Uncertainty-Aware Few-Shot Image Classification

Zhizheng Zhang[1*], **Cuiling Lan**[2†], **Wenjun Zeng**[2], **Zhibo Chen**[1†], **Shih-Fu Chang**[3]
[1]University of Science and Technology of China
[2]Microsoft Research Asia
[3]Columbia University
zhizheng@mail.ustc.edu.cn, {culan, wezeng}@microsoft.com
chenzhibo@ustc.edu.cn, sc250@columbia.edu

Abstract

Few-shot image classification learns to recognize new categories from limited labelled data. Metric learning based approaches have been widely investigated, where a query sample is classified by finding the nearest prototype from the support set based on their feature similarities. A neural network has different uncertainties on its calculated similarities of different pairs. Understanding and modeling the uncertainty on the similarity could promote the exploitation of limited samples in few-shot optimization. In this work, we propose Uncertainty-Aware Few-Shot framework for image classification by modeling uncertainty of the similarities of query-support pairs and performing uncertainty-aware optimization. Particularly, we exploit such uncertainty by converting observed similarities to probabilistic representations and incorporate them to the loss for more effective optimization. In order to jointly consider the similarities between a query and the prototypes in a support set, a graph-based model is utilized to estimate the uncertainty of the pairs. Extensive experiments show our proposed method brings significant improvements on top of a strong baseline and achieves the state-of-the-art performance.

1 Introduction

The strong capability of deep learning in part relies on the using of a large amount of labeled data for training, while humans are more readily able to learn knowledge quickly given a few examples. Few-shot learning [Vinyals *et al.*, 2016; Finn *et al.*, 2017; Karlinsky *et al.*, 2019; Dong and Xing, 2018; Mishra *et al.*, 2018] strives to a further step to develop deep learning approaches which generalize well to unseen tasks/classes with just few labeled samples. Among them, few-shot image classification [Vinyals *et al.*, 2016; Snell *et al.*, 2017; Sung *et al.*, 2018; Oreshkin *et al.*, 2018; Finn *et al.*, 2017; Nichol *et al.*, 2018; Lee *et al.*, 2019] has attracted much attention, which aims to classify data (query samples) of new categories given only a few labeled samples of these categories as examples (support samples).

For few-shot image classification, several latest research works reveal that good feature embedding is important to deliver favorable performance for the similarity-based classification [Chen *et al.*, 2019; Chen *et al.*, 2020; Huang and Tao, 2019; Tian *et al.*, 2020]. They average the feature embeddings of the support samples of a category to be the prototype of this category. Thus, with each prototype being category-specific, the set of prototypes plays the role of similarity-based classifier to identify the category of a query sample by finding the nearest prototype over the support set in the embedding space. During training, for a query sample, the logit vector (classification probability) is obtained by feeding the similarities between the query sample and the prototypes to a SoftMax function. Hence, the reliability of the estimated similarity is vital to the classification performance.

The quality of network output has been investigated and modeled with uncertainty in regression, classification, segmentation, multi-task learning [Kendall *et al.*, 2015; Kendall and Gal, 2017; Chang *et al.*, 2020; Kendall *et al.*, 2018] to benefit the optimization. Aleatoric uncertainty identifies the confidence level of the network on its output for a sample, which captures the noise inherent in the observations. For few-shot image classification, the network actually has different uncertainties on the calculated similarities of different query-prototype pairs. An observed similarity of a query-prototype pair, being a one time sampling, suffers from observation noise, where the higher of the uncertainty, the less reliable of the estimated similarity. For each few-shot task during the optimization, the number of experienced query-prototype pairs is limited and thus the side effect of observation noises due to the high uncertainty limits the optimization. Therefore, modeling uncertainty is especially vital for few-shot learning with the limited samples, but still under-explored.

In this paper, we propose an efficient uncertainty-aware few-shot image classification framework to model uncertainty and perform uncertainty-aware optimization. For each query-prototype pair, we convert the observed feature similarity between the query and the prototype from a deterministic scalar to a distribution, with the variance of this distribution characterizing the uncertainty of the observed similarity. The similarity-based classification losses are calculated based on

[*]This work was done when Zhizheng Zhang was a visiting student at Columbia University.
[†]Corresponding Author

the Monte Carlo sampling on the similarity distributions to alleviate the influence of observation noises. Since the classification probability of a query sample is determined based on the similarity of this query sample with its N prototypes in the support set, this is a joint determination process. Thus, we adopt a graph-based model to jointly estimate the uncertainties for the N query-prototype pairs, which facilitates the information passing among them for the joint optimization.

In summary, our contributions lie in the following aspects:

- We are the first to explore and model the uncertainty of the similarities of query-prototype pairs in few-shot image classification, where a better understanding of the limited data is particularly important.
- We perform uncertainty-aware optimization to make the few-shot learning more robust to observation noises.
- We design a model to jointly estimate the uncertainty of the similarities between a query image and the prototypes in the support set.

We conduct comprehensive experiments and demonstrate the effectiveness of our proposed method in the popular inductive setting, where each query sample is evaluated independently from other query samples. Our scheme achieves the state-of-the-art performance on multiple datasets.

2 Related Work

2.1 Few-shot Image Classification

Few-shot image classification aims to recognize novel (unseen) classes upon limited labeled examples. Representative approaches can be summarized into four categories.

Classification-based methods train both a feature extractor and classifiers with meta-learning and learn a new classifier (*e.g.*, linear softmax classifier) for the novel classes during meta-testing [Chen *et al.*, 2019; Ren *et al.*, 2019; Lee *et al.*, 2019]. They in general need to update/fine-tune the network given a few-shot testing task.

Optimization-based methods exploit more efficient meta-learning optimization strategies for few-shot learning [Finn *et al.*, 2017; Grant *et al.*, 2018; Nichol *et al.*, 2018; Lee *et al.*, 2019]. MAML [Finn *et al.*, 2017] is a representative work, which trains models by aligning the gradients of several learning tasks, such that only a few iterations are needed to achieve rapid adaptation for a new task.

Hallucination-based methods enhance the performance by augmenting data, which alleviates data deficiency by using generated data. The generators transfer appearance variations [Hariharan and Girshick, 2017; Gao *et al.*, 2018] or styles [Antoniou *et al.*, 2018] from base classes to novel classes. Wang *et al*. create an augmented training set through a generator which is trained end-to-end along with the classification [Wang *et al.*, 2018b].

Similarity-based/Metric-based methods classify an unseen instance into its nearest class based on the similarities with a few labeled examples. They learn to project different instances to a metric space wherein the similarities among instances of the same category are larger than that of instances of different categories. Matching networks [Vinyals *et al.*, 2016] propose to assign an unlabeled data with the label of its nearest labeled data in one-shot learning. Furthermore, the prototypical network [Snell *et al.*, 2017] averages the features of several samples of the same category as the prototype of this class and classify unseen data by finding the nearest prototype in the support set, where a prototype is taken as the representation of a class. Relation networks [Sung *et al.*, 2018] propose an additional relation module which learns to determine whether a query and a prototype belongs to the same class and is jointly trained with deep feature representations.

Some recent metric-based works reveal the importance of learning good feature representations via classification-based pre-training for few-shot tasks [Chen *et al.*, 2019; Tian *et al.*, 2020; Chen *et al.*, 2020]. They propose strong baselines by training the network with classification supervision over all base categories and further performing meta-learning on sampled categories to simulate the few-shot setting. Chen *et al*. confirmed that the classification-based pre-training can provide extra transferability from base classes to novel classes in the meta-learning stage [Chen *et al.*, 2020].

In this paper, on top of such a strong baseline, we look into an important but still under-explored issue, *i.e.*, modeling and exploiting the inherent uncertainty of the calculated similarities for more effective optimization.

2.2 Uncertainty in Deep Learning

There are two main types of uncertainty studied for deep neural networks: aleatoric uncertainty, and epistemic uncertainty [Kendall and Gal, 2017; Gal, 2016; Kendall *et al.*, 2018]. Epistemic uncertainty captures model-related stochasticity, including parameters choices, architecture, *etc*. Aleatoric (Data-dependent) uncertainty captures the noise inherent in the observations [Kendall and Gal, 2017; Gal, 2016; Kendall *et al.*, 2018], which depends on the input sample of the model. A model usually has high uncertainty on noisy input or rarely seen input.

For few-shot classification, the "hardness" of a few-shot episode is quantified with a metric in [Dhillon *et al.*, 2020] but is only used to evaluate few-shot algorithms rather than improve the optimization. In [Wang *et al.*, 2020], a statistical approach is adopted to measure the credibility of predicted pseudo-labels for unlabeled samples, where the most trustworthy pseudo-labeled instances are selected to update the classifier in the semi-supervised or transductive few shot learning setting. There is still a lack of exploration of the uncertainty for more general inductive few-shot setting, where the network does not use/need testing samples for updating. In this paper, we model the data-dependent uncertainty of the calculated similarities between a query sample and the prototypes, and leverage it to better optimize the network. To our best knowledge, we are the first to explore the uncertainty of pair-similarity in few-shot classification towards more efficient optimization.

3 Proposed Method

We propose a Uncertainty-Aware Few-Shot (UAFS) image classification framework (see Figure 1), which models uncertainty of the query-prototype similarity and perform uncertainty-aware optimization. We first present the problem formulation and a strong baseline in Section 3.1. Then,

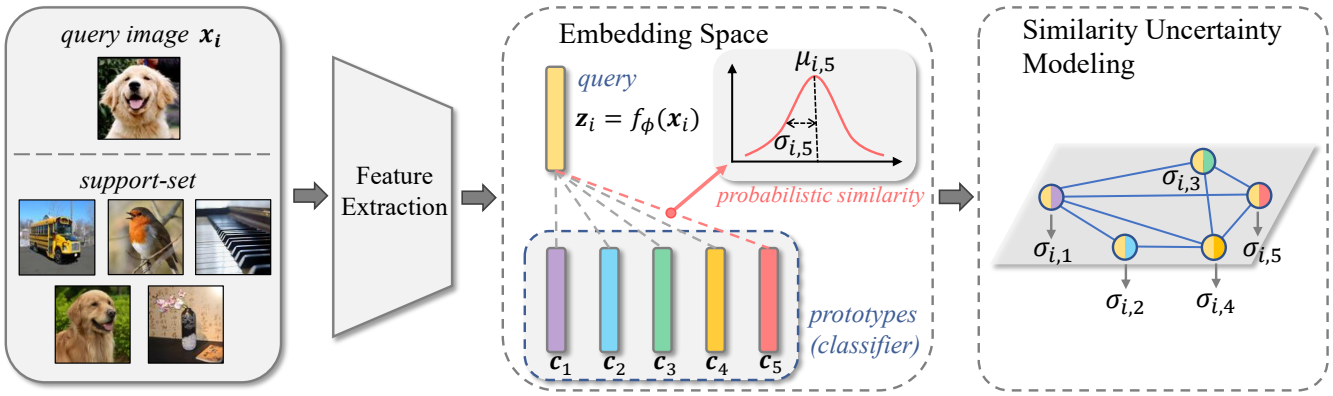

Figure 1: **Pipeline of proposed Uncertainty-Aware Few-Shot (UAFS) image classification.** To alleviate the influence of observation noises on the similarity of a query-prototype pair, rather than a scalar, we model the similarity between a query feature $\mathbf{z}_i$ and a prototype feature $\mathbf{c}_j$ (where $j = 1, \cdots, N$) by a distribution, *i.e.*, *probabilistic similarity*, based on graph-based similarity uncertainty estimation. For a query sample $\mathbf{x}_i$ and N prototypes in the support set, we take each query-prototype pair as a node and employ a graph-based model to jointly infer the similarity uncertainty $\sigma_{i,j}$ for each query-prototype pair. The network is optimized with the similarity-based classification losses which exploit the estimated uncertainty (not shown in this figure, please see the subsection around Eq. (5)).

we elaborate our proposed uncertainty-aware few-shot image classification in Section 3.2.

3.1 Preliminaries and a Strong Baseline

Problem Formulation. For few-shot image classification, all the categories/classes of the dataset are divided into base classes $\mathcal{C}_{base}$ for training and novel classes $\mathcal{C}_{novel}$ for testing without class overlapping [Snell *et al.*, 2017; Chen *et al.*, 2019; Dhillon *et al.*, 2020]. In few-shot learning, episodic training is widely used. Each N-way K-shot task randomly sampled from base classes is defined as an episode, where the support set $\mathcal{S}$ includes N classes with K samples per class, and the query set $\mathcal{Q}$ contains the same N classes with M samples per class. Our method is applied to the metric-based few-shot learning which performs similarity-based classification based on matching the prototypes in the support set for a given query sample.

Given an image $\mathbf{x}_i$, we feed it to a Convolution Neural Network (CNN) to obtain a feature map, and perform global averaging pooling to get a D-dimensional feature vector $\mathbf{z}_i = f_\phi(\mathbf{x}_i) \in \mathbb{R}^D$ as the embedding result of $\mathbf{x}_i$, where $f_\phi(\cdot)$ denotes the feature extractor. The prototype of a class indexed by k is calculated by averaging the feature embeddings of all the samples of this class in the support set as:

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in S_k} f_\phi(\mathbf{x}_i), \tag{1}$$

where $\mathbf{x}_i$ denotes the sample indexed by i and $\mathbf{y}_i$ denotes its label, S_k denotes the set of samples of the k-th class in the support set $\mathcal{S} = \{S_1, \cdots, S_N\}$, $|S_k|$ denotes the number of samples in S_k. For a query image $\mathbf{x}$, the probability of classifying it into the class k is:

$$p(\mathbf{y}_i = k|\mathbf{x}_i) = \frac{exp(\tau \cdot s(f_\phi(\mathbf{x}_i), \mathbf{c}_k))}{\sum_{j=1}^{N} exp(\tau \cdot s(f_\phi(\mathbf{x}_i), \mathbf{c}_j))}, \tag{2}$$

where the temperature parameter τ is a hyper-parameter, and N is the number of classes in the support set (*i.e.*, N-way). The $s(f_\phi(\mathbf{x}_i), \mathbf{c}_k)$ represents the similarity between the given query-sample $\mathbf{x}_i$ and the prototype $\mathbf{c}_k$ of the class k. Here, the N prototypes of N classes in the support set construct a similarity-based classifier.

Baseline. Many works [Vinyals *et al.*, 2016; Snell *et al.*, 2017; Chen *et al.*, 2019; Li *et al.*, 2019; Chen *et al.*, 2020] reveal that the core of metric-based few-shot learning is to learn a good feature embedding function $f_\phi(\cdot)$ from base classes $\mathcal{C}_{base}$ which could generalize well on unseen classes. We follow the latest work [Chen *et al.*, 2020] to build a strong baseline with two-stage training:

Stage-1: *Classification-based pre-training*. We employ a Fully-Connected (FC) layer as the classifier with cosine similarity over all base classes to perform pre-training. We train the feature extractor $f_\phi(\cdot)$ (backbone) and the classifier in an end-to-end manner using a standard cross-entropy loss.

Stage-2: *Meta-learning*. For each episode/task, we sample N-way K-shot from base classes for training, which can be understood as a simulation of few-shot test process. In each task, $N \times K$ images and $N \times M$ images are sampled to constitute the support set and the query set, respectively. We adopt the cross-entropy loss formulated by $\mathcal{L} = -\sum_{i=1}^{N \times M} \mathbf{y}_i \cdot \log(p(\mathbf{y}_i|\mathbf{x}_i))$, where $p(\mathbf{y}_i|\mathbf{x}_i)$ is calculated in Eq. (2) with $s(\cdot, \cdot)$ instantiated by cosine similarity.

3.2 Uncertainty-aware Few-shot Learning

A neural network has different uncertainties regarding its outputs for different input data and there will be observation noises for an instantiated output [Kendall and Gal, 2017]. For few-shot image classification, the model also has uncertainty on the estimated similarities for the query-prototypes pairs. The small number of samples in each few-shot episode exacerbate the side effect of observation noises from uncertainty. To address this, we model the data-dependent uncertainty of similarities and leverage it for better optimizing the network.

Probabilistic Similarity Representation. To characterize the uncertainty of similarity of a query-prototype pair, we

convert the similarity representation of this query-prototype pair from a deterministic scalar to a probabilistic representation, *i.e.*, a parameterized distribution.

Given a query sample $\mathbf{x}_i$ (with its feature $\mathbf{z}_i = f_\phi(\mathbf{x}_i)$ in the latent space) and a prototype $\mathbf{c}_j$ obtained by Eq. (1), instead of using a deterministic scalar, we model the similarity as a probability distribution. Similar to other works [Kendall and Gal, 2017; Gal, 2016], we simply model the distribution of the similarity between this query and the prototype $\mathbf{c}_j$ by a Gaussian distribution with mean μ_{ij} and variance σ_{ij}^2:

$$p(s_{ij}|\mathbf{z}_i, \mathbf{c}_j) = \mathcal{N}(s_{ij}; \mu_{ij}, \sigma_{ij}^2 I), \tag{3}$$

where $\mathcal{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_N\}$ denotes the N prototypes in the support set, $\mathbf{c}_j$ is the j-th prototype within it.

We estimate the mean of the similarity by their inner product, *i.e.*, $\mu_{ij} = \langle \mathbf{z}_i, \mathbf{c}_j \rangle$, which denotes the most likely value of the similarity between $\mathbf{z}_i$ and $\mathbf{c}_i$. The variance σ_{ij}^2 is used to characterize the uncertainty of this pairwise similarity. We estimate it by a graph neural network, which will be elaborated in the *subsection "Similarity Uncertainty Estimation"*.

Uncertainty-aware Optimization. With the similarity of a query-support pair being represented by a distribution rather than a scalar, we incorporate such probabilistic similarity representation in the classification loss for better optimization, which could be more robust to the observation noises.

The analytic solution of integrating over these distributions for the optimization of classification loss function is difficult. Following [Kendall and Gal, 2017], we thus approximate the optimization objective by Monte Carlo integration. Particularly, Monte Carlo sampling is performed on the N similarity distributions. For a given query image $\mathbf{x}_i$ and N prototypes $\mathbf{c}_j, j = 1, \cdots, N$ which form N similarity pairs, we repeat T random sampling over the similarity distributions for each query-prototype pair s_{ij} (see (4) with $t = 1, \cdots, T$) to obtain statistical results. We enable a differentiable representation of the probabilistic similarity for a query and prototype pair s_{ij} by re-parameterizing it as:

$$s_{ij,t} = \mu_{ij} + \sigma_{ij}\epsilon_t, \quad \epsilon_t \in \mathcal{N}(0, 1). \tag{4}$$

For a given query image $\mathbf{x}_i$ with groundtruth label $\mathbf{y}_i = k$, we obtain its corresponding classification loss as:

$$\mathcal{L}(\mathbf{x}_i, \mathbf{y}_i = k) = -\log(\frac{1}{T}\sum_{t=1}^{T}(e^{s_{ik,t}} / \sum_{j=1}^{N} e^{s_{ij,t}})). \tag{5}$$

Similarity Uncertainty Estimation. We characterize the uncertainty by variance σ_{ij}^2 as mentioned before. Hereafter, we elaborate on how to estimate the uncertainty.

For a N-way similarity-based classification, the given query is compared with all the N prototypes and the class of the prototype with the highest similarity value is taken as the predicted class, which is *a joint determination process*.

The joint determination nature inspires us to estimate uncertainty parameter σ_{ij} for each pair after taking a global glance of the N pairs with $j = 1, \cdots, N$. Therefore, we design a graph-based model as the similarity uncertainty estimator to jointly infer all the σ_{ij} for a given query $\mathbf{x}_i$ and N prototypes $\mathbf{c}_j$. This enables the exploration of global scope information for the current task to jointly estimate the similarity uncertainties. Besides, the graph-based design has another advantage, *i.e.*, its scalability to the number of categories. This allows us to pre-train the parameters of the graph-based model it in Stage-1. We will demonstrate its effectiveness by comparing it with other alternative designs: 1) using Convolutional Neural Network (CNN) to infer σ_{ij} independently for each pair; 2) adopting Fully-Connected (FC) based model that is able to exploit global information for N pairs but cannot handle the consistency classes in different episodics well.

Given a query and N prototypes as illustrated in Fig. 1, we build a graph of N nodes, with the j^{th} node denoted by the information of the query and the j^{th} prototype. The output of the node is the predicted uncertainty of the similarity of this query-prototype pair. We feed a query image $\mathbf{x}_i$ to the feature extractor to get its embedding vector $\mathbf{z}_i = f_\phi(\mathbf{x}_i) \in \mathbb{R}^D$. Similarly, we get the N prototypes of the support set as $\mathbf{c}_j \in \mathbb{R}^D, j = 1, \cdots, N$. We use the group-wise relations/similarities between the query and the j^{th} prototype feature to represent the information of this query-prototype pair. Specifically, we split both $\mathbf{z}_i$ and $\mathbf{c}_j$ into L groups along their channel dimensions to have $\{z_i^l | l = 1, \cdots, L\}$ and $\{c_j^l | l = 1, \cdots, L\}$, where $L \leq D$. We calculate the cosine similarity of the l^{th} group as $r_{ij}^l = \frac{z_i^l {c_j^l}^T}{\|z_i^l\| \|c_j^l\|}$, and stack them to be a relation feature vector $\mathbf{v}_{ij} = [r_{ij}^1, \cdots, r_{ij}^L] \in \mathbb{R}^L$. Note that the relations/similarities observed from multiple groups provide valuable hints of the uncertainty since they reflect the similarity from different perspectives.

The N nodes in the graph are represented by a matrix $V_i = (\mathbf{v}_{i1}; \cdots; \mathbf{v}_{iN}) \in \mathbb{R}^{N \times L}$. We use graph convolutional network (GCN) to learn the similarity uncertainty for each node. Similar to [Shi *et al.*, 2019; Wang *et al.*, 2018a; Wang and Gupta, 2018], the edge from the j^{th} node to the j'^{th} node is modeled by their affinity in the embedded space, which is formulated as:

$$E_i(j, j') = \varphi_1(\mathbf{v}_{ij})\varphi_2(\mathbf{v}_{ij'})^T, \tag{6}$$

where φ_1 and φ_2 denote two embedding functions implemented by FC layers. All edges constitute an adjacent matrix denoted by $E_i \in \mathbb{R}^{N \times N}$. We normalize each row of E_i with SoftMax function so that all the edge values connected to a target node is 1, yielding a numerically-stable message passing through the modeled graph. Similar to [Wang *et al.*, 2018a], we denote the normalized adjacent matrix by G_i and update the nodes through GCN as:

$$V_i' = V_i + Y_i W_y, \ \ where \ \ Y_i = G_i V_i W_v, \tag{7}$$

$W_v \in \mathbb{R}^{L \times L}$ and $W_y \in \mathbb{R}^{L \times L}$ are two learnable transformation matrices. W_v is implemented by a 1×1 convolutional layer. W_y is implemented by two stacked blocks. Each block consists of a 1×1 convolution layer, followed by a Batch Normalization (BN) layer and an LeakyReLU activation layer. We thus infer the similarity uncertainty vector $\mathbf{u}_i = [\sigma_{i1}, \cdots, \sigma_{iN}] \in \mathbb{R}^N$ for the N pairs from the updated graph nodes, which can be formulated as:

$$\mathbf{u}_i = \alpha(\mathrm{BN}(V_i' W_{u1}))W_{u2}, \tag{8}$$

Model	Stage1	Stage2	mini-ImageNet		tiered-ImageNet		CIFAR-FS		FC-100	
			1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
1	w/o U	no	59.28 ± 0.81	78.26 ± 0.60	67.02 ± 0.84	83.56 ± 0.61	71.95 ± 0.81	84.21 ± 0.58	39.42 ± 0.77	54.12 ± 0.79
2	w U	no	61.53 ± 0.83	78.83 ± 0.60	68.77 ± 0.82	83.92 ± 0.42	73.81 ± 0.78	84.54 ± 0.63	41.76 ± 0.52	56.63 ± 0.24
3 (Meta-Base)	w/o U	w/o U	63.10 ± 0.85	79.44 ± 0.65	67.72 ± 0.80	83.61 ± 0.62	72.36 ± 0.67	84.43 ± 0.50	40.23 ± 0.22	56.16 ± 0.56
4	w U	w/o U	63.16 ± 0.66	79.56 ± 0.67	68.92 ± 0.77	83.96 ± 0.56	73.10 ± 0.70	84.68 ± 0.48	40.34 ± 0.59	56.24 ± 0.55
5	w/o U	w U	62.57 ± 0.33	79.35 ± 0.49	69.11 ± 0.83	84.12 ± 0.55	72.51 ± 0.68	84.46 ± 0.49	39.49 ± 0.33	53.24 ± 0.24
6 (Meta-UAFS)	w U	w U	$\mathbf{64.22 \pm 0.67}$	$\mathbf{79.99 \pm 0.49}$	$\mathbf{69.13 \pm 0.84}$	$\mathbf{84.33 \pm 0.59}$	$\mathbf{74.08 \pm 0.72}$	$\mathbf{85.92 \pm 0.42}$	$\mathbf{41.99 \pm 0.58}$	$\mathbf{57.43 \pm 0.38}$

Table 1: Comparison on applying the proposed similarity uncertainty modeling and optimization in different training stages. "Stage1" refers to the *classification-based pre-training* stage while "Stage2" refers to the *meta-learning* stage as described in the manuscript. "no" denotes the corresponding training stage is not used. "w U" denotes that we apply the proposed similarity *uncertainty* modeling and optimization (UMO) in the corresponding training stage while the "w/o U" denotes we do not apply it.

where the j^{th}-dimension of $\mathbf{u}_i$ is the aforementioned σ_{ij}, denoting the similarity uncertainty for the query $\mathbf{x}_i$ and the j^{th} prototype. $W_{u1} \in \mathbb{R}^{L \times L}$ and $W_{u2} \in \mathbb{R}^{L \times 1}$ are transformation matrices both implemented by a 1×1 convolutional layer. "BN" refers to the Batch Normalization layer and $\alpha(\cdot)$ refers to the LeakyReLU activation function.

Discussion: We model pairwise uncertainty for the similarity instead of for a sample itself like in other tasks [Kendall and Gal, 2017]. This is because the classification of a query image is influenced by not only the query itself but also the prototypes in the support set. We further conduct extensive experiments to compare the effectiveness of modeling feature uncertainty (query itself) vs. similarity uncertainty (query and prototypes) for few-shot image classification in our experiments. Unlike the previous work [Garcia and Bruna, 2018], which adopts graph-based model to propagate information from labeled samples to unlabeled ones thus delivers a transductive solution, we use graph-based model to achieve the joint estimation of uncertainty of N query-prototype pairs for a given query (non-transductive), where the uncertainty is used to better optimize the network. We do not introduce any complexity increase in the testing where the GCN-based uncertainty estimator is discarded. In contrast, the GCN in [Garcia and Bruna, 2018] is a part of their network and is needed during testing.

Joint Training. Similar to the baseline configuration described in Section 3.1, we train the entire network in two stages: *classification-based pre-training* and *meta-learning*.

In the *classification-based pre-training* stage, the FC coefficients can be considered as the prototypes and the number of input nodes in GCN is the total number of classes in $\mathcal{C}_{base}$. In the *meta-learning* stage, the number of input nodes in GCN is N. Because our graph-based uncertainty estimator has the scalability to the number of involved categories for classification, the parameters in GCN are shared in the two stages. We fine-tune the backbone and the similarity uncertainty estimator in the *meta training* stage.

4 Experiments

4.1 Dataset and Implementation

Datasets. For few-shot image classification, we conduct experiments on four public benchmark datasets: mini-ImageNet [Vinyals *et al.*, 2016], tiered-ImageNet [Ren *et al.*, 2018], CIFAR-FS [Bertinetto *et al.*, 2018], and FC100 [Oreshkin *et al.*, 2018].

Methods	U-Estimator	mini-ImageNet		CIFAR-FS	
		1-shot	5-shot	1-shot	5-shot
Meta-Base	-	63.10 ± 0.85	79.44 ± 0.65	72.36 ± 0.67	84.43 ± 0.50
B+ SampleU	Conv-based	63.25 ± 0.81	79.39 ± 0.63	72.11 ± 0.80	84.26 ± 0.52
B+ SimiU	FC-based	57.42 ± 0.80	77.01 ± 0.66	70.33 ± 0.84	79.58 ± 0.78
B+ SimiU	Conv-based	63.31 ± 0.68	79.63 ± 0.46	73.22 ± 0.75	84.40 ± 0.69
B+ SimiU	Graph-based	$\mathbf{64.22 \pm 0.67}$	$\mathbf{79.99 \pm 0.49}$	$\mathbf{74.08 \pm 0.72}$	$\mathbf{85.92 \pm 0.42}$

Table 2: Comparison on different uncertainty modelling methods. "SampleU" denotes that we estimate the uncertainty of *a query sample* while "SimiU" denotes that we estimate the uncertainty of *the similarity between a query and its prototype*. Besides, we compare different designs of the uncertainty estimator (U-Estimator), including using CNNs ("Conv-based") to process each sample or each query-prototype pair independently, using GCNs ("Graph-based") to process the N query-prototype pairs jointly. "B" denotes the abbreviation of "Meta-Base".

Networks and Training. Following recent common practices [Oreshkin *et al.*, 2018; Lee *et al.*, 2019], we adopt a wider ResNet-12 with more channels as the backbone in our work unless claimed otherwise. We build our strong baseline (see Section 3.1) with two-stage training by following [Chen *et al.*, 2020]. For all our models, the classification temperature τ in (2) is a trainable parameter which is initialized with 10. For the parameter L, we found the performance is very similar when it is in the range of 16 to 64 and we set it to 32.

Evaluation. We conduct all experiments in the inductive (non-transductive) setting (*i.e.*, each query sample is evaluated independently from other query samples). We create each few-shot episode by uniformly sampling 5 classes (*i.e.*N=5) from the test set and further uniformly sampling support and query samples for each sampled class accordingly. Consistent with other works, we report the mean accuracy and the standard deviation with a 95% confidence interval over 1000 randomly sampled few-shot episodes.

4.2 Ablation Study

Effectiveness of Proposed UAFS. Table 1 shows the ablation study on the effectiveness of our proposed method. Model-3 corresponds to our strong baseline *Meta-Base* (see Table 3) with two-stage training. Model-6 corresponds to our final scheme *Meta-UAFS* with the proposed similarity uncertainty modeling and optimization (UMO).

We have the following observations. **1)** Our proposed *Meta-UAFS* (Model-6) improves the strong baseline *Meta-Base* (Model-3) by 1.12%, 1.41%, 1.72% and 1.76% in 1-shot classification on four datasets respectively, which demonstrates the effectiveness of our UMO. **2)** *Meta-Base* (Model-

Methods	Backbone	mini-ImageNet		tiered-ImageNet		CIFAR-FS		FC-100	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchingNet [Vinyals *et al.*, 2016]	64-64-64-64	46.6	60.0	-	-	-	-	-	-
MAML [Finn *et al.*, 2017]	32-32-32-32	48.70 ± 1.84	63.11 ± 0.92	51.67 ± 1.81	70.30 ± 1.75	-	-	-	-
ProtoNet† [Snell *et al.*, 2017]	64-64-64-64	49.42 ± 0.78	68.20 ± 0.66	53.31 ± 0.89	72.69 ± 0.74	-	-	-	-
RelationNet [Sung *et al.*, 2018]	64-96-128-256	50.44 ± 0.82	65.32 ± 0.70	54.48 ± 0.93	71.32 ± 0.78	-	-	-	-
TADAM [Oreshkin *et al.*, 2018]	ResNet-12	58.50 ± 0.30	76.70 ± 0.30	-	-	-	-	40.10 ± 0.40	56.10 ± 0.40
LEO† [Rusu *et al.*, 2019]	WRN-28-10	61.76 ± 0.08	77.59 ± 0.10	66.33 ± 0.05	81.44 ± 0.09	-	-	-	-
TapNet [Yoon *et al.*, 2019]	ResNet-12	61.65 ± 0.15	76.36 ± 0.10	63.08 ± 0.15	80.26 ± 0.12	-	-	-	-
Shot-free [Ravichandran *et al.*, 2019]	ResNet-12	59.04 ± 0.43	$77.64 \pm 0:39$	66.87 ± 0.43	82.64 ± 0.39	69.20 ± 0.40	84.70 ± 0.40	-	-
MetaOptNet [Lee *et al.*, 2019]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53	72.00 ± 0.70	84.20 ± 0.50	41.10 ± 0.60	55.50 ± 0.60
CAN [Hou *et al.*, 2019]	ResNet-12	63.85 ± 0.48	79.44 ± 0.34	$\mathbf{69.89 \pm 0.51}$	84.23 ± 0.37	-	-	-	-
Baseline2020 [Dhillon *et al.*, 2020]	WRN-28-10	56.17 ± 0.64	73.31 ± 0.53	67.45 ± 0.70	82.88 ± 0.53	70.26 ± 0.70	83.82 ± 0.49	36.82 ± 0.51	49.72 ± 0.55
MetaBaseline [Chen *et al.*, 2020][1]	ResNet-12	63.17 ± 0.23	79.26 ± 0.17	68.62 ± 0.27	83.29 ± 0.18	-	-	-	-
Meta-Base	ResNet-12	63.10 ± 0.85	79.44 ± 0.65	67.72 ± 0.80	83.61 ± 0.62	72.36 ± 0.67	84.43 ± 0.50	40.23 ± 0.22	56.16 ± 0.56
Meta-UAFS	ResNet-12	$\mathbf{64.22 \pm 0.67}$	$\mathbf{79.99 \pm 0.49}$	69.13 ± 0.84	$\mathbf{84.33 \pm 0.59}$	$\mathbf{74.08 \pm 0.72}$	$\mathbf{85.92 \pm 0.42}$	$\mathbf{41.99 \pm 0.58}$	$\mathbf{57.43 \pm 0.38}$

Table 3: Accuracy (%) comparison for 5-way few-shot classification of our schemes, our baselines and the state-of-the-arts on four benchmark datasets. For the backbone network, "l_1-l_2-l_3-l_4" denotes a 4-layer convolurional network with the number of convolutional filters in each layer, respectively. "Meta-Base" and "Meta-UAFS" refer to the strong baseline model and our final UAFS model trained with *classification-based pre-training* stage and *meta-learning* stage, respectively. The superscript † refers to that the model is trained on the combination of training and validation sets while others only use the training set. Note that for fair comparisons, all the presented results are from inductive (non-transductive) setting. Bold numbers denotes the best performance while numbers with underlines denotes the second best performance.

3) obviously outperforms Model-1 that has only Stage1 training, demonstrating the effectiveness of two-stage training. **3)** Enabling UMO in the *classification-based pre-training stage* (Stage1) brings obvious improvement for one-stage training (Model-2 vs. Model-1), but only slight gains for two-stage training (Model-4 vs. Model-3). The higher the performance, the more difficult it is to obtain gain. On top of the strong baseline, the UMO is not helpful for Stage1 which does not suffer from limited training data as Stage2. **4)** Adopting UMO only in Stage2 (Model-5) does not always bring improvements (when compared with *Meta-Base*), while applying UMO on both stages (*Meta-UAFS*) brings significant improvements. For Model-5, the uncertainty estimator is trained only in Stage2 (from scratch without pre-training in Stage1) and may be difficult to optimize. Our graph-based uncertainty estimator has the scalability to the number of involved categories for classification, which allows us to pre-train the parameter-sharable uncertainty estimator in Stage1 and fine-tune it in Stage2. When we enable the pre-training of uncertainty estimator in Stage1, our final scheme (*Meta-UAFS*) consistently delivers the best results. Note that the backbone network is trained in both stages.

Similarity Uncertainty Estimator Designs. Since the query is classified in a joint determination by comparing it with the N prototypes simultaneously, we propose to use a GCN to jointly estimate the uncertainties. This design is superior because it 1) enables the joint estimation of similarity uncertainties and 2) has the scalability to the number of categories which facilitates the pre-training of the uncertainty estimator in Stage-1. Table 2 shows that 1) our *Graph-based* design (last row) is superior to *Conv-based* design (the penultimate row) which uses 1×1 convolutional layers to independently infer the uncertainty for each query-prototype pair. FC-based design is inferior to our proposed design.

Sample Uncertainty vs. Pair Similarity Uncertainty. Table 2 shows that the scheme *B+SimiU* with Graph-based uncertainty estimator which models the *uncertainty of the query-prototype similarity* is more effective than the scheme *B+SampleU* which models the *uncertainty of a sample*. That is because in the few shot classification, for a given query, its estimated class depends on both the query and the N prototypes. Therefore, we model the uncertainty of the similarity between the query and the prototypes.

4.3 Comparison with State-of-the-arts

In Table 3, we compare our proposed *Meta-UAFS* with the state-of-the-art approaches. Note that our baseline *Meta-Base* is the same as MetaBaseline [Chen *et al.*, 2020] and they have the similar performance. Compared to *Meta-Base* [Chen *et al.*, 2020], our *Meta-UAFS* achieves significant improvement of 1.12%, 1.41%, 1.72%, and 1.76% in 1-shot accuracy on mini-ImageNet, tiered-ImageNet, CIFAR-FS, and FC00, respectively for 1-shot classification. Our *Meta-UAFS* achieves the state-of-the-art performance. CAN [Hou *et al.*, 2019] introduces cross attention module to highlight the target object regions, making the extracted feature more discriminative. Our uncertainty modeling and optimization which reduces the side effects of observation noises is complementary to their attention design and incorporating their attention design into our framework would lead to superior performance.

5 Conclusion

In this paper, we propose an Uncertainty-Aware Few-Shot image classification approach where data-dependent uncertainty modeling is introduced to alleviate the side effect of observation noise. Particularly, we convert the similarity between a query sample and a prototype from a deterministic value to a probabilistic representation and optimize the networks by exploiting the similarity uncertainty. We design a graph-based uncertainty estimator to jointly estimate the similarity uncertainty of the query-support pairs for a given query sample. Extensive experiments demonstrate the effectiveness of our proposed method and our scheme achieves state-of-the-arts performance on all the four benchmark datasets.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China 2018AAA0101400 and NSFC under Grant U1908209, 61632001.

References

[Antoniou *et al.*, 2018] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. In *ICLR*, 2018.

[Bertinetto *et al.*, 2018] Luca Bertinetto, Joao F Henriques, Philip HS Torr, et al. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

[Chang *et al.*, 2020] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. Data uncertainty learning in face recognition. In *CVPR*, 2020.

[Chen *et al.*, 2019] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019.

[Chen *et al.*, 2020] Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020.

[Dhillon *et al.*, 2020] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *ICLR*, 2020.

[Dong and Xing, 2018] Nanqing Dong and Eric Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, 2018.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[Gal, 2016] Yarin Gal. *Uncertainty in deep learning*. PhD thesis, University of Cambridge, 2016.

[Gao *et al.*, 2018] Hang Gao, Zheng Shou, Alireza Zareian, Hanwang Zhang, and Shih-Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. In *NeurIPS*, 2018.

[Garcia and Bruna, 2018] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. In *ICLR*, 2018.

[Grant *et al.*, 2018] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

[Hariharan and Girshick, 2017] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017.

[Hou *et al.*, 2019] Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *NeurIPS*, 2019.

[Huang and Tao, 2019] Shaoli Huang and Dacheng Tao. All you need is a good representation: A multi-level and classifier-centric representation for few-shot learning. *arXiv preprint arXiv:1911.12476*, 2019.

[Karlinsky *et al.*, 2019] Leonid Karlinsky, Joseph Shtok, Sivan Harary, et al. Repmet: Representative-based metric learning for classification and few-shot object detection. In *CVPR*, 2019.

[Kendall and Gal, 2017] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017.

[Kendall *et al.*, 2015] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[Kendall *et al.*, 2018] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.

[Lee *et al.*, 2019] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.

[Li *et al.*, 2019] 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.

[Mishra *et al.*, 2018] Ashish Mishra, Vinay Kumar Verma, M Shiva Krishna Reddy, S Arulkumar, Piyush Rai, and Anurag Mittal. A generative approach to zero-shot and few-shot action recognition. In *WACV*, 2018.

[Nichol *et al.*, 2018] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[Oreshkin *et al.*, 2018] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018.

[Ravichandran *et al.*, 2019] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *ICCV*, 2019.

[Ren *et al.*, 2018] Mengye Ren, Eleni Triantafillou, Sachin Ravi, et al. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

[Ren *et al.*, 2019] Mengye Ren, Renjie Liao, Ethan Fetaya, and Richard Zemel. Incremental few-shot learning with attention attractor networks. In *NIPS*, 2019.

[Rusu *et al.*, 2019] Andrei A Rusu, Dushyant Rao, et al. Meta-learning with latent embedding optimization. In *ICLR*, 2019.

[Shi *et al.*, 2019] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[Sung *et al.*, 2018] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, et al. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

[Tian *et al.*, 2020] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020.

[Vinyals *et al.*, 2016] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016.

[Wang and Gupta, 2018] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018.

[Wang *et al.*, 2018a] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[Wang *et al.*, 2018b] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, 2018.

[Wang *et al.*, 2020] Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *CVPR*, 2020.

[Yoon *et al.*, 2019] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *ICML*, 2019.