face” once they learn this knowledge from “kangaroo” and “horse”. Thus, we treat the primitive knowledge ($R$ and $H$), part/attribute features $Z$ and $Z$ and the incomplete prototypes $p_k$ as input and the completed prototypes $\hat{p}_k$ as output, and then build an encoder-aggregator-decoder network, as shown in Fig. 4. Here, the encoder aims to form a low-dimensional representation for prototypes and part/attributes. Then, the aggregator accounts for evaluating the importance of different parts/attributes and combining them with a weighted sum. Finally, the decoder is in charge of the prediction of complete prototypes $\hat{p}_k$. Next, we detail them, respectively.

**The Encoder:** In the training part, the encoding process involves a sampling step of an attribute feature $z_a$, from its distribution $N(\mu_a, diag(\sigma_a^2))$ or $N(\mu_a, diag(\sigma_a^2))$, followed by an encoder $g_{\theta_{pe}}()$ that encodes the attribute feature $z_a$ and the estimated prototypes $p_k$ to a latent code $z'_a$ and $z'_k$, respectively. Intuitively, the estimation $N(\mu_a, diag(\sigma_a^2))$ is more accurate than $N(\mu_a, diag(\sigma_a^2))$ for each seen part/attribute $a_i \in A_{seen}$ since the former is directly calculated by using visual feature (see (2) and (3)) but the latter is inferred by using the semantic embedding of each part/attribute. Therefore, the feature distribution $N(\mu_a, diag(\sigma_a^2))$ is a good choice for all seen parts/attributes, but it will affect the generalization of inferred feature distribution $N(\mu_a, diag(\sigma_a^2))$ for unseen parts/attributes $a_i \in A_{unseen}$ if we only use the feature distribution $N(\mu_a, diag(\sigma_a^2))$ as inputs in the training phase. To alleviate the generalization issue, in the training phase, we propose a random selecting strategy with a probability $\rho = 0.5$ to sample the attribute feature $z_a$, from the feature distribution $N(\mu_a, diag(\sigma_a^2))$ and the feature distribution $N(\mu_a, diag(\sigma_a^2))$. The overall process can be expressed as

$$z_a \sim \begin{cases} N(\mu_a, diag(\sigma_a^2)) & a_i \in A_{seen} \& r < \rho, \\ N(\mu_a, diag(\sigma_a^2)) & \text{otherwise} \end{cases},$$

$$z'_a = g_{\theta_{pe}}(z_a),$$

$$z'_k = g_{\theta_{pe}}(z_k),$$

(12)

where $\theta_{pe}$ denotes the parameters of the encoder and $r$ is a random number from 0 to 1. Note that, in the meta-test phase, we regard $N(\mu_a, diag(\sigma_a^2))$ as the feature distribution of seen parts/attributes and $N(\mu_a, diag(\sigma_a^2))$ as the ones of unseen parts/attributes; and we remove the sampling step and use the mean $\mu_a$ and $\mu_a$ to replace $z_a$.

**The Aggregator:** Intuitively, different parts/attributes make varying contributions to distinct classes, for example, the “nose” is more representative for elephants than tigers to complete their prototypes. Hence, differentiating their contributions in the completion is important. To this end, we employ an attention-based aggregator $g_{\theta_{at}}()$. Here, we calculate the attention weights $\alpha_{ka_i}$ by using the semantic embeddings $h_k$ and $h_a_i$ of the class $k$ and the attribute $a_i$, and the incomplete prototypes $p_k$. Then, we apply them to combine the latent codes $z'_a$ and $z'_k$, and obtain the aggregated result $g_k$ as follows:

$$\alpha_{ka_i} = R_{ka_i}g_{\theta_{at}}(p_k || h_k || h_{a_i}),$$

$$g_k = \sum_{a_i} \alpha_{ka_i} z'_a + z'_k,$$

(13)
where $\theta_{ca}$ is the parameters of the aggregator and $||$ is a concatenation operation.

**The Decoder:** Finally, we use the aggregated result $g_k$ to decode the complete prototypes $\hat{p}_k$ for each class $k$ by the decoder module $g_{θ_{cd}}()$. That is

$$\hat{p}_k = g_{θ_{cd}}(g_k),$$

(14)

where $θ_{cd}$ denotes the parameters of the decoder.

### E. Prototype Fusion Strategy

Till now, we have two prototype estimations, i.e., the mean-based prototypes $p_k$ and the completed prototypes $\hat{p}_k$. Next, we will discuss why and how to fuse these two estimations from the perspective of Bayesian estimation.

1) Why Do We Fuse Prototypes?: Actually, both the estimates $p_k$ and $\hat{p}_k$ have their own biases. The former is mainly due to the scarcity or incompleteness of labeled samples in novel classes, which produces biased means; while the latter is brought by primitive knowledge noises or class differences. The fact implies that the two estimates can remedy each other. When the labeled samples are very scarce and incomplete, the completed prototypes $\hat{p}_k$ are more reliable because the completion is learned from a great number of base class tasks. As more and more labeled samples become available, the mean-based prototypes are more representative because the ProtoComNet may result in completion error problem under the effects of primitive knowledge noises or class differences.

Fig. 5(a) shows an 5-way $K$-shot example to demonstrate this point. From the figure, we observe that the completed prototypes are more accurate on 1/2-shot tasks while the mean-based ones are better on 3/4/5-shot tasks. Thus, a prototype fusion strategy is desired to combine their advantages and form more representative prototypes.

2) How to Fuse Prototypes? : We apply the Bayesian estimation to fuse the two kinds of prototypes. Specifically, we assume that the estimated prototypes $p_k$ and $\hat{p}_k$ follow a Multivariate Gaussian Distribution (MGD). The first assumption (i.e., the mean-based prototype estimation $p_k$ follows an MGD) is reasonable and intuitive since the samples of each base/novel class in the pre-trained space are continuous and are clustered as a compact group (shown in Fig. 1). As for the completed prototype estimation $\hat{p}_k$, it is the output of $p_k$, i.e., the distribution of prototype estimation $p_k$ almost be the product of $p_k$ and another MGD. According to the proposition 1) of Appendix A, available online, i.e., the product of two MGD still obeys an MGD, we can prove that assuming the prototype estimation $\hat{p}_k$ also follows a MGD (i.e., the second assumption) is also reasonable.

Based on the above assumption, $p_k$ can be regarded as a sample from the MGD with mean $μ_k$ and diagonal covariance $diag(σ_k^2)$, i.e., $N(μ_k, diag(σ_k^2))$. Likewise, $\hat{p}_k$ is a sample from $N(\hat{μ}_k, diag(\hat{σ}_k^2))$ with mean $\hat{μ}_k$ and diagonal covariance $diag(\hat{σ}_k^2)$. As shown in Fig. 5(b), from the view of Bayesian estimation, we regard the distribution $N(\hat{μ}_k, diag(\hat{σ}_k^2))$ as a prior, and treat the distribution $N(μ_k, diag(σ_k^2))$ as the conditional likelihood of observed few labeled samples. Then, the Bayesian estimation of fused prototypes can be expressed as their product, i.e., a posterior MGD $N(μ'_k, diag(σ'_k^2))$ with mean $μ'_k = \frac{σ_k^2 \hat{μ}_k + σ_k^2 μ_k}{σ_k^2 + σ_k^2}$ and diagonal covariance $diag(σ'_k^2) = diag(\frac{σ_k^2 \hat{σ}_k^2 + σ_k^2 σ_k^2}{σ_k^2 + σ_k^2})$, where $∘$ is element-wise product (Please refer to Appendix A, available online for its derivations). Finally, we take the mean $μ'_k$ as the fused prototypes $\mu'_k$ to solve the few-shot tasks (Please refer to Section III-E4 for its theoretic analysis).

In this paper, we term the overall Bayesian estimation procedure as Gaussian-based prototype fusion strategy (Gauss-Fusion). We can see that $μ'_k$ is determined by four unknown variables $μ_k$, $σ_k$, $μ'_k$, and $σ'_k$. Next, we introduce four types of methods to estimate them.

3) How to Estimate $μ_k$, $σ_k$, $μ'_k$, and $σ'_k$?: In this part, we discuss four methods to estimate the four unknown variables $μ_k$, $σ_k$, $μ'_k$, and $σ'_k$, including i) assumption-based estimation method, ii) two-step estimation method, iii) EM-based estimation method, and iv) improved EM-based estimation method. Among them, the methods i) and ii) belong to non-iterative approaches, where the former follows the estimate strategy proposed in [15] and the latter is our conference strategy [18]. The rest of these methods (i.e., the methods iii) and iv) ) are all iterative approaches, which are newly-developed in this paper.

**Assumption-Based Estimation Method:** The Mean-based Prototype Fusion (MeanFusion) strategy proposed in [15] regards the averaged prototypes as the fused prototypes $\mu'_k = 0.5(p_k + \hat{p}_k)$. This strategy can be considered as a special case of our GaussFusion, where we assume that the two means satisfy $μ_k = p_k$ and $μ'_k = p_k$, and the two diagonal covariance is also equal, i.e., $σ_k = \hat{σ}_k$. However, the assumption is too strong to fit the real prototype distribution. Thus, the performance improvement of the MeanFusion is limited for FSL.

**Two-Step Estimation Method:** Inspired by transductive FSL [53], we propose to estimate the four variables by leveraging the unlabeled samples in a two-step manner: Step 1) we calculate the probability of each sample $x \in S \cup Q$ belonging to class $k$ by regarding $p_k$ and $\hat{p}_k$ as the prototypes, respectively. For example, when we take $p_k$ as the prototypes, the probability of each unlabeled sample $x \in Q$ can be computed as

$$P(y = k|x) = \frac{e^{d(f_{θ_{cy}}(x), p_k)} \cdot λ}{\sum_{c} e^{d(f_{θ_{cy}}(x), p_c)} \cdot λ},$$

(15)

where $d(\cdot)$ denotes the cosine similarity of two vectors and $λ$ is a hyper-parameter. Following [20], $λ = 10$ is used. As for each labeled sample $x \in S$, the probability turns into a one-hot vector
by its labels. \( P(y = k|x) \) can be computed in a similar manner by using prototypes \( \hat{p}_k \). Step 2) we take \( P(y = k|x) \) as sample weights and estimate the mean \( \mu_k \) and the diagonal covariance \( \text{diag}(\sigma_k^2) \) of each prototype distribution in a weighted average manner. That is

\[
\mu_k = \frac{1}{\sum_{x \in \mathcal{S} \cup \mathcal{Q}} P(k|x)} \sum_{x \in \mathcal{S} \cup \mathcal{Q}} P(k|x) f_{\theta_j}(x), \quad (16)
\]

\[
\sigma_k = \left( \frac{1}{\sum_{x \in \mathcal{S} \cup \mathcal{Q}} P(k|x)} \sum_{x \in \mathcal{S} \cup \mathcal{Q}} P(k|x)(f_{\theta_j}(x) - \mu_k)^2 \right). \quad (17)
\]

Similarly, the mean \( \hat{\mu}_k \) and the diagonal covariance \( \text{diag}(\hat{\sigma}_k^2) \) can be calculated in a similar manner by regarding \( P(y = k|x) \) as sample weights. The two step prediction strategy is the method proposed in our conference version [18].

**EM-Based Estimation Method:** The EM (Expectation-Maximization) algorithm [89] is a widely used parameter estimation method, which adopts an iterative strategy to polish the parameter estimation. Thus, we attempt to estimate the above four variables by employing the EM algorithm. Specifically, we regard the support and query samples \( x \in \mathcal{S} \cup \mathcal{Q} \) as the observation data from Gaussian mixture distribution with unknown mean \( \mu_k \) or \( \hat{\mu}_k \) and diagonal covariance \( \text{diag}(\sigma_k) \) or \( \text{diag}(\hat{\sigma}_k) \) \( (k = 0, 1, \ldots, N - 1) \), and regard the prototypes \( p_k \) or \( \hat{p}_k \) as the initial mean of the \( k \)-th Gaussian distribution. Our goal is to fit the mean and diagonal covariance to the observation data \( x \in \mathcal{S} \cup \mathcal{Q} \). That is, maximizing the likelihood estimate for \( \mu_k \) and \( \sigma_k \) (Note that \( \hat{\mu}_k \) and \( \hat{\sigma}_k \) are similar) as

\[
l((\mu_k, \sigma_k)^{N-1}) = \prod_{k=0}^{N-1} \sum_{x \in \mathcal{S} \cup \mathcal{Q}} z \cdot N(x; \mu_k, \text{diag}(\sigma_k^2)), \quad (18)
\]

where \( z \) is a hidden variable denoting the posterior probability that \( x \) belongs to class \( k \).

We adopt EM algorithm to optimize \( (18) \), which includes following three steps: 1) initializing the mean \( \mu_k \) or \( \hat{\mu}_k \) by using the prototypes \( p_k \) or \( \hat{p}_k \) and diagonal covariance \( \sigma_k \) or \( \hat{\sigma}_k \) in a constant (We empirically find that our method can obtain high classification performance when it is set as 35); 2) Performing E step to estimate the posterior probability \( z \) that a given observation \( x \) belongs to a given class \( k \) by using the probability density function \( N(x; \mu_k, \text{diag}(\sigma_k^2)) \) or \( N(x; \hat{\mu}_k, \text{diag}(\hat{\sigma}_k^2)) \). Note that we estimate the probability of each support sample \( x \in \mathcal{S} \) by a one-hot vector of its label since its label is known; 3) Performing M step to maximize the posterior probability and find the optimal mean \( \mu_k \) or \( \hat{\mu}_k \) and diagonal covariance \( \sigma_k \) or \( \hat{\sigma}_k \); 4) Repeatedly carrying out these two steps (i.e., E step and M step) until convergence. Finally, we take the resulting \( \mu_k \) or \( \hat{\mu}_k \) and \( \sigma_k \) or \( \hat{\sigma}_k \) as our estimation.

**Improved EM-Based Estimation Method:** In the above EM-based method, the posterior probability \( z \) is estimated by using the Gaussian probability density. Its calculation is similar to the Mahalanobis distance. However, recent studies [14, 20] found that the cosine distance-based classifier show better performance on the estimation of posterior probability \( z \) for FSL. Inspired by this fact, we estimate it by leveraging the cosine-based classifier (i.e., (15)). In particular, the improved EM-based method can be regarded as an extension of the above Two-Step Method by using the EM algorithm. Specifically, we first initialize the mean \( \mu_k \) or \( \hat{\mu}_k \) by using the prototypes \( p_k \) or \( \hat{p}_k \). Second, the step 1) (described in Two-Step Estimation Method) can be regarded as an E-Step, i.e., regarding the mean \( \mu_k \) or \( \hat{\mu}_k \) as the prototypes of cosine classifier and then estimating the posterior probability \( z \) that a given observation \( x \) belongs to a given class \( k \). This is done by using (15). Third, the step 2) can be regarded as an M-Step, i.e., maximizing the posterior probability to find the optimal mean \( \mu_k \) or \( \hat{\mu}_k \) and diagonal covariance \( \text{diag}(\sigma_k) \) or \( \text{diag}(\hat{\sigma}_k) \). This is done according to (16) and (17). Finally, the above two steps are repeated until convergence. Here, we denote the number of iteration as a hyper-parameter \( n_{\text{iter}} \) and empirically find that setting it to 6 is sufficient to converge. For clarity, we summarize the improved EM-based method in Appendix B, available online.

4) **Theoretic Analysis:** Here, we provide a brief theoretic analysis on the Gaussian-based prototype fusion strategy described in Section III-E2. By the strategy, we can obtain five estimations, i.e., \( \hat{p}_k, \mu_k, \hat{\mu}_k, p_k, \) and \( \hat{p}_k \). Next, we analyze why the prototypes \( \hat{p}_k \) produced by the prototype fusion strategy are better.

**Proposition 1:** \( \mu_k \) (\( \hat{\mu}_k \)) is more representative than \( p_k \) (\( \hat{p}_k \)).

**Proof:** We take \( \mu_k \) and \( p_k \) as an example to prove the Proposition 1. The proof for \( \hat{\mu}_k \) and \( \hat{p}_k \) is similar. Let us first revisit how are the variables \( \mu_k \) and \( \sigma_k \) estimated. In these EM-based fusion parameter estimation methods, the estimation of \( \mu_k \) and \( \sigma_k \) is regarded as a fitting problem of observation data \( x \in \mathcal{S} \cup \mathcal{Q} \) with a \( N \)-components Gaussian mixture model. Thus, our goal is to optimize the \( N \)-components parameters \( \psi^{t+1} = \{p^{t+1}_k, \sigma^{t+1}_k\}_{k=0}^{N-1} \) iteratively by maximizing the log-likelihood \( L(\psi^{t}) \)

\[
\max_{\psi^t} L(\psi^t) = \log \prod_{x \in \mathcal{S} \cup \mathcal{Q}} P(x|\psi^t) = \sum_{x \in \mathcal{S} \cup \mathcal{Q}} \log \sum_{k=0}^{N-1} P(x, k|\psi^t), \quad (19)
\]

where \( k \) is the label of \( k \)-th Gaussian components. As our solution follows the EM optimization, we have \( L(\psi^{t+1}) \geq L(\psi^{t}) \). This means that each iteration of the improved EM-based algorithm increases the log likelihood \( L(\psi_t) \), i.e., the parameters \( \psi_t \) is more effective than \( \psi_t^{t+1} \) for fitting observation data \( x \in \mathcal{S} \cup \mathcal{Q} \). Thus, the variable \( \mu_k \) obtained by the improved EM-based methods is more representative than the initial variable \( p_k \).

**Proposition 2:** \( \hat{\mu}_k \) is more representative than \( \mu_k \), and \( \hat{\mu}_k \).

**Proof:** Let us revisit the fused prototype distribution, i.e., the posterior MGD \( N(\hat{\mu}_k, \text{diag}(\hat{\sigma}_k^2)) \). Here, \( \hat{\sigma}_k^2 = \frac{\sigma_k^2 + \sigma_k^2}{\sigma_k^2 + \sigma_k^2} \) denotes the estimation variance of prototypes \( \hat{\mu}_k \) (Note that we assume the covariance is diagonal). Then, we have the two inequalities since these terms \( \frac{\sigma_k^2}{\sigma_k^2 + \sigma_k^2} \) and \( \frac{\sigma_k^2}{\sigma_k^2 + \sigma_k^2} \) are always greater than or equal to 0

\[
\sigma_k^2 = \sigma_k^2 \odot \hat{\sigma}_k^2 = \sigma_k^2 - \frac{\sigma_k^2}{\sigma_k^2 + \sigma_k^2} \leq \hat{\sigma}_k^2, \quad (20)
\]

\[
\hat{\sigma}_k^2 = \sigma_k^2 \odot \hat{\sigma}_k^2 = \sigma_k^2 - \frac{\hat{\sigma}_k^2}{\sigma_k^2 + \sigma_k^2} \leq \hat{\sigma}_k^2, \quad (21)
\]
where the right equation is satisfied only when $\sigma_i^2$ or $\hat{\sigma}_i^2$ is zero. The (20) and (21) imply that the variance of prototypes $\hat{\mu}_k$ decreases for each class $k$ after fusing $\mu_k$ and $\hat{\mu}_k$. Thus, $\hat{\mu}_k$ is more representative than $\mu_k$ and $\hat{\mu}_k$.

Based on the above propositions 1 and 2, we know that $\hat{\mu}_k$ is more representative than $\mu_k$, $\hat{\mu}_k$, $p_k$, and $\hat{p}_k$. Hence, we take the mean $\hat{\mu}_k$ as the final fused prototype $\hat{p}_k$.

IV. AN ECONOMIC VERSION: PROTOTYPE COMPLETION WITHOUT PROMITIVE KNOWLEDGE

Till now, we have introduced all details of our prototype completion framework with primitive knowledge. Its advantage is that incomplete prototypes can be effectively completed with the help of primitive knowledge. As a result, a more representative prototype can be obtained for FSL. However, acquiring such primitive knowledge may be difficult or impossible for some applications, for example fine-grained classification since it needs more fine attributes/parts annotations. To address this issue, in this section, we develop an economic prototype completion version (called PCWOPK) where a more representative prototype can be obtained without primitive knowledge. Next, we introduce the proposed PCWOPK method. For clarity, we mainly introduce the differences between the economic version and the previous version described in Section III.

A. Overall Framework

The overall framework of economic prototype completion without primitive knowledge is similar to that with primitive knowledge described in Fig. 2. The only difference is the design of “Learning to Complete Prototypes” phase, where we remove the Step 1 and Step 3, and revise the Step 2. Our main insight of the new “Learning to Complete Prototypes” phase without primitive knowledge is that 1) the local visual features (e.g., eye, foot, head, or tail image patches) in each image implicitly reflect latent attributes/parts collected in our primitive knowledge, which provide discriminative and transferable primitive visual features across base and novel classes; and 2) these similar local visual features of different classes and images can be grouped together as a cluster. Thus, the key challenge is how to represent these local visual features and then leverage an unsupervised method to discover these latent cross-categories shared attribute/part clusters without the help of primitive knowledge.

To this end, we propose to use dense representations to embed these local features for each image from all base classes and then apply a K-means based clustering method to these local features for acquiring these shared attribute/part clusters. As shown in Fig. 6, the “Learning to Complete Prototypes” phase consists of two steps:

Step A: Similar to the Step 2 of Section III-B2, our goal is still extracting two types of information as priors, namely base class prototypes and latent shared/transfered parts/attribute features, by resorting to the pre-trained feature extractor $f_{\theta_j}()$ in the pre-training phase. The calculation detail of base class prototypes $p_k^{real}$ is the same as (1). The difference mainly lies in the calculation of latent part/attribute features $Z$ since we assume that the parts/attributes are unknown for each class, i.e., the primitive knowledge is unknown. To calculate the part/attribute features, we first remove the last global pooling layer of feature extractor $f_{\theta_j}()$ to extract the dense representation $\Psi_x$ (i.e., an $h \times w \times d$ feature map which can be viewed as a set of $m$ ($m = h \times w$) $d$-dimensional local features) for each image $x \in D_{base}$. As a result, a set of local features $\Psi = \bigcup_{x \in D_{base}} \Psi_x$ from all the samples of base classes can be obtained. That is

$$\Psi = \{\psi^0_x, \psi^1_x, \ldots, \psi^{D_{base}}_{x} \mid \times m-1 \} \in \mathbb{R}^{|D_{base}| \times m \times d}, \ (22)$$

where $h$ and $w$ denote the size of feature map, and $|\cdot|$ denotes the size of set. Then, based on all local features $\Psi$ from all base sample $x \in D_{base}$, we conduct clustering on all local features $\Psi$ by using the K-means method. As a result, $F$ clusters can be obtained, which can be regarded as the feature distribution $Z = \{z_{a_i} \sim N(\mu_{a_i}, \text{diag}(\sigma_{a_i}^2))\}$ of latent shared/transfered parts/attributes $\{a_i\}_{i=1}^{F-1}$. Here, the mean $\mu_{a_i}$ and diagonal covariance $\text{diag}(\sigma_{a_i}^2)$ of each shared/transfered parts/attributes $a_i$ can be calculated as

$$\mu_{a_i} = \frac{1}{|\Psi_{a_i}|} \sum_{\psi \in \Psi_{a_i}} \psi, \ \ (23)$$

$$\sigma_{a_i} = \sqrt{\frac{1}{|\Psi_{a_i}|} \sum_{\psi \in \Psi_{a_i}} (\psi - \mu_{a_i})^2}, \ \ (24)$$

where $\Psi_{a_i}$ denotes the set of local features from cluster $a_i$. Fig. 6. The workflow of “Learning to Complete Prototypes” phase in the economic prototype completion framework without primitive knowledge.
Step B: The step is similar to the Step 4 of Section III-B2. The only difference is the input, i.e., we remove the primitive knowledge and replace the parts/attributes features with the ones obtained in (23) and (24).

B. Prototype Completion Network

The structure of ProtoComNet is also similar to previous version described in Fig. 4. The only difference is the input of our encoder and aggregator. As shown in Fig. 7, for the encoder, we remove these unseen attributes and replace the feature distribution of seen attributes by (23) and (24) since these unseen attributes are unknown in case of missing primitive knowledge. For the aggregator, due to the absence of primitive knowledge, we cannot access the semantic knowledge of latent shared/transferred parts/attributes \( \{a_i\}_{i=0}^{F-1} \). To address this issue, we remove the semantic embeddings \( H \) of aggregator and replace it by using the part/attribute feature \( z_{a_i} \sim N(\mu_{a_i}, \text{diag}(\sigma_{a_i}^2)) \). The intuition of such design is that resorting to the part/attribute feature \( z_{a_i} \) to learn the completion weight \( \alpha_{ka_i} \) for each part/attribute \( a_i \) in a data-driven manner.

V. PERFORMANCE EVALUATION

In this section, we evaluate the proposed framework on general and fine-grained few-shot classification tasks, and then discuss the experiment results and present our statistical analysis, ablation study, and visualization in details.

A. Datasets Settings and Knowledge Construction

1) Datasets Settings: MiniImagenet: The data set is a subset of ImageNet, which includes 100 classes and each class consists of 600 images. Following [15], we split the data set into 64 classes for training, 16 classes for validation, and 20 classes for test, respectively. The class parts/attributes are extracted from WordNet by using the relation of “part_holonyms()”. Please refer to Section V-A2 for more construction details.

2) Primitive Knowledge Construction Details: WordNet [62] is a common knowledge graph, which contains abundant entity information (e.g., entity definition) and detailed entity relations (e.g., hyponymy relations and meronymy relations). Resorting to the hyponymy relations and meronymy relations in WordNet [62], we construct the primitive knowledge for each base/novel classes of miniImagenet and tieredImagenet datasets. Specifically, as shown in Algorithm 1, given a base/novel class \( k \in C_{base} \cup C_{novel} \), we first identify its entity \( E_k \) corresponding to WordNet (Line 1). Then, we leverage the meronymy relations (i.e., \( \text{part}_\text{holonyms}() \)) to initialize the part/attribute set \( A_k \) of the class \( k \) (Line 2). After that, we traverse all hypernyms \( E^+_k \) of entity \( E_k \) by using the relation of “hypernyms()” (Lines 3~5) and update the part/attribute set \( A_k \) by using the meronymy relations (i.e., \( \text{part}_\text{holonyms}() \)). Finally, the set of parts/attributes \( A_k \) contained in the class \( k \) is obtained. The intuition behind such construction strategy is that these parts/attributes contained in the hypernym \( E^+_k \) of entity \( E_k \) should also appear in the entity \( E_k \) (i.e., the class \( k \)).

B. Implementation Details

Architecture: Following [14], we employ ResNet12 as the feature extractor. In PATNet, we use a single-layer MLP with 512 units for the embedding layer, and a two-layer MLP with 512-dimensional hidden units for the mean module and diagonal covariance module, respectively. In ProtoComNet, we use a
single-layer MLP with 256 units for the encoder, a two-layer MLP with a 300-dimensional hidden layer for the aggregator, and a two-layer MLP with 512-dimensional hidden layers for the decoder. Here, ReLU is used as the activation function for all network. The number of iteration, namely $n_{iter}$, is set to 6 for GaussFusion. Besides, the number $F$ of cluster in the economic prototype completion version is set to 200 on the above all datasets.

### Training Details:
We first pre-train the feature extractor with 100 epochs on base classes via an SGD with momentum of 0.9 and weight decay of 0.0005. The learning rate is initially set to 0.1, and then decayed by 0.1 at epochs 60, 80, and 90, respectively. Second, we train the PATNet with 20000 epochs by using an Adam with weight decay of 0.0005. The learning rate is initially set to 0.001, and then decayed by 0.1 at 10000 epochs. Third, we train the ProtoComNet with 100 epochs in an episodic manner by using an SGD with momentum of 0.9 and weight decay of 0.0005. The learning rate is initially set to 0.1, and then changed at epochs 15, 40, and 80. Finally, we fine-tune all modules with 40 epochs in an episodic manner. The learning rate is initially set to 0.01, and then decayed by 0.1 at epochs 15, 25, and 30.

### Evaluation:
We conduct few-shot classification on 600 randomly sampled episodes from the test set and report the mean accuracy together with the 95% confidence interval. In each episode, we randomly sample 15 query images per class for evaluation in 5-way 1-shot/5-shot tasks.

### C. Discussion of Results
For a comparison, some state-of-the-art approaches are also applied to the few-shot classification and few-shot fine-grained classification tasks as baselines. These methods can be roughly from five types, i.e., metric-based, optimization-based, semantics-based, graph-based, and pre-training based approaches. For a fair comparison, 1) we employ the MeanFusion and GaussFusion strategy to evaluate the performance of our framework on inductive and transductive FSL setting, respectively; and 2) we report the performance of our two prototype completion versions.

1) In Few-Shot Classification: Table I shows the results of our two methods (i.e., PCWOPK and PCWPK) and baseline methods on miniImagenet and tieredImageNet. It can be found that 1) comparing with these FSL methods without external knowledge, our PCWOPK achieves superior or comparable performance; and 2) our PCWPK method also achieves superior performance over these FSL methods with external knowledge, on both inductive and transductive FSL settings. This verifies the effectiveness of our methods.

### Table I: Experiment Results on the MinImageNet and TieredImageNet Data Sets

| Setting       | Method            | Type         | Backbone | Use Knowledge | miniImagenet  | tieredImageNet |
|---------------|-------------------|--------------|----------|---------------|---------------|----------------|
|               |                   |              |          |               | 5-way 1-shot  | 5-way 5-shot   | 5-way 1-shot  | 5-way 5-shot   |
|               | RestoreNet [15]   | Metric       | ResNet18 | No            | -             | -              | -             | -              |
|               | ConstellationNet [91] | Metric       | ResNet12 | No            | -             | -              | -             | -              |
|               | RAP-ProtoNet [52] | Optimization | ResNet10 | No            | -             | -              | -             | -              |
|               | MAML [8]          | Optimization | ResNet12 | No            | -             | -              | -             | -              |
|               | MetaOptNet [28]   | Optimization | ResNet12 | No            | -             | -              | -             | -              |
|               | ALFA [92]         | Optimization | ResNet12 | No            | -             | -              | -             | -              |
|               | AM3-TRAML [54]    | Semantics    | WD       | -             | -             | -              | -             | -              |
|               | MultiSem [57]     | Semantics    | Dense-211| CD            | 67.3%         | -              | -             | -              |
|               | FSLRT [59]        | Semantics    | ConvNet28| CH+WD        | 64.4%         | 74.1%          | -             | -              |
|               | CITA [72]         | Semantics    | ResNet18 | CA            | 63.21%        | 79.84%         | -             | -              |
|               | META-Baseline [14] | Pre-training | ResNet12 | No            | -             | -              | -             | -              |
|               | Neg-CoNisse [36]  | Pre-training | ResNet12 | No            | -             | -              | -             | -              |
|               | CentAlign [57]    | Pre-training | ResNet18 | No            | -             | -              | -             | -              |
|               | Maximum Likelihood + DC [38] | Pre-training | WRN-28-10 | No          | 66.91%        | 70.74%         | -             | -              |
|               | SVM + DC [38]     | Pre-training | WRN-28-10 | No            | -             | -              | -             | -              |
|               | LogitReg + Reg. + DC [38] | Pre-training | WRN-28-10 | No            | -             | -              | -             | -              |
|               | PCWOPK (MeanFusion) | Semantics   | ResNet12 | No            | 69.68 ± 0.76% | 81.65 ± 0.54% | 75.19 ± 0.90% | 86.09 ± 0.60% |
|               | PCWPK (MeanFusion) | Semantics   | ResNet12 | No            | 72.49 ± 0.95% | 85.66 ± 0.63% | 72.49 ± 0.95% | 85.66 ± 0.63% |
|               | PCWOPK (Two-Step) | Semantics   | ResNet12 | No            | 61.14 ± 0.22% | -               | -             | -              |
|               | PCWPK (Two-Step)  | Semantics   | ResNet12 | No            | 67.77 ± 0.32% | 84.60 ± 0.43% | 72.45 ± 0.51% | 87.24 ± 0.33% |
|               | PCWOPK (EM)       | Semantics   | ResNet12 | No            | 65.80 ± 0.83% | 83.07 ± 0.74% | 71.21 ± 0.87% | 85.98 ± 0.98% |
|               | PCWPK (Improved EM) | Pre-training | ResNet12 | No            | 70.48 ± 0.38% | 85.42 ± 0.46% | 73.59 ± 0.45% | 88.13 ± 0.28% |
|               | PCWPK (Two-Step)  | Semantics   | ResNet12 | No            | 73.9 ± 0.31%  | 85.0 ± 0.31%  | 79.9 ± 0.31%  | 88.5 ± 0.31%  |
|               | PCWPK (EM)        | Semantics   | ResNet12 | No            | 67.73 ± 0.68% | 84.90 ± 0.72% | 73.84 ± 0.71% | 86.50 ± 0.95% |
|               | PCWPK (Improved EM) | Pre-training | ResNet12 | No            | 70.0 ± 0.65%  | 72.4 ± 0.44%  | -             | -              |
|               | PCWPK (Two-Step)  | Semantics   | ResNet12 | No            | 72.11 ± 0.13% | 82.31 ± 0.14% | 78.98 ± 0.21% | 86.39 ± 0.16% |
|               | PCWPK (EM)        | Semantics   | ResNet12 | No            | 74.22 ± 0.20% | 84.51 ± 0.15% | 80.56 ± 0.35% | 87.93 ± 0.35% |
|               | PCWPK (Improved EM) | Pre-training | ResNet12 | No            | 65.77 ± 0.72% | 79.84 ± 0.55% | -             | -              |

The best results are highlighted in bold. In and Tran. indicate inductive and transductive FSL setting, respectively. *"_" denotes the absent results in original paper. 'CD', 'CH', 'CA', and 'WE' denote class description, class hierarchy, class attribute, and word embeddings from glove [87], respectively.
This demonstrates the proposed prototype completion is more effective; 2) Our PCWOPK method is more effective than the optimization-based methods, with an improvement of 3%~5%. Different from them, we focus on learning representative prototypes instead of optimizers; (3) From the results of the pre-training based approaches, we have the following observations. First, our PCWOPK exceeds the MetaBaseline method by a large margin, around 2%~4% (1-shot) and 1%~2% (5-shot). This verifies our motivation that estimating more accurate prototypes is better than fine-tuning feature extractor during meta-learning. Besides, the improvement of performance on 1-shot tasks is more obvious than on 5-shot tasks. This is reasonable because the problem of inaccurate estimation of prototypes on 1-shot is more remarkable than 5-shot tasks. Second, our PCWOPK outperforms Neg-Cosine and CentAlign, by around 1%~5%. This is because our method focuses on estimating more representative prototypes, instead of pre-training strategy or generating more training samples. Finally, our PCWOPK achieves comparable performance over DC method on miniImageNet, while performs slightly worse than DC on tieredImageNet. The reason is that the DC method leverages a deeper backbone WRN-28-10 instead of ResNet12 and a complex power transformation.

As for the inductive FSL methods exploring external knowledge (i.e., semantic-based approaches), they also leverage the external knowledge, i.e., class description (CD), class hierarchy (CH), class attribute (CA), and word embeddings (WE) from GloVe [87]. However, our PCWPK utilizes the external knowledge (i.e., CH, CA and WE) to learn to complete prototypes, instead of to combine modality or to learn the feature extractor. The result validates the superiority of our manner to incorporate the external knowledge. Note that our PCWPK method achieves competitive performance with the MultiSem method on 5-shot tasks on miniImageNet. We would like to emphasize that this is because MultiSem leverages a more complex backbone, namely Dense-121 with 121 layers, instead of ResetNet12 in our model.

In transductive FSL setting without exploring external knowledge, SRestoreNet is very related with our PCWOPK, which also explores the query samples to restore prototypes. However, different from it, we leverage the query samples to estimate the prototype distribution and then to fuse prototypes. The result validates the superiority of our PCWOPK. Compared with the graph-based approaches, our method obtains competitive classification performance, especially in 1-shot tasks. This is because our method exploits unlabeled data to combine mean-based and completed prototypes, which is better than embedding or label propagation in graph-based approaches. Finally, comparing with the pre-training based approaches, our PCWOPK obtains superior performance, which further validates the superiority of learning representative prototypes.

As for the transductive FSL setting with external semantic knowledge, we have the following observations: 1) Our PCWPK exceeds the ECKPN method by around 9% on 1-shot tasks and achieves comparable performance on 5-shot tasks, which show superiority of our method; 2) Compared with the conference version [18], the extended PCWPK version (EM-based and Improved EM-based) exceeds it by 1%~6%. The main reason is that we explore unseen parts/attributes and enhance the GaussFusion by introducing an iterative parameter estimation algorithm; 3) our improved EM-based method performs the best in all extended methods, thus it is used as a default setting in subsequent discussion.

2) In Few-Shot Fine-Grained Classification: Table II summarizes the results on CUB-200-2011, which lead to similar observations as those in Table I. We observe that our method i) also achieves superior performance over state-of-the-art methods with an improvement of 1%~9% (inductive FSL) and 5%~16% (transductive FSL); ii) exceeds the conference version around 1%; iii) obtains almost consistent performance on 1-shot and 5-shot tasks, while the improvements on 1-shot task over baselines are more significant than on 5-shot. The results on few-shot fine-grained classification tasks further verify the effectiveness of the proposed method, especially for 1-shot classification tasks.

### D. Statistical Analysis

We conduct additional statistical experiments on our PCWPK method to answer the following four questions:

1) Is Our Idea Reasonable on Realistic Data? We randomly select five classes from the novel classes of miniImageNet and retrieve top-5 nearest and farthest samples from its ground-truth class center in the feature space. As shown in Fig. 8, the nearest images are more complete; however, the farthest samples are missing partial parts/attributes due to its incompleteness, noise background, or obscured details.

2) Does Our Method Obtain More Accurate Prototypes? We calculate the average cosine similarity between the estimated prototypes and the real prototypes on 1000 episodes.

### TABLE II

**EXPERIMENT RESULTS ON THE CUB-200-2011 DATA SET**

| Setting | Method                  | S-way 1-shot | 5-way 5-shot |
|---------|-------------------------|--------------|--------------|
|         | RESTORENET [15]         | 74.32 ± 0.91% | - ± - %      |
|         | RAP-ProtoNet [52]       | 75.17 ± 0.63% | 88.29 ± 0.34% |
|         | MAML [8]                | 55.92 ± 0.95% | 72.09 ± 0.76% |
|         | MultiSem [57]           | 76.1%         | 82.5%        |
|         | CPAE [5]                | 80.11 ± 0.34% | 89.28 ± 0.33% |
|         | CFA [72]                | 73.80 ± 0.80% | 86.80 ± 0.50% |
|         | Neg-Cosine [36]         | 72.66 ± 0.85% | 89.40 ± 0.43% |
|         | CentAlign [37]          | 72.11 ± 0.99% | 88.65 ± 0.55% |
|         | Maximum likelihood + DC [38] | 77.22 ± 0.14% | 89.58 ± 0.27% |
|         | SVM + DC [38]           | 79.41 ± 0.32% | 90.67 ± 0.53% |
|         | Logistic Regression + DC | 79.56 ± 0.87% | 90.67 ± 0.35% |
|         | PCWOPK (MeanFusion)     | 82.04 ± 0.90% | 90.58 ± 0.50% |
|         | PCWPK (MeanFusion)      | 88.59 ± 0.58% | 90.05 ± 0.34% |

The best results are highlighted in bold. In and Trans. indicate inductive and transductive setting, respectively.
TABLE III

THE COSINE SIMILARITY BETWEEN THE ESTIMATED AND REAL PROTOTYPES ON 1000 EPISODES (5-WAY 1-SHOT) OF MINIIMAGENET, TIEREDIMAGENET, AND CUB-200-2011

| Methods               | miniImageNet | tieredImageNet | CUB-200-2011 |
|-----------------------|--------------|----------------|--------------|
| SRestoreNet           | 0.72         | 0.86           | 0.68         |
| BD-CSNP               | 0.72         | 0.67           | 0.79         |
| Conference Version [18] | 0.72     | 0.84           | 0.68         |
| PCWPK                 | 0.72         | 0.85           | 0.80         |

For a fair comparison, we report the results of SRestoreNet, FSLKT, and BD-CSNP as the baselines. As shown in Table III, the results show that our PCWPK obtains more accurate prototypes than these baselines and the conference version [18]. Note that the prototypes ̂p_k from SRestoreNet is better than our PCWPK. This is reasonable because they leverage unlabeled samples before restoring prototypes. However, we exploit them after completing prototypes.

3) Is Our Method Effective for the Samples Far Away From Its Class Center? On the novel classes of miniImageNet, tieredImageNet, and CUB-200-2011, we calculate the cosine similarity between each noise image and its class center and sort them in descending order (i.e., the larger the sample number is, the farther away it is from the class center). Then, we take the noise images as inputs to predict the prototypes by using our PCWPK and RestoreNet, respectively. The cosine similarity between predicted prototypes and real class centers is shown in Fig. 9. Note that i) we smoothen the curve through moving average with 50 samples; ii) we show the average results for all novel classes. From the results of the above three datasets, we observe our PCWPK achieves more accurate prototypes than RestoreNet and the improvement becomes larger as the samples are farther away from its center. This means that our PCWPK can recover representative prototypes, especially when they are far away from their ground-truth centers.

4) How to set the Number of Iterations n_iter for GaussFusion With Improved EM-Based Method and is it Converge? To find the optimal n_iter, we conduct experiments on 5-way 1-shot and 5-shot tasks of miniImageNet, tieredImageNet, and CUB-200-2011, respectively, and report the test accuracy of the proposed method with different n_iter. The results are shown in Fig. 10. From these experimental results, we observe that 1) the iteration process of parameter estimation is very important; and 2) our improved EM-based method can converge to a stable result only by performing 6-step EM iterations and obtains the best performance on all datasets. This verifies the efficiency of our improved EM-based method on parameter estimation.

E. Ablation Study

We conduct an ablation study on miniImageNet, tieredImageNet, and CUB-200-2011, to answer four questions.

Is Effective Our ProtoComNet? To assess the effects of ProtoComNet. In Table IV, i) we remove all components of our PCWPK, i.e., classifying each sample by the mean-based prototypes; ii) we add the ProtoComNet proposed in the conference version [18] (i.e., removing unseen parts/attributes) on i) and classify each sample by the completed prototypes; iii) we extend [18] by introducing the PATNet on ii) to explore the unseen parts/attributes for ProtoComNet and classify each sample by the completed prototypes. From the results of i) and ii) in Table IV, we observe that 1) the latter exceeds the former in 1-shot tasks, by
Fig. 10. Performance of GaussFusion with different iterations on 5-way 1/5-shot tasks of miniImagenet, tieredImagenet and CUB-200-2011.

### TABLE IV
**ABLATION STUDY ON MINIIMAGENET, TIEREDIMAGENET AND CUB-200-2011**

| Method | miniImagenet | tieredImagenet | CUB-200-2011 |
|--------|--------------|----------------|--------------|
| PCPWPK (MeanFusion) | 69.68 ± 0.76% | 74.19 ± 0.90% | 88.95 ± 0.38% |
| w/o MSE Loss defined in Eq. 5 | 62.90 ± 0.79% | 70.42 ± 0.91% | 82.56 ± 0.78% |
| GaussFusion | 71.66 ± 0.92% | 72.35 ± 0.90% | 84.06 ± 1.00% |
| w/o MSE Loss defined in Eq. 5 | 70.42 ± 0.91% | 80.76 ± 0.75% | 88.60 ± 0.57% |

The experimental results are reported in Table V. From the results, we can see that the classification performance of our method decreases by around 1%~5% after removing the MSE loss. This means that employing the MSE loss is beneficial for our method. This is because 1) the real prototypes defined in (1) provide an explicit optimization target for training the prototype completion network; and 2) the phase of learning to complete prototypes can provide good initial parameters for prototype completion network, which alleviates the complexity of jointly learning the feature extractor and prototype completion network in meta-training phase.

### TABLE V
**ABBLATION STUDY OF THE MSE LOSS DEFINED IN (5) ON MINIIMAGENET, TIEREDIMAGENET, AND CUB-200-2011**

| Settings | 5-way 1-shot | 5-way 5-shot |
|----------|--------------|--------------|
| miniImagenet | 69.68 ± 0.76% | 81.65 ± 0.54% |
| tieredImagenet | 74.19 ± 0.90% | 86.09 ± 0.60% |
| CUB-200-2011 | 88.95 ± 0.38% | 94.05 ± 0.34% |

Is Effective Our GaussFusion? To verify the effectiveness of our GaussFusion. In Table IV, iv) we fuse the mean-based and completed prototypes by MeanFusion; v) we replace the MeanFusion of iv) by our two-step estimation method-based GaussFusion, i.e., the conference version [18]; vi) we replace the two-step estimation method by the improved EM-based estimation method on v), where we don’t use the EM-based estimation method because we have proved that the improved EM-based methods is more effective than EM-based methods in Tables I and II. According to the result in iv) and v) of Table IV, we find that 1) the problem of ProtoComNet with poor performance on 5-shot tasks is effectively solved after we use the MeanFusion strategy (i.e., the assumption-based distribution estimation method); 2) the performance of the ProtoComNet can be further improved when it is combined with the GaussFusion with the two-step distribution estimation method, which is our conference strategy, by around 3%. The result suggests that the two-step method is more effective than the assumption-based method. The key reason is the two-step method effectively estimates prototype distribution by exploiting the unlabelled samples. Besides, from the results of v) and vi), we observe that the latter achieve 1%~2% higher classification accuracy. This is because the improved EM-based estimation method estimates more accurate prototype distribution for GaussFusion in an iterative manner.

Can Our GaussFusion Alleviate the Prototype Completion Error Problem? To verify this point, we analyze the impacts performance with an improvement of 1%~2%. This implies that exploiting unseen parts/attributes is beneficial for estimating representative prototypes.

around 4%, which means that learning to complete prototypes is effective; 2) the latter obtains poor performance in 5-shot tasks. As our analysis in Section III-D, the phenomenon results from the bias of ProtoComNet, namely the primitive knowledge noises or base-novel class differences. Besides, comparing the results of ii) and iii), we find that the latter achieves superior learning to complete prototypes is effective; 2) the latter obtains poor performance in 5-shot tasks. As our analysis in Section III-D, the phenomenon results from the bias of ProtoComNet, namely the primitive knowledge noises or base-novel class differences. Besides, comparing the results of ii) and iii), we find that the latter achieves superior performance with an improvement of 1%~2%. This implies that exploiting unseen parts/attributes is beneficial for estimating representative prototypes.

Is Effective Our GaussFusion? To verify the effectiveness of our GaussFusion. In Table IV, iv) we fuse the mean-based and completed prototypes by MeanFusion; v) we replace the MeanFusion of iv) by our two-step estimation method-based GaussFusion, i.e., the conference version [18]; vi) we replace the two-step estimation method by the improved EM-based estimation method on v), where we don’t use the EM-based estimation method because we have proved that the improved EM-based methods is more effective than EM-based methods in Tables I and II. According to the result in iv) and v) of Table IV, we find that 1) the problem of ProtoComNet with poor performance on 5-shot tasks is effectively solved after we use the MeanFusion strategy (i.e., the assumption-based distribution estimation method); 2) the performance of the ProtoComNet can be further improved when it is combined with the GaussFusion with the two-step distribution estimation method, which is our conference strategy, by around 3%. The result suggests that the two-step method is more effective than the assumption-based method. The key reason is the two-step method effectively estimates prototype distribution by exploiting the unlabelled samples. Besides, from the results of v) and vi), we observe that the latter achieve 1%~2% higher classification accuracy. This is because the improved EM-based estimation method estimates more accurate prototype distribution for GaussFusion in an iterative manner.

Can Our GaussFusion Alleviate the Prototype Completion Error Problem? To verify this point, we analyze the impacts performance with an improvement of 1%~2%. This implies that exploiting unseen parts/attributes is beneficial for estimating representative prototypes.

around 4%, which means that learning to complete prototypes is effective; 2) the latter obtains poor performance in 5-shot tasks. As our analysis in Section III-D, the phenomenon results from the bias of ProtoComNet, namely the primitive knowledge noises or base-novel class differences. Besides, comparing the results of ii) and iii), we find that the latter achieves superior performance with an improvement of 1%~2%. This implies that exploiting unseen parts/attributes is beneficial for estimating representative prototypes.

Is Effective Our GaussFusion? To verify the effectiveness of our GaussFusion. In Table IV, iv) we fuse the mean-based and completed prototypes by MeanFusion; v) we replace the MeanFusion of iv) by our two-step estimation method-based GaussFusion, i.e., the conference version [18]; vi) we replace the two-step estimation method by the improved EM-based estimation method on v), where we don’t use the EM-based estimation method because we have proved that the improved EM-based methods is more effective than EM-based methods in Tables I and II. According to the result in iv) and v) of Table IV, we find that 1) the problem of ProtoComNet with poor performance on 5-shot tasks is effectively solved after we use the MeanFusion strategy (i.e., the assumption-based distribution estimation method); 2) the performance of the ProtoComNet can be further improved when it is combined with the GaussFusion with the two-step distribution estimation method, which is our conference strategy, by around 3%. The result suggests that the two-step method is more effective than the assumption-based method. The key reason is the two-step method effectively estimates prototype distribution by exploiting the unlabelled samples. Besides, from the results of v) and vi), we observe that the latter achieve 1%~2% higher classification accuracy. This is because the improved EM-based estimation method estimates more accurate prototype distribution for GaussFusion in an iterative manner.

around 4%, which means that learning to complete prototypes is effective; 2) the latter obtains poor performance in 5-shot tasks. As our analysis in Section III-D, the phenomenon results from the bias of ProtoComNet, namely the primitive knowledge noises or base-novel class differences. Besides, comparing the results of ii) and iii), we find that the latter achieves superior
of primitive knowledge with different noise levels $\epsilon$ on classification performance of our PCWPK. We report the results of miniImagenet, tieredImageNet, and CUB-200-2011 datasets in Table VI. Here, we introduce noises by randomly adding or removing class parts/attributes with probability $\epsilon$. It can be observed that our method is more robust to primitive knowledge noises when GaussFusion is applied.

### Is Effective the Random Selecting Strategy Defined in (12)?

To verify the effectiveness of such strategy, we conduct an ablation study on PCWPK by removing the random selecting strategy defined in (12). The experimental results are reported in Table VII. We find that the classification performance of our method decreases by around $1\% \sim 3\%$ after removing the random selecting strategy, which implies employing random selecting strategy is effective.

### F. Visualization

In this subsection, we conduct visualization analysis on feature space to answer the following two questions:

#### How are the Part/Attribute Distributed in the Feature Space?

To understand how our method complete prototypes by using extracted part/attribute features, we randomly select two part/attribute from miniImagenet, i.e., “paw” and “tail”. We visualize of all classes by t-SNE in the feature space, where the classes with the part/attribute “paw” or “tail” are marked in color “red”, otherwise in color “blue”. As shown in Fig. 11, we find these classes that have the same attributes are clustered together, which is beneficial to learn to complete prototypes.

![Visualization of part/attribute feature on miniImageNet.](image1)

#### How Does Our Method Work?

To understand how does the proposed method work, we select a 5-way 1-shot and 5-shot classification task from the meta-test set of miniImageNet to visualize the prototypes and samples by t-SNE. As shown in Fig. 12, after completing and fusing the class prototypes, the fused prototypes (marked in squares) become closer to real prototypes (marked in stars).

![Visualization of a 5-way 1/5-shot task sampled from the meta-test set of miniImageNet. Best viewed in color.](image2)

### VI. Conclusion

For few-shot learning, a simple pre-training on base classes can obtain a good feature extractor, where the novel class samples can be well clustered together. The key challenge is how to obtain more representative prototypes because the novel class samples spread as groups with large variances. To solve the issue, we introduce primitive knowledge and extract representative feature for seen attributes as priors. Then we propose a part/attribute transfer network to infer the visual features for unseen parts/attributes as supplementary priors, a prototype completion network to complete prototypes via primitive knowledge and these priors, and a Gaussian-based prototype fusion strategy to alleviate the prototype completion error problem. Particularly, in the fusion strategy, we develop three methods to estimate fusion parameters, i.e., two-step method, EM (Expectation Maximization)-based method, and improve EM-based estimation method. Beside, we also develop an economic prototype completion version for fair comparison with existing method without external knowledge, which does not need to collect primitive knowledge. Experiments show that our method obtains superior performance on five benchmarks.
received the bachelor's degree from the Harbin Institute of Technology, Shenzhen, China, in 2009 and 2013, respectively. He is currently an associate professor with Harbin Institute of Technology, in 2023. He is with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai, China, in 2018, pp. 719–729.

Shigeto I., Suzuki K., Har ma, M. Shimbo, and Y. Matsumoto, “Ridge regression, hubness, and zero-shot learning,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases, 2015, pp. 135–151.

M. Radovanovic, A. Nanopoulos, and M. Ivanovic, “Hubs in space: Popular nearest neighbors in high-dimensional data,”J. Mach. Learn. Res., vol. 11, no. sept, pp. 2487–2531, 2010.

Z. Ji, X. Yu, Y. Yu, Y. Pang, and Z. Zhang, “Semantic-guided classimbalance learning model for zero-shot image classification,”IEEE Trans. Cybern., vol. 52, no. 7, pp. 6543–6554, Jul. 2021.

S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, “Synthesized classifiers for zero-shot learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 5327–5336.

H. Jiang, R. Wang, S. Shan, and X. Chen, “Learning class prototypes via structure alignment for zero-shot recognition,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 118–134.

X. Zhao, Y. Shen, S. Wang, and H. Zhang, “Boosting generative zero-shot learning by synthesizing diverse features with attribute augmentation,” in Proc. AAAI Conf. Artif. Intell., 2022, pp. 3454–3462.

Z. Li, Q. Chen, and Q. Liu, “Augmented semantic feature based generative network for generalized zero-shot learning,”Neural Netw., vol. 143, pp. 1–11, 2021.

A. Braytey, M. Naji, A. Anaissi, K. Chaturvedi, and M. Prasad, “Zero-shot learning with missing attributes using semantic correlations,” in Proc. Int. Joint Conf. Neural Netw., 2021, pp. 1–7.

Y. Geng et al., “OntoZSL: Ontology-enhanced zero-shot learning,” in Proc. Web Conf., 2021, pp. 3325–3336.

J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in Proc. Conf. Empir. Methods Natural Lang. Process., 2014, pp. 1532–1543.

O. Vinyals et al., “Matching networks for one shot learning,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2016, pp. 3630–3638.

A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” J. Roy. Stat. Soc.: Ser. B (Methodological), vol. 39, no. 1, pp. 1–22, 1977.

M. Ren et al., “Meta-learning for semi-supervised few-shot classification,” in Proc. Int. Conf. Learn. Representations, 2018, pp. 1–15.

W. Xu, Y. Xu, H. Wang, and Z. Tu, “Attentional constellation nets for few-shot learning,” in Proc. Int. Conf. Learn. Representations, 2021, pp. 1–16.

S. Baik, M. Choi, J. Choi, H. Kim, and K. M. Lee, “Meta-learning with adaptive hyperparameters,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 1532–1543.

S. Xu et al., “Empirical bayes transductive meta-learning with synthetic gradients,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 1–16.

L. Bertinetto, J. F. Henriques, P. Torr, and A. Vedaldi, “Meta-learning with differentiable closed-form solvers,” in Proc. Int. Conf. Learn. Representations, 2018, pp. 1–15.

B. Oreshkin, P. Rodríguez López, and A. Lacoste, “TADAM: Task dependent adaptive metric for improved few-shot learning,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2018, pp. 719–729.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Yunming Ye received the PhD degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2004. He is currently a professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include data mining, text mining, and ensemble learning algorithms.

Shanshan Feng received the PhD degree in computer science from Nanyang Technological University, Singapore, in 2017. He is currently an associate professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include sequential data mining and social network analysis.