

# UNCERTAINTY-BASED NETWORK FOR FEW-SHOT IMAGE CLASSIFICATION

Minglei Yuan, Qian Xu, Chunhao Cai, Yin-Dong Zheng, Tao Wang, Tong Lu\* and Wenbin Li

National Key Laboratory for Novel Software Technology, Nanjing University  
 {mlyuan, chinyxu}@mail.nju.edu.cn, chunhao.cai@smartmore.com,  
 {ydzheng0331, taowangzj}@gmail.com, {lutong, liwenbin}@nju.edu.cn

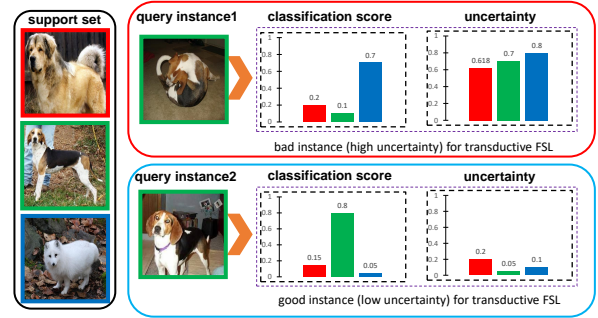
## ABSTRACT

The transductive inference is an effective technique in the few-shot learning task, where query sets update prototypes to improve themselves. However, these methods optimize the model by considering only the classification scores of the query instances as confidence while ignoring the uncertainty of these classification scores. In this paper, we propose a novel method called Uncertainty-Based Network, which models the uncertainty of classification results with the help of mutual information. Specifically, we first data augment and classify the query instance and calculate the mutual information of these classification scores. Then, mutual information is used as uncertainty to assign weights to classification scores, and the iterative update strategy based on classification scores and uncertainties assigns the optimal weights to query instances in prototype optimization. Extensive results on four benchmarks show that Uncertainty-Based Network achieves comparable performance in classification accuracy compared to state-of-the-art methods.

**Index Terms**— Few-shot learning, Uncertainty, Mutual information

## 1. INTRODUCTION

Few-shot image classification aims to learn knowledge with a few labeled samples quickly. Recent few-shot image classification methods can be divided into two types. The first type of method [1, 2, 3] is called inductive inference, where each unlabeled instance is predicted independently. Although these methods perform well, they ignore the information of unlabeled query instances. Thus, another type of method called transductive inference [4, 5] is proposed to explore the information of the unlabeled instances. Since transduction inference methods need to use pseudo-labels of unlabeled instances in the query set during the training phase, they often use few-shot learning models to obtain pseudo-labels and adopt classification scores as confidence.



**Fig. 1. Examples of classification scores and uncertainty.** Two query instances in 3-way 1-shot transductive FSL, the smaller the uncertainty, the higher the confidence of the classification result.

Although these transduction inference methods usually have better performance than inductive methods, due to the lack of sufficient data to enhance the generalization ability of the model, the models trained with few samples often obtain classification scores with high uncertainty. This limitation can affect the effectiveness of the transductive FSL methods, which only use classification scores to refine prototypes. Fig. 1 shows that a high classification score may have high uncertainty. If only classification results are considered, query instance 1 will be used to optimize class 3, misleading the model. Hence, the uncertainty information of the classification score is especially crucial for the performance of the transductive FSL but still under-explored.

In this paper, we propose a novel uncertainty-based method for few-shot learning. It can model the uncertainty of classification results for each query instance and perform uncertainty-aware transductive few-shot learning. Specifically, we first augment each query instance with multiple data augmentation methods and classify these augmented instances. Then, we compute the mutual information (MI) to measure the uncertainty of the classification scores and calculate the average classification scores for each query instance. Next, a weight generator with learnable parameters is used to generate the weights of the average classification scores based on the MI. These weights and the average classification scores are then used to calculate the weights of the query instances.

This work is supported by the Natural Science Foundation of China under Grant 61672273 and Grant 61832008, and Scientific Foundation of State Grid Corporation of China (Research on Ice-wind Disaster Feature Recognition and Prediction by Few-shot Machine Learning in Transmission Lines).

The weights of the query instances are used to update the prototype by the weighted average of the query instances and the corresponding support instances. The refined prototype is affected by the pseudo-classification scores of all query instances and their uncertainty.

Our contributions can be summarized as follows:

1. To the best of our knowledge, we are the first to measure the uncertainty of the classification result of each query instance by calculating its mutual information from various augmentation algorithms in the transduction few-shot image classification.
2. We propose a novel method called Uncertainty-Based Network (UCN), which refines prototypes by iteratively performing transduction inference based on the pseudo classification results and corresponding mutual information.
3. The proposed method has been validated on four datasets, and the classification accuracy outperforms other methods by 1%-18%. It also achieves state-of-the-art performance in the semi-supervised setting.

## 2. RELATED WORK

### 2.1. Transductive few-shot learning

Compared with inductive few-shot learning methods, transductive few-shot learning methods aim to explore how to utilize the query set to help improve the performance of image classification. The first type is GNN-based methods that enhance classification performance by exploiting Graph Neural Networks (GNN) in the transductive few-shot learning task. Typical works include TPN [4], EGNN [6] and DPGN [7]. Some other methods that use query instances to refine prototypes seem more straightforward. For example, CAN [8] adopts an iterative process during transductive inference. Specifically, in each iteration, the top-k confident query instances are chosen and used to update the prototypes. The uncertainty confidence for each query instance equals the cosine similarity between itself and the prototype. MCT [9] also adopts the iterative process, but it attempts to adapt the confidence to a specific task by adding an extra learnable temperature to the Euclidean distance metric.

### 2.2. Uncertainty-based learning method

Facts show that uncertainty-based methods have made good progress in the practice of various downstream tasks. For example, Yarin et al. [10] were the first to explore the nature of uncertainty in regression and classification tasks extensively, expounded that uncertainty is indispensable, and developed the necessary tools. In semi-supervised learning, UPS [11] proposes an uncertainty-aware pseudo-label selection framework to increase the accuracy of pseudo-labeling by reducing the noise during training. UAFS [12] proposes a framework based on few-shot image classification, which converts

the observed similarity of query-support pairs into probabilistic representations and uses graph convolutional networks (GCN) to take advantage of this uncertainty to achieve more effective optimization results. TIM [13] introduces the concept of mutual information to measure the uncertainty of model prediction results and optimize the model based on the uncertainty of similarity.

## 3. APPROACH

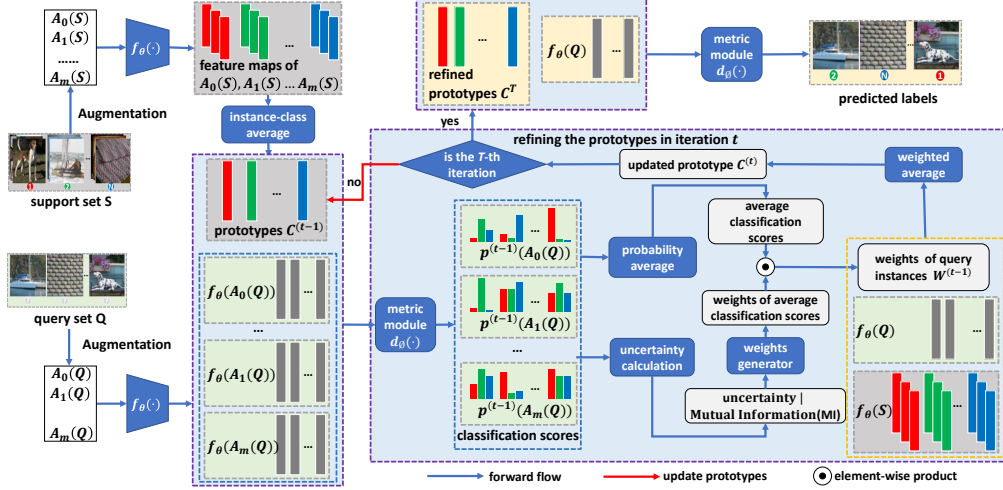
For different input instances, the output of the neural network has different uncertainties, while the observation noise also has an impact on the uncertainty [14]. Thus, we propose an Uncertainty-Based Network (UCN), which models the uncertainty of model classification results and optimizes the prototype by considering the uncertainty of model classification results and classification results.

### 3.1. Problem Definition

To present UCN well, we first introduce some notations about few-shot learning in this section. The training set, validation set and test set are denoted as  $\mathcal{D}_{\text{train}}$ ,  $\mathcal{D}_{\text{val}}$  and  $\mathcal{D}_{\text{test}}$  respectively. The Few-shot learning method aims to learn a model from the support set  $\mathcal{S}$  and use the model to identify sample categories in the query set  $\mathcal{Q}$ . Most few-shot learning methods adopt a special paradigm named  $N$ -way  $K$ -shot classification to evaluate their performance. To build a  $N$ -way  $K$ -shot classification task, we first need to randomly sample  $N$  classes from the training set. After that, we not only need to sample  $K$  labeled instances to make up the support set  $\mathcal{S}$  (i.e.,  $\mathcal{S} = \{(\mathbf{x}_{i,j}^S, y_{i,j}^S) \mid i = 1 : N, j = 1 : K\}$ ) for each chosen class, but also need to sample some instances without labels from these  $N$  classes to form the query set  $\mathcal{Q}$  (i.e.,  $\mathcal{Q} = \{\mathbf{x}_i^Q \mid i = 1 : q\}$ , where  $q$  is the size of the query set). In addition, there is no crossover between the samples of the support set  $\mathcal{S}$  and query set  $\mathcal{Q}$ , but they share the label space.

### 3.2. Overall Pipeline

This section explains the overview of UCN. The framework of UCN is in Fig. 2. First,  $m + 1$  different data augmentation algorithms are applied to each support and query instance as perturbations, leading to  $m + 1$  different variants for each instance. For convenience, we use  $A_k(\cdot)$  ( $0 \leq k \leq m$ ) to denote the data augmentation algorithms, where  $A_0(\cdot)$  presents the identity function. After that, each augmented instance is fed into a feature extractor  $f_\theta(\cdot)$  to obtain their feature maps. Next, the initial prototypes  $\mathcal{C}^{(0)} = \{\mathbf{c}_i^{(0)}\}_{i=1}^N$  are computed by averaging the feature map of these augmented support instances,  $i$  indicates the  $i$ -th class in support set  $\mathcal{S}$ . After obtaining the initial prototypes, we perform  $T$  consecutive iterations to refine the prototypes. The  $t$ -th iteration updates the prototypes based on the feature maps of the augmented instances and the prototypes generated in the previous iteration. The details of the  $t$ -th iteration are shown in the section 3.3.



**Fig. 2. The framework of the proposed UCN.** We optimize the prototype using uncertain information iteratively. First, we model the mutual information (MI) and average classification score for each query instance based on the classification score obtained by applying various augmentations  $A(\cdot)$  to support set  $S$  and query set  $Q$ . Then there is a process of  $T$  iterations to optimize the prototype. For the  $t$ -th iteration, the MI is fed into a weight generator to calculate the weights of average classification scores, combined with the average classification scores to update the weights of the query instances. The refined prototype for a specific class can be obtained by weighted averaging all corresponding support and query instances.

### 3.3. Refining the Prototypes in Iteration $t$

As the  $T$  iterations are the core part of UCN, we introduce the details of the  $t$ -th iteration in this section.

**Classify the Augmented Instances.** To capture the uncertainty of the model prediction results, we employ different data augmentation methods to perturb the prediction results of the model and then calculate the uncertainty of the classification results.

At the beginning of the  $t$ -th iteration,  $m$  data augmentation methods are applied on each instance in the query set  $Q$ . All the augmented query instances are then classified using a metric-based approach, which is shown as:

$$P^{(t)}(y = i | \mathbf{x}_j^Q, A_k) = \frac{\exp(-d_\phi(f_\theta(A_k(\mathbf{x}_j^Q)), \mathbf{c}_i^{(t-1)}))}{\sum_{i=1}^N \exp(-d_\phi(f_\theta(A_k(\mathbf{x}_j^Q)), \mathbf{c}_i^{(t-1)}))}, \quad (1)$$

where  $\mathbf{x}_j^Q$  denotes  $j$ -th instance of the query set  $Q$ ,  $\mathbf{c}_i^{(t-1)}$  is the  $i$ -th prototype in the  $(t-1)$ -th iteration, and  $d_\phi(\cdot, \cdot)$  indicates the Euclidean distance metric with a learnable parameter  $\phi$ , which is formulated as:

$$d_\phi(\mathbf{x}_1, \mathbf{x}_2) = \left\| \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|_2 g_\phi(\mathbf{x}_1)} - \frac{\mathbf{x}_2}{\|\mathbf{x}_2\|_2 g_\phi(\mathbf{x}_2)} \right\|_2, \quad (2)$$

where  $\mathbf{x}_1, \mathbf{x}_2$  represent two different sample,  $\|\cdot\|_2$  is the Euclidean norm, and  $g_\phi(\cdot)$  is a temperature generator with learnable parameters. In this paper, we use instance-specific temperatures because, as demonstrated in [15, 16], the scaling temperature in a softmax operation can significantly affect the performance of a few learning models. Therefore, an instance-specific temperature adapts the metric to a specific task.

**Calculate the Average and MI of the Classification Scores.** The average classification scores  $\bar{P}^{(t)}$  of  $m+1$  augmented instances for each query sample is first calculated, as:

$$\bar{P}^{(t)}(y = i | \mathbf{x}_j^Q) = \frac{1}{m+1} \sum_{k=0}^m P^{(t)}(y = i | \mathbf{x}_j^Q, A_k), \quad (3)$$

where  $i$  is the possible classes of  $\mathbf{x}_j^Q$ ,  $j$  indicates the  $j$ -th element in the query set.

We draw on the tractable approach proposed by [17] to compute mutual information, which approximates the inferred mutual information by adjusting the parameters of the model to obtain multiple predictions. In this paper, we propose to calculate the uncertainty of the model prediction results with the help of several data augmentation methods. Specifically,  $\mathbb{I}^{(t)}[y | \mathbf{x}_j^Q]$  is calculated based on the classification scores of  $m+1$  augmented instances via Shannon entropy  $\mathbb{H}(\cdot)$ , which is shown as:

$$\mathbb{I}^{(t)} = \mathbb{H}(\bar{P}^{(t)}) - \frac{1}{m+1} \sum_{k=0}^m \mathbb{H}(P^{(t)}). \quad (4)$$

**Infer Weights for Query Instances.** This paper proposes an uncertainty-aware few-shot learning method, which considers uncertainty information and instance classification scores to select appropriate query instances. All the uncertainty information of query instances is first fed to a weight generator  $h_\psi(\cdot)$ , which is a multi-head attention module in this paper. Then, the outputs are applied on the average classification scores to get the weight matrix  $\mathbf{W}^{(t)}$  of query instances, as Eq. 5, to update the prototypes. The weight matrix  $\mathbf{W}^{(t)}$  integrates mutual information and probability distributions through multi-head attention mechanism and provides a hint on the uncertainty of different query instances since the learned weights  $\mathbf{W}^{(t)}$  reflect the impact of uncertainty on the query instances.

$$\mathbf{W}^{(t)} = [h_\psi(\mathbb{I}^{(t)}) \odot \bar{P}^{(t)}] = [w_{i,j}^{(t)}], \quad (5)$$

where  $\odot$  represents element-wise product. The weight generator  $h_\psi(\cdot)$  is constructed by one multi-head attention mod-

ule, as the multi-head attention can discover the relationship among elements of the input sequence.

**Optimize Prototypes.**  $\mathbf{W}^{(t)}$  is used to update the prototypes, which are linear combinations of the embedding of the support instance and the query instance with a fixed weight of 1 for the support instances and a weight  $\mathbf{W}^{(t)}$  for the query instances. The updated prototypes are in Eq. 6.

$$\mathbf{c}_i^{(t)} = \frac{\sum_{j=1}^K 1 \cdot \mathbf{x}_{i,j}^S + \sum_{j=1}^q w_{i,j}^{(t)} \cdot \mathbf{x}_j^Q}{K + \sum_{j=1}^q w_{i,j}^{(t)}}. \quad (6)$$

The prototype  $\mathbf{c}_i^{(t)}$  is updated with a weighted average using the above weights  $w_{i,j}^{(t)}$  combined with query instances and support instances. When selecting a query sample for prototype optimization, we consider the uncertainty of the classification result and the classification result for this query sample.

### 3.4. Loss Function

In this work, a joint loss function  $L$  is proposed to train UCN. It combines a classification loss  $L_{\text{cls}}$  and a regularization loss  $L_{\text{gen}}$  as Eq. 7.

$$L = L_{\text{cls}} + \lambda L_{\text{gen}}, \quad (7)$$

where  $\lambda$  is the hyperparameter to balance two losses. We empirically set it to 0.5 in the experiment.

The classification loss  $L_{\text{cls}}$  is the cross-entropy loss of the classification scores of all non-augmented query instances, and the goal of  $L_{\text{cls}}$  is to increase the classification accuracy of the model. The regularization loss  $L_{\text{gen}}$  aims to regularize the weight generator. The outputs of the support instances are as close as possible to their one-hot labels since the labels of support instances are already known. The regularization loss  $L_{\text{gen}}$  is formulated as Eq. 8.

$$L_{\text{gen}} = -\frac{1}{NK} \sum_{i=1}^N \sum_{j=1}^K \text{BCE} \left( w_{i,j}^{(T)}, \text{onehot}(i, N) \right), \quad (8)$$

where  $\text{BCE}(\cdot, \cdot)$  indicates the binary cross entropy loss,  $w_{i,j}^{(t)}$  denotes the weight of the  $j$ -th supported instance in the  $i$ -th class inferred from the weight generator  $h_{\psi}(\cdot)$ .  $N$  is the class number in support set,  $K$  is the instance number in each class, and  $\text{onehot}(i, N)$  indicates the one-hot vector whose length equals to  $N$  with its  $i$ -th position (starts from 1) being 1.

## 4. EXPERIMENTS

### 4.1. Datasets, Backbone and Implementation Details

To validate UCN, we conduct extensive experiments on four public datasets: *miniImageNet* [2], *tieredImageNet* [18], FC100 [15], and CIFAR-FS [19]. We choose different backbones (e.g., Conv-64/256, Resnet12/18 and WRN-28-10) for evaluation. More implementation details and experimental results are shown in the supplemental material.

### 4.2. Comparison with State-of-the-art Methods

**Quantitative Comparison.** Table 1, 2 show that UCN surpasses most of the methods on all datasets and demonstrates the effectiveness of UCN. It shows that the transductive FSL learning methods work better than the inductive FSL learning

**Table 1.** Classification accuracy on *miniImageNet* and *tieredImageNet* reported with 95% confidence intervals. "FE" column describes the backbone, "Tr" indicates whether methods are transductive, and the partially transductive methods are marked as "BN" because of using the mean and variance of query instance batches in BatchNorm layers [6, 7]. The superscript  $\dagger$  indicates dense classification [20] is adopted over feature maps. C64/256 is for Conv-64/256, R12/18 for ResNet-12/18 and W28 for WRN-28-10.

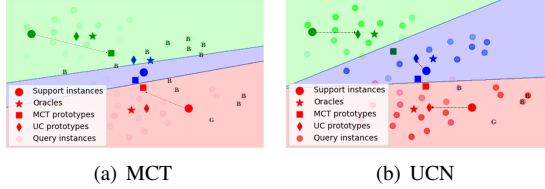
Method	FE	Tr	<i>miniImageNet</i> (%)		<i>tieredImageNet</i> (%)	
			1-shot	5-shot	1-shot	5-shot
ProtoNet [1]	C64	No	49.42 $\pm$ 0.78	68.20 $\pm$ 0.66	53.31 $\pm$ 0.89	72.69 $\pm$ 0.74
MatchNet [2]	C64	No	43.56 $\pm$ 0.84	55.31 $\pm$ 0.73	-	-
MAML [21]	C64	BN	48.70 $\pm$ 1.84	63.15 $\pm$ 0.91	-	-
RN [3]	C64	BN	50.44 $\pm$ 0.82	65.32 $\pm$ 0.70	54.48 $\pm$ 0.93	71.32 $\pm$ 0.78
TPN [4]	C64	Yes	53.75	69.43	57.53	72.85
Ent [5]	C64	Yes	50.46 $\pm$ 0.62	66.68 $\pm$ 0.52	58.05 $\pm$ 0.68	74.24 $\pm$ 0.56
UCN	C64	Yes	<b>57.97<math>\pm</math>0.66</b>	<b>73.01<math>\pm</math>0.75</b>	<b>58.95<math>\pm</math>1.13</b>	<b>75.82<math>\pm</math>0.79</b>
EGNN [6]	C256	Yes	-	76.37	-	70.15
UCN	C256	Yes	<b>59.19<math>\pm</math>1.14</b>	<b>77.70<math>\pm</math>0.64</b>	<b>68.90<math>\pm</math>1.17</b>	<b>82.78<math>\pm</math>0.67</b>
TADAM [15]	R12	No	58.5 $\pm$ 0.3	76.7 $\pm$ 0.3	-	-
CAN [8]	R12	No	63.85 $\pm$ 0.48	79.44 $\pm$ 0.34	69.89 $\pm$ 0.51	84.23 $\pm$ 0.37
DeepEMD [22]	R12	No	65.91 $\pm$ 0.82	82.41 $\pm$ 0.56	71.16 $\pm$ 0.87	86.03 $\pm$ 0.58
FEAT [23]	R12	No	66.78 $\pm$ 0.20	82.05 $\pm$ 0.14	70.80 $\pm$ 0.23	84.79 $\pm$ 0.16
CAN [8]	R12	Yes	67.19 $\pm$ 0.55	80.64 $\pm$ 0.35	73.21 $\pm$ 0.58	84.93 $\pm$ 0.38
DPGN [7]	R12	Yes	67.77 $\pm$ 0.32	84.60 $\pm$ 0.43	72.45 $\pm$ 0.51	87.24 $\pm$ 0.39
EPNet [24]	R12	Yes	66.50 $\pm$ 0.89	81.06 $\pm$ 0.60	76.53 $\pm$ 0.87	87.32 $\pm$ 0.64
Ent [5]	R12	Yes	62.35 $\pm$ 0.66	74.53 $\pm$ 0.54	68.41 $\pm$ 0.73	83.41 $\pm$ 0.52
IEPT[25]	R12	Yes	67.05 $\pm$ 0.44	82.90 $\pm$ 0.30	72.24 $\pm$ 0.50	86.73 $\pm$ 0.34
UAFS [12]	R12	Yes	64.22 $\pm$ 0.67	79.99 $\pm$ 0.49	69.13 $\pm$ 0.84	84.33 $\pm$ 0.59
UCN	R12	Yes	<b>76.78<math>\pm</math>1.01</b>	<b>85.35<math>\pm</math>0.61</b>	<b>80.00<math>\pm</math>1.12</b>	<b>87.50<math>\pm</math>0.64</b>
MCT $^\dagger$ [9]	R12	Yes	78.55 $\pm$ 0.86	86.03 $\pm$ 0.42	<b>82.32<math>\pm</math>0.81</b>	87.36 $\pm$ 0.50
UCN $^\dagger$	R12	Yes	<b>86.40<math>\pm</math>0.46</b>	<b>87.22<math>\pm</math>0.43</b>	82.02 $\pm$ 1.01	<b>90.83<math>\pm</math>0.39</b>
CTM [26]	R18	No	64.12 $\pm$ 0.82	80.51 $\pm$ 0.13	68.41 $\pm$ 0.39	84.24 $\pm$ 1.73
UCN	R18	Yes	<b>73.98<math>\pm</math>0.94</b>	<b>83.94<math>\pm</math>0.59</b>	<b>75.79<math>\pm</math>1.03</b>	<b>87.02<math>\pm</math>0.43</b>
CC+rot [27]	W28	Yes	62.93 $\pm$ 0.45	79.87 $\pm$ 0.33	70.53 $\pm$ 0.51	84.98 $\pm$ 0.36
EPNet [24]	W28	Yes	70.74 $\pm$ 0.85	84.34 $\pm$ 0.53	78.50 $\pm$ 0.91	88.36 $\pm$ 0.57
SIB [28]	W28	Yes	70.0 $\pm$ 0.6	79.2 $\pm$ 0.4	-	-
E <sup>3</sup> BM [29]	W28	Yes	71.4 $\pm$ 0.5	81.2 $\pm$ 0.4	75.6 $\pm$ 0.6	84.3 $\pm$ 0.4
UCN	W28	Yes	<b>78.55<math>\pm</math>0.97</b>	<b>85.99<math>\pm</math>0.57</b>	<b>79.66<math>\pm</math>1.02</b>	<b>88.87<math>\pm</math>0.63</b>

methods, indicating that query samples benefit prototype optimization. Note that the R12 backbone and pre-train strategy of the proposed UCN are most similar to FEAT [23]. Compare to FEAT, UCN achieves significant improvement on *miniImageNet* and *tieredImageNet* in 1-shot accuracy. Unlike UAFS [12] which model uncertainty of the similarities of query support pairs, we propose a new way of computing uncertainty information of query instances by computing the mutual information of the augmented query instances. Moreover, the obtained uncertainty information and the classification results of the query instances are used to guide the transductive inference process. Relative to UAFS, our prediction accuracy increased by a large margin on all metrics. In addition, we obtain competitive results in a semi-supervised setting and present the results in supplementary materials.

**Qualitative Comparison.** To analyze the effectiveness of UCN, we compare it with the previous state-of-the-art method MCT [9] based on the prototype refining strategy. Figure 3 visualized ProtoNet-style [1] prototype (support instances), instance feature maps (query instances), refined prototype generated by our method (UC prototype), MCT prototype, and the oracle prototype. The arrows mark the changes made

**Table 2.** Classification accuracy on FC100 and CIFAR-FS are reported with 95% confidence intervals.

Method	FE	Tr	FC100(%)		CIFAR-FS(%)	
			1-shot	5-shot	1-shot	5-shot
TADAM [15]	R12	No	40.1 $\pm$ 0.4	56.1 $\pm$ 0.4	-	-
DeepEMD [22]	R12	No	46.47 $\pm$ 0.48	<b>63.22 <math>\pm</math> 0.71</b>	-	-
DPGN [7]	R12	Yes	-	-	77.9 $\pm$ 0.5	90.2 $\pm$ 0.4
UAFS [12]	R12	Yes	41.99 $\pm$ 0.58	57.43 $\pm$ 0.38	74.08 $\pm$ 0.72	85.92 $\pm$ 0.42
UCN	R12	Yes	<b>48.00 <math>\pm</math> 1.04</b>	60.39 $\pm$ 0.82	<b>88.90 <math>\pm</math> 0.95</b>	<b>92.18 <math>\pm</math> 0.63</b>
SIB [28]	W28	Yes	-	-	80.0 $\pm$ 0.6	85.3 $\pm$ 0.4
UCN	W28	Yes	<b>47.61 <math>\pm</math> 0.99</b>	59.74 $\pm$ 0.78	<b>85.64 <math>\pm</math> 0.93</b>	<b>88.90 <math>\pm</math> 0.58</b>



**Fig. 3.** t-SNE figures for the results produced by MCT [9] and our UCN on 3-way 1-shot tasks. Classes are represented by red, green, and blue, respectively. The closer to Oracles, the better the prototype. The transparency of query instances means the maximum weight it gets.

to the prototype by the MCT and UCN methods. The oracle prototype is obtained by averaging the features of all samples of the same class in the dataset, which can represent the commonality of instances in a category because the averaging operation neutralizes the random noise over a wide range of instances. The RGB value of the points for query instances  $\mathbf{x}_j^Q$  is determined by the proportion among the weights  $w_{0,j}^{(T)}$ ,  $w_{1,j}^{(T)}$  and  $w_{2,j}^{(T)}$ , and its transparency reflects the maximum weight of the corresponding query instance, that is  $\max(w_{0,j}^{(T)}, w_{1,j}^{(T)}, w_{2,j}^{(T)})$ . In this way, we can intuitively understand the contribution of each query instance to the refined prototypes. In addition, Figure 3(a), 3(b) also draw the approximate decision boundaries based UC prototypes and MCT prototypes, respectively. All misclassified query instances are also marked with ground truth labels.

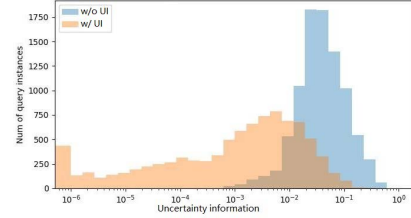
From Fig. 3, we can see that compared with MCT, the prototype generated by UCN is closer to the ideal state (*i.e.*, the oracle prototype). The query instances of the same category have higher weights, while the misclassified instances often have lower weights. This result reveals that UCN can make the decision boundary more reasonable. These phenomena are consistent with our hypothesis that using the uncertainty information of classification score can improve the prototype in transductive few-shot learning.

### 4.3. Ablation Study

We perform ablation experiments on several components of UCN to study their effect, including the impact of uncertainty information, the effect of  $T$ , the choice of augmentations, and the structure of the weight generator  $h_\psi(\cdot)$ . Due to space limitations, some of these ablation experiments are in the supple-

**Table 3.** Ablation study of uncertainty information (UI) and regularization loss ( $L_{gen}$ ) on *miniImageNet*, measured in %.

Method	UI	$L_{gen}$	Tr	1-shot	5-shot
baseline	✗	✗	✗	65.13 $\pm$ 0.80	82.29 $\pm$ 0.53
UCN_A	✗	✗	✓	69.65 $\pm$ 1.10	84.21 $\pm$ 0.55
UCN_B	✓	✗	✓	73.92 $\pm$ 1.06	84.20 $\pm$ 0.57
UCN	✓	✓	✓	<b>76.78 <math>\pm</math> 1.01</b>	<b>85.35 <math>\pm</math> 0.61</b>



**Fig. 4.** Distributions of uncertainty information of classification scores.

mental material.

In Table 3, we conduct an ablation study of the uncertainty information (UI) and the proposed regularization loss function ( $L_{gen}$ ) in *miniImageNet*. The baseline represents the model of ProtoNet-style [1], and it also takes the average of the corresponding augmented instances as the basis. UCN\_A and UCN\_B have the same structure and parameters as UCN, except for some differences in UI and loss functions, where UCN\_A directly uses the classification score of query instances as UI. Table 3 illustrates the effectiveness of the proposed UI. It shows that with the help of UI, the accuracy of the experiment increases by 4.27% for the 5-way 1-shot setting. We argue that the uncertainty of the model prediction results is high in the 5-way 1-shot setting. More beneficial instances in the query set can be selected to optimize the prototypes after taking the uncertainty information into account. Table 3 also demonstrates the effectiveness of  $L_{gen}$ , especially on 5-way 1-shot setting.

In Fig. 4, we conduct 3,000 5-way 1-shot episodes using the UCN\_A (made with w/o UI) and UCN\_B (marked with w/UI) and collect the top three query instances in each episode (ordered by their weights in class 0). The distribution shows that samples with a lower uncertainty in classification scores usually receive relatively higher weights when uncertainty information is introduced into the prototype refinement.

## 5. CONCLUSION

In this paper, we propose an Uncertainty-based network for few-shot learning, which utilizes the uncertainty information of query instances to refine the prototypes through transductive inference. Notably, we compute the weights of all the query instances based on both the classification scores and the uncertainty information of these scores. Extensive experiments on various benchmarks show that UCN outperforms the state-of-the-arts significantly on both transductive few-shot learning and semi-supervised few-shot learning tasks.

## 6. REFERENCES

- [1] Jake Snell, Kevin Swersky, and Richard S. Zemel, “Prototypical networks for few-shot learning,” in *NIPS*, 2017, pp. 4077–4087.
- [2] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra, “Matching networks for one shot learning,” in *NIPS*, 2016, pp. 3630–3638.
- [3] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in *CVPR*, 2018, pp. 1199–1208.
- [4] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang, “Learning to propagate labels: Transductive propagation network for few-shot learning,” in *ICLR*, 2019.
- [5] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto, “A baseline for few-shot image classification,” in *ICLR*, 2020.
- [6] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo, “Edge-labeling graph neural network for few-shot learning,” in *CVPR*, 2019, pp. 11–20.
- [7] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu, “DPGN: distribution propagation graph network for few-shot learning,” in *CVPR*, 2020, pp. 13387–13396.
- [8] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen, “Cross attention network for few-shot classification,” in *NeurIPS*, 2019, pp. 4005–4016.
- [9] Seong Min Kye, Haebeom Lee, Hoirin Kim, and Sung Ju Hwang, “Transductive few-shot learning with meta-learned confidence,” *CoRR*, vol. abs/2002.12017, 2020.
- [10] Gal Yarin, “Uncertainty in deep learning,” *University of Cambridge*, vol. 1, no. 3, 2016.
- [11] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh Singh Rawat, and Mubarak Shah, “In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning,” in *ICLR*. 2021, OpenReview.net.
- [12] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, Zhibo Chen, and Shih-Fu Chang, “Uncertainty-aware few-shot image classification,” in *IJCAI*, 2021, pp. 3420–3426.
- [13] Malik Boudiaf, Imtiaz Ziko, Jérôme Rony, Jose Dolz, Pablo Piantanida, and Ismail Ben Ayed, “Information maximization for few-shot learning,” in *NeurIPS*, 2020.
- [14] Bharath Hariharan and Ross Girshick, “Low-shot visual recognition by shrinking and hallucinating features,” in *ICCV*, Oct 2017.
- [15] Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste, “TADAM: task dependent adaptive metric for improved few-shot learning,” in *NeurIPS*, 2018, pp. 719–729.
- [16] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean, “Distilling the knowledge in a neural network,” *CoRR*, vol. abs/1503.02531, 2015.
- [17] Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel, “Bayesian active learning for classification and preference learning,” *CoRR*, vol. abs/1112.5745, 2011.
- [18] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel, “Meta-learning for semi-supervised few-shot classification,” in *ICLR*, 2018.
- [19] Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi, “Meta-learning with differentiable closed-form solvers,” in *ICLR*, 2019.
- [20] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc, “Dense classification and implanting for few-shot learning,” in *CVPR*, 2019, pp. 9258–9267.
- [21] Chelsea Finn, Pieter Abbeel, and Sergey Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *ICML*, 2017, vol. 70, pp. 1126–1135.
- [22] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen, “Deepemd: Few-shot image classification with differentiable earth mover’s distance and structured classifiers,” in *CVPR*, 2020, pp. 12200–12210.
- [23] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha, “Few-shot learning via embedding adaptation with set-to-set functions,” in *CVPR*, 2020, pp. 8805–8814.
- [24] Pau Rodríguez, Issam H. Laradji, Alexandre Drouin, and Alexandre Lacoste, “Embedding propagation: Smoother manifold for few-shot classification,” in *ECCV*, 2020, pp. 121–138.
- [25] Manli Zhang, Jianhong Zhang, Zhiwu Lu, Tao Xiang, Mingyu Ding, and Songfang Huang, “IEPT: instance-level and episode-level pretext tasks for few-shot learning,” in *ICLR*, 2021.
- [26] Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang, “Finding task-relevant features for few-shot learning by category traversal,” in *CVPR*, 2019, pp. 1–10.
- [27] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord, “Boosting few-shot visual learning with self-supervision,” in *ICCV*, 2019, pp. 8058–8067.
- [28] Shell Xu Hu, Pablo Garcia Moreno, Yang Xiao, Xi Shen, Guillaume Obozinski, Neil D. Lawrence, and Andreas C. Damianou, “Empirical bayes transductive meta-learning with synthetic gradients,” in *ICLR*, 2020.
- [29] Yaoyao Liu, Bernt Schiele, and Qianru Sun, “An ensemble of epoch-wise empirical bayes for few-shot learning,” in *ECCV*, 2020, vol. 12361, pp. 404–421.