META-LEARNED CONFIDENCE FOR TRANSDUCTIVE FEW-SHOT LEARNING

Anonymous authors
Paper under double-blind review

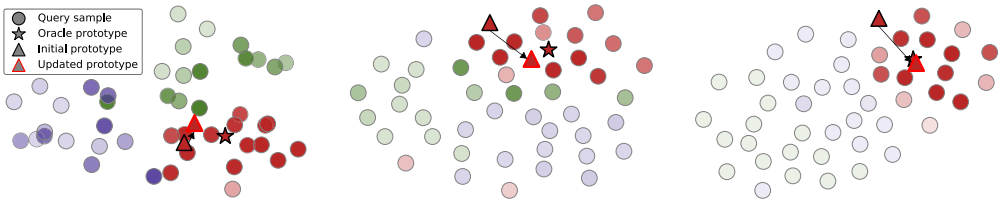
ABSTRACT

Transductive inference is an effective means of tackling the data deficiency problem in few-shot learning settings. A popular transductive inference technique for few-shot metric-based approaches, is to update the prototype of each class with the mean of the most confident query examples, or confidence-weighted average of all the query samples. However, a caveat here is that the model confidence may be unreliable, which may lead to incorrect predictions. To tackle this issue, we propose to meta-learn the confidence for each query sample, to assign optimal weights to unlabeled queries such that they improve the model's transductive inference performance on unseen tasks. We achieve this by meta-learning an input-adaptive distance metric over a task distribution under various model and data perturbations, which will enforce consistency on the model predictions under diverse uncertainties for unseen tasks. We validate our few-shot learning model with meta-learned confidence on four benchmark datasets, on which it largely outperforms strong recent baselines and obtains new state-of-the-art results. Further application on semi-supervised few-shot learning tasks also yields significant performance improvements over the baselines.

1 INTRODUCTION

Few-shot learning, the problem of learning under data scarcity, is an important challenge in deep learning as large number of training instances may not be available in many real-world settings. While the recent advances in meta-learning made it possible to obtain impressive performance on few-shot learning tasks (Hou et al., 2019; Li et al., 2019; Lifchitz et al., 2019), it still remains challenging in cases where we are given very little information (e.g. one-shot learning). Some of the metric-based meta-learning approaches tackle this problem using *transductive learning* or *semi-supervised learning*, by leveraging the structure of the unlabeled instances at the inference time (Hou et al., 2019; Kim et al., 2019; Li et al., 2019; Ren et al., 2018). Popular approach for these problem includes leveraging nearest neighbor graph for propagating labels (Kim et al., 2019; Liu et al., 2018; Yang et al., 2020), or using predicted soft or hard labels on unlabeled samples to update the class prototype (Hou et al., 2019; Ren et al., 2018). However, all these transductive or semi-supervised inference approaches are fundamentally limited by the intrinsic *unreliability* of the labels predicted on the unseen samples.

In this work, we aim to tackle this problem by proposing a novel confidence-based transductive inference scheme for metric-based meta-learning models. The most challenging problem is that the confidence prediction on the test instances for *unseen* task should be inevitably unreliable, since the samples come from an unknown distribution. To account for such uncertainties of prediction on an unseen task, we first propose to generate various model and data perturbations, such as random dropping of residual blocks and random augmentations. This randomness helps the model better learn the confidence measure by considering various uncertainties for an unseen task (see Figure 1), and also allows us to take an ensemble over the confidence measures under random perturbations at test time. In order to enhance learning confidence, we further meta-learn the distance metric (or metric) to assign different confidence scores to each query (or test) instance for each class, such that the updated prototypes obtained by confidence-weighted averaging of the queries improve classification of the query samples. This is done by learning a metric length-scale term for each individual instance or a pair of instances. We refer to this transductive inference using meta-learned input-adaptive confidence under various perturbations as *Meta-Confidence Transduction* (MCT).



(a) ProtoNets (+4.44%) (b) Instance-wise metric (+8.89%) (c) MCT (+15.56%)

Figure 1: Transductive inference with confidence scores. (a) ProtoNets with Euclidean distance; (b) ProtoNets with instance-wise metric; (c) ProtoNets with model/data perturbations and instance-wise metric; We visualize t-SNE embeddings on a 3-way 1-shot task, where each color stands for different class. The numbers show the accuracy increase after transduction for this task. The transparency shows the confidence scores for *red* class.

We validate our transductive inference scheme for metric-based meta-learning models on four benchmark datasets against existing transductive approaches, which shows that the models using meta-learned confidence significantly outperform existing transductive inference methods, and obtain new state-of-the-art results. We further verify the generality of our MCT on semi-supervised learning tasks, where we assign confidence scores to unlabeled data. The results show that MCT outperforms relevant baselines by large margins, which shows the efficacy of our method. Further ablation studies show that both meta-learning of various perturbations and input-adaptive distance metric are crucial in the success of our method in assigning correct confidence to each test sample.

Our main contributions are as follows:

- We propose to meta-learn the confidence with various types of **model** and **data perturbations** during meta-learning, such that the meta-learned confidence can better account for uncertainties at unseen tasks.
- We further propose to meta-learn an **input-adaptive distance metric**, which allows to output an accurate and reliable confidence for an unseen test samples that can directly improve upon the transductive inference performance.
- We validate our model on four benchmark datasets for few-shot classification and achieve **new state-of-the-art** results, largely outperforming all baselines. Further experimental validation of our model on semi-supervised few-shot learning also verifies its efficacy.

2 Related Work

Distance-based meta-learning for few-shot classification The goal of few-shot classification is to correctly classify query set examples given only a handful of support set examples. Due to its limited amount of data, each task-specific classifier should resort to the meta-knowledge accumulated from the previous tasks, which is referred to as meta-learning (Thrun & Pratt, 1998). Meta-learning of few-shot classification can roughly be divided into several categories such as optimization-based method (Finn et al., 2017; Grant et al., 2018; Lee & Choi, 2018; Ravi & Larochelle, 2017; Rusu et al., 2019; Zintgraf et al., 2019), distance-based approaches (Snell et al., 2017; Sung et al., 2018; Vinyals et al., 2016), class or task-wise network modulation with amortization (Gordon et al., 2018; Requeima et al., 2019), or some combination of those approaches (Das & Lee, 2019; Na et al., 2020; Mangla et al., 2020; Triantafillou et al., 2019). We use a distance-based approach in this work, which allows us to directly compare distance between examples on a metric space. For example, Matching Networks (Vinyals et al., 2016) use cosine distance, whereas Prototypical Networks (Snell et al., 2017) use euclidean distance with each class prototype set to the mean of support embeddings.

Transductive learning Since few-shot classification is intrinsically challenging, we may assume that we can access other unlabeled query examples, which is called transductive learning (Vapnik, 1998). Here we name a few recent works. TPN (Liu et al., 2018) constructs a nearest-neighbor graph and propagate labels to pseudo-label the unlabeled query examples. EGNN (Kim et al., 2019) similarly constructs a nearest-neighbor graph, but utilizes both edge and node features in the update steps. On the other hand, Hou et al. (2019) tries to update class prototypes by picking top-k confident queries with their own criteria. Our approach also updates class prototypes for each transduction step, but makes use of all the query examples instead of a small subset of k examples.

Semi-supervised learning In the few-shot classification, semi-supervised learning can access additional large amount of unlabeled data. Ren et al. (2018) proposed several variants of soft k-means method in Prototypical Networks (Snell et al., 2017), where soft label is predicted confidence of unlabeled sample. Li et al. (2019) proposed the self-training method with pseudo labeling module based on gradient descent approaches (Finn et al., 2017; Sun et al., 2019). Basically, if an unlabeled query set is used for few-shot classification instead of an additional unlabeled set, it becomes transductive learning, and vice versa. Our approach has connection to soft k-means method of Ren et al. (2018), but we predict the confidence with input-adaptive distance metric and use meta-learned confidence under various perturbations.

3 PRELIMINARIES

3.1 FEW-SHOT CLASSIFICATION

We start by introducing notations. In the conventional C-way N-shot classification, we first sample C classes randomly from the entire set of classes, and then sample N and M examples from each class for the support set and query set, respectively. We define this sampling distribution as $p(\tau)$. As a result, we have a support set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{C \times N}$ and query set $\mathcal{Q} = \{(\tilde{\mathbf{x}}_i, \tilde{y}_i)\}_{i=1}^{C \times M}$, where $y, \tilde{y} \in \{1, \dots, C\}$ are the class labels. If some portion of the support set is unlabeled, then the problem becomes semi-supervised learning. The convention for the evaluation of few-shot classification models is to use $N \in \{1, 5\}$ (i.e. 1- or 5-shot) and $M = 15$.

The goal of few-shot classification is to correctly classify query examples in $\mathcal{Q}$ given the support set $\mathcal{S}$. Since $\mathcal{S}$ includes only a few examples for each class, conventional learning algorithms will mostly fail due to overfitting (e.g. consider 1-shot classification). Thus, most existing approaches tackle this problem by meta-learning over a task distribution $p(\tau)$, such that the later tasks can benefit from the knowledge obtained over the previous training episodes.

One of the most popular and successful approaches for few-shot classification is the metric-based approach, in which we aim to learn an embedding function $f_\theta(\mathbf{x}) \in \mathbb{R}^l$ that maps an input $\mathbf{x}$ to a latent embedding $\mathbf{z}$ in an l-dimensional metric space (which is usually the penultimate layer of a convolutional network). Support set and query examples are then mapped into this space, such that we can measure the distance between class prototypes and query embeddings.

3.2 TRANSDUCTIVE INFERENCE WITH SOFT k-MEANS

We now describe and discuss transductive inference using the confidence scores of query examples computed by soft k-means algorithm (Ren et al., 2018). Suppose that we are given an episode consisting of support set $\mathcal{S}$ and query set $\mathcal{Q}$. We also define $\mathcal{S}_c$ as the set of support examples in class c and $\mathcal{Q}_x = \{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_{C \times M}\}$ as the set of all query instances. Starting from Prototypical Networks (Snell et al., 2017), we first compute the initial prototype $P_c^{(0)} = \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} f_\theta(\mathbf{x})$ for each class $c = 1, \dots, C$. Then, for each step $t = 1, \dots, T$, and for each query example $\tilde{\mathbf{x}} \in Q_x$, we compute its confidence score, which denote the probability of it belonging to each class c, as follows:

$$q_c^{(t-1)}(\tilde{\mathbf{x}}) = \frac{\exp(-d(f_\theta(\tilde{\mathbf{x}}), P_c^{(t-1)}))}{\sum_{c'=1}^{C} \exp(-d(f_\theta(\tilde{\mathbf{x}}), P_{c'}^{(t-1)}))} \tag{1}$$

where $d(\cdot, \cdot)$ is Euclidean distance and $P^{(t-1)}$ denotes $t-1$ steps updated prototype. We then update the prototypes of class c based on the confidence scores (or soft labels) $q_c^{(t-1)}(\tilde{\mathbf{x}})$ *for all* $\tilde{\mathbf{x}} \in \mathcal{Q}_x$:

$$P_c^{(t)} = \frac{\sum_{\mathbf{x} \in \mathcal{S}_c} 1 \cdot f_\theta(\mathbf{x}) + \sum_{\tilde{\mathbf{x}} \in \mathcal{Q}_x} q_c^{(t-1)}(\tilde{\mathbf{x}}) \cdot f_\theta(\tilde{\mathbf{x}})}{\sum_{\mathbf{x} \in \mathcal{S}_c} 1 + \sum_{\tilde{\mathbf{x}} \in \mathcal{Q}_x} q_c^{(t-1)}(\tilde{\mathbf{x}})} \tag{2}$$

which is the weighted average that we previously mentioned. Note that the confidence of the support examples is always 1, since their class labels are observed. We repeat the process until $t = 1, \dots, T$.

Questions However, confidence-based transduction, such as soft k-means, leads to a new question, which is the focus of this work: *Is using the confidence of the model indeed helpful in transductive inference?*

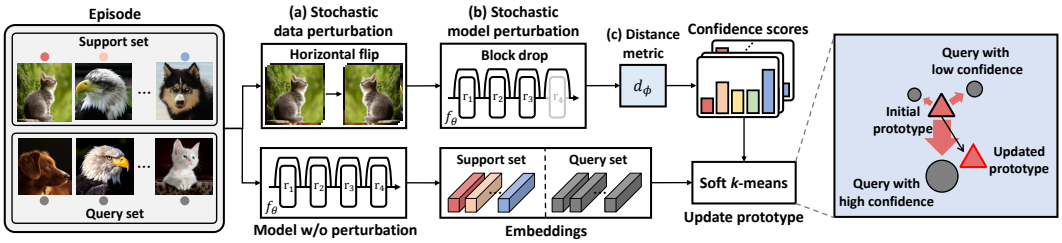


Figure 2: **Overview.** (a) To capture data uncertainty, we randomly apply horizontal flip augmentation to the whole data in episode. (b) Along with data uncertainty, we randomly drop the last residual block to capture the model uncertainty. (c) In order to efficiently train the confidence under these perturbations, we *meta-learn* the input-adaptive distance metric.

4 META-CONFIDENCE TRANSDUCTION

In order to address the question, we propose to *Meta-Confidence Transduction* (MCT). As shown in the method overview in Figure 2, we *meta-learn* the confidence using the various perturbations with input-dependent temperature scaling.

4.1 MODEL AND DATA PERTURBATIONS

The model confidence from few-shot tasks is intrinsically unreliable due to the data scarcity problems, even if the model has been meta-learned over similar tasks. One way to output more reliable confidence scores is to enforce the model to output consistent predictions while perturbing either the model or the data. In this work, we consider the following two sources of perturbations:

Model perturbation: We consider two confidence scores, one from the full network (full-path) and the other from a sub-network generated by dropping a block (drop-path) (Veit et al., 2016; Wu et al., 2018) from the full network. As discussed in Veit et al. (2016), dropping single lower block in ResNet doesn't significantly affect the model performance. Furthermore, we empirically found that block drop allows us to obtain a model that is less correlated to the original model compared to dropout.

Data perturbation: We also consider two confidence scores from two images, one from the original image and the other from horizontally flipped image. This allows us to perturb the data without loss of information, to consistently obtain perturbed confidences at training and test time.

By jointly considering these two sources of perturbations, we can have a total of four (2×2) scenarios (or sources) of possible transductive inferences. As shown in Algorithm 1, at training time, we randomly select a source of confidence and simulate a single transduction step. The reason we optimize only a single full-path is as follows. First, since we randomly apply horizontal flipping to all examples in each episode, perturbed spaces with flipped images are optimized through the sequence of episodes. Secondly, as drop-path is one of the ensemble path of full-path, it is jointly optimized with full-path.

At test time, we perform transductive inference for all scenarios using the ensemble confidence obtained from all perturbed sources. This process is done T times to get the final confidence scores. By doing so, we can enforce the model to perform well under various transduction scenarios with different perturbations, leading to better performance due to the ensemble effect of meta-learned confidences. (See Appendix A for more details of transductive inference).

4.2 META-LEARNING CONFIDENCE WITH INPUT-ADAPTIVE DISTANCE METRIC

In order to enhance the reliability of confidence under various perturbations, we meta-learn the input-adaptive distance metric by performing transductive inference during training with query instances, to obtain a metric that yield performance improvements when performing transductive inference using it. While Liu et al. (2018) proposed to learn input-adaptive length scale metrics for the Gaussian distance kernel to construct nearest neighbor graphs for transductive label propagation, it was aiming

Algorithm 1 Meta-learning confidence with model and data perturbation.

Require: The set of support examples $\mathcal{S}_c$, for each class $c \in \{1, \dots, C\}$.
Require: The set of all query examples $(\tilde{\mathbf{x}}, \tilde{y}) \in \mathcal{Q}$.
Require: Full-path embedding function f_θ and block-dropped embedding function f_θ^D.
Require: Flip augmentation $\mathrm{Aug}(\cdot)$ and define f_θ^A as $f_\theta(\mathrm{Aug}(\cdot))$.

$h_\theta \leftarrow$ Sample from $\{f_\theta, f_\theta^D, f_\theta^A, f_\theta^{A,D}\}$ ▷ Select a confidence space
for $c \in \{1, \dots, C\}$ **do**
  $P'_c \leftarrow \frac{1}{|\mathcal{S}_c|} \sum_{\mathbf{x} \in \mathcal{S}_c} h_\theta(\mathbf{x})$. ▷ Compute prototype on confidence space
for $c \in \{1, \dots, C\}$ **do**
  $q_c(\tilde{\mathbf{x}}) \leftarrow \frac{\exp\left(-d(h_\theta(\tilde{\mathbf{x}}), P'_c)\right)}{\sum_{c'=1}^{C} \exp\left(-d(h_\theta(\tilde{\mathbf{x}}), P'_{c'})\right)}$ for all $\tilde{\mathbf{x}} \in \mathcal{Q}_x$ ▷ Compute confidence score
  $P_c \leftarrow \frac{\sum_{\mathbf{x} \in \mathcal{S}_c} 1 \cdot f_\theta(\mathbf{x}) + \sum_{\tilde{\mathbf{x}} \in \mathcal{Q}_x} q_c(\tilde{\mathbf{x}}) \cdot f_\theta(\tilde{\mathbf{x}})}{\sum_{\mathbf{x} \in \mathcal{S}_c} 1 + \sum_{\tilde{\mathbf{x}} \in \mathcal{Q}_x} q_c(\tilde{\mathbf{x}})}$ ▷ Compute prototype on full-path space
$J \leftarrow 0$ ▷ Initialize loss
for $(\tilde{\mathbf{x}}, \tilde{y}) \in \mathcal{Q}$ **do**
  $J \leftarrow J + \frac{1}{|\mathcal{Q}_x|} \left[d(f_\theta(\tilde{\mathbf{x}}), P_{\tilde{y}}) + \log \sum_{c'} \exp\left(-d(f_\theta(\tilde{\mathbf{x}}), P_{c'})\right) \right]$ ▷ Update loss

to learn the similarity between instances. On the other hand, we meta-learn an input-adaptive metric with perturbations to learn the *confidence* score for explicitly weighting each unlabeled example for transductive inference.

Specifically, we meta-learn the distance metric d_ϕ in Eq. 3 and 4, which we define as Euclidean distance with normalization, instance-wise metric scaling g_ϕ^I, and pair-wise metric scaling g_ϕ^P:

$$d_\phi^I(\mathbf{a}_1, \mathbf{a}_2) = \left\| \frac{\mathbf{a}_1 / \|\mathbf{a}_1\|_2}{g_\phi^I(\mathbf{a}_1)} - \frac{\mathbf{a}_2 / \|\mathbf{a}_2\|_2}{g_\phi^I(\mathbf{a}_2)} \right\|_2^2 \tag{3}$$

$$d_\phi^P(\mathbf{a}_1, \mathbf{a}_2) = \frac{1}{g_\phi^P(\mathbf{a}_1, \mathbf{a}_2)} \left\| \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|_2} - \frac{\mathbf{a}_2}{\|\mathbf{a}_2\|_2} \right\|_2^2 \tag{4}$$

for all $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^l$. The instance-wise metric scaling makes the metric space more flexible, whereas the pair-wise metric scaling additionally adjusts the distance between embeddings, allowing us to obtain adequate confidence. Here, the normalization allows the confidence to be mainly determined by metric scaling so that it is well learned. In order to obtain the optimal scaling function $g_\phi \in \{g_\phi^I, g_\phi^P\}$ for transduction, we first compute the query likelihoods after T transduction steps, and then optimize ϕ, the parameter of the scaling function g_ϕ by minimizing the following episodic-wise loss for $d_\phi \in \{d_\phi^I, d_\phi^P\}$:

$$L^\tau(\theta, \phi) = \frac{1}{|\mathcal{Q}|} \sum_{(\tilde{\mathbf{x}}, \tilde{y}) \in \mathcal{Q}} -\log p(\tilde{y} | \tilde{\mathbf{x}}, \mathcal{S}; \theta, \phi) \tag{5}$$

$$= \frac{1}{|\mathcal{Q}|} \sum_{(\tilde{\mathbf{x}}, \tilde{y}) \in \mathcal{Q}} \left\{ d_\phi(f_\theta(\tilde{\mathbf{x}}), P_c^{(T)}) + \log \sum_{c'=1}^{C} \exp(-d_\phi(f_\theta(\tilde{\mathbf{x}}), P_{c'}^{(T)})) \right\}. \tag{6}$$

As for g_ϕ, we simply use a CNN with fully-connected layers which takes either the feature map of an instance or the concatenated feature map of a pair of instances as an input. We set the number of transduction steps to $T = 1$ for training to minimize the computational cost, but use $T = 10$ for test.

5 Experiments

5.1 Datasets

We validate our method on four benchmark datasets for few-shot classification:

1) miniImageNet. This dataset (Vinyals et al., 2016) consists of a subset of 100 classes sampled from the ImageNet dataset (Russakovsky et al., 2015). Each class has 600 images, resized to 84×84 pixels. We use the split of $64/16/20$ for training/validation/test. **2) tieredImageNet.** This dataset (Ren et al., 2018) is another subset of ImageNet, that consists of $779,165$ images of 84×84 pixels collected from 608 classes. The task is to generalize the few-shot classifier over 34 different superclasses. Thus the entire dataset is split into $20/6/8$ superclasses for training/validation/test, where each superclass contains 351, 97, and 160 low-level classes, respectively. **3) CIFAR-FS.** This dataset (Bertinetto et al., 2019) is a variant of CIFAR-100 dataset used for few-shot classification, which contains 100 classes that describe general object categories. For each class, there are 600 images of 32×32 pixels. The dataset is split into $64/16/20$ classes for training/validation/test. **4) FC100.** This is another few-shot classification dataset (Oreshkin et al., 2018) compiled by reorganizing the CIFAR-100 dataset. The task for this dataset is to generalize across 20 superclasses, as done with the tieredImageNet dataset. The superclasses are divided into $12/4/4$ classes for training/validation/test, each of which contains $60/20/20$ low-level classes, respectively.

5.2 Network architectures

We consider ResNet-12 backbone and conventional 4-block convolutional networks with 64-64-64-64 (ConvNet-64) or 64-96-128-256 (ConvNet-256) channels for each layer. We implement the metric scaling function as a single convolutional block followed by two fully-connected layers (FC-layers). The convolutional block consists of 3x3 convolution, batch normalization, ReLU activation and 2x2 max pooling. The first FC-layer is followed by batch normalization and ReLU activation, whereas the last FC-layer followed by sigmoid function to ensure non-negativity. Finally, in order to balance the effect of the scaling and normalized distance on confidence, we apply scaling ($exp(\alpha)$) and shifting ($exp(\beta)$) to the output of the sigmoid function, where α and β are initialized to 0.

5.3 Experimental setup

Here we mention a few important experimental settings of our model. To avoid overfitting, we apply data augmentation techniques suggested in Cubuk et al. (2019); DeVries & Taylor (2017) and use an auxiliary dense classifier as done in Lifchitz et al. (2019). (See Appendix B for more details of the auxiliary dense classifier).

We use SGD optimizer with the Nesterov momentum of 0.9 and set the weight decay to 0.0005. Following Snell et al. (2017), we use higher way (15-way) classification for training and 5-way for test. The number of query examples for each class is set to 8 for training and 15 for test. For miniImageNet, CIFAR-FS and FC100, we set the initial learning rate to 0.1 and cut it to 0.006 and 0.0012 at $25,000$ and $35,000$ episodes, respectively. For tieredImageNet, we set the initial learning rate to 0.1 and decay it by a factor of 10 at every $20,000$ episode until convergence.

We experiment for semi-supervised few-shot classification as follows. We split both miniImageNet and tieredImageNet into labeled and unlabeled sets, following previous works (Li et al., 2019; Ren et al., 2018). Before we train the model with semi-supervised learning, we pre-train the model with conventional supervised manner (e.g. 64-way classification for miniImageNet). At the meta-training phase, we additionally use 15 instances for each class. At meta-test phase, we use 30 and 50 unlabeled instances for each class on 1-shot and 5-shot task, respectively, following Li et al. (2019). For fair comparison with masked soft k-means of Ren et al. (2018), we use single update step with unlabeled set for both training and testing.

5.4 Main Results

Transductive inference Table 1 and Table 2 show the results of transductive inference with the baselines and our full model, Meta-Confidence Transduction (MCT), which performs transductive inference with the meta-learned confidence. We achieve new **state-of-the-art results** on one-shot classification for all datasets. As for the 5-shot, we achieve comparable performance through a quarter parameter back-

Model	Backbone	miniImageNet	
		1-shot	5-shot
TPN (Liu et al. (2018))	ConvNet-64	55.51±0.86	69.86±0.65
MCT (Instance)	ConvNet-64	**63.53±0.91**	**75.15±0.56**
EGNN (Kim et al. (2019))	ConvNet-256	59.63±0.52	76.34±0.48
MCT (Instance)	ConvNet-256	**70.10±0.87**	**80.56±0.49**

Table 3: Comparison with other transductive models.

Table 1: **Average classification performance** over 1000 randomly generated episodes, with 95% confidence intervals. We consider 5-way classification on all the datasets. * denotes it is reported from Yang et al. (2020).

Model	Backbone	miniImageNet		tieredImageNet	
		1-shot	5-shot	1-shot	5-shot
TPN (Liu et al., 2018)	ConvNet-64	55.51±0.86	69.86±0.65	59.91±0.94	73.30±0.75
EGNN* (Kim et al., 2019)	ConvNet-256	59.63±0.52	76.34±0.48	63.52±0.52	80.24±0.49
MAML+SCA (Antoniou & Storkey, 2019)	DenseNet	62.86±0.79	77.46±1.18	-	-
CAN + Top-k (Hou et al., 2019)	ResNet-12	67.19±0.55	80.64±0.35	73.21±0.58	84.93±0.38
DPGN (Yang et al., 2020)	ResNet-12	67.77±0.32	84.60±0.43	72.45±0.51	87.24±0.39
TEAM (Qiao et al., 2019)	ResNet-18	60.07	75.90	-	-
Fine-tuning (Dhillon et al., 2020)	WRN-28-10	65.73±0.68	78.40±0.52	73.34±0.71	85.50±0.50
SIB (Hu et al., 2020)	WRN-28-10	70.0±0.6	79.2±0.4	-	-
TIM-GD (Boudiaf et al., 2020)	WRN-28-10	77.8	**87.4**	82.1	**89.8**
MCT (Pair)	ResNet-12	76.16±0.89	85.22±0.42	80.68±0.89	86.63±0.89
MCT (Instance)	ResNet-12	**78.55±0.86**	86.03±0.42	**82.32±0.81**	87.36±0.50

Table 2: **Average classification performance** on CIFAR-FS and FC100.

Model	Backbone	CIFAR-FS		FC100	
		1-shot	5-shot	1-shot	5-shot
DPGN (Yang et al., 2020)	ResNet-12	77.90±0.50	90.20±0.40	-	-
Fine-tuning (Dhillon et al., 2020)	WRN-28-10	76.58±0.68	85.79±0.50	43.16±0.59	57.57±0.55
SIB (Hu et al., 2020)	WRN-28-10	80.0±0.6	85.3±0.4	-	-
MCT (Pair)	ResNet-12	**87.28±0.70**	**90.50±0.43**	**51.27±0.80**	62.59±0.60
MCT (Instance)	ResNet-12	85.61±0.69	90.03±0.46	51.16±0.88	**63.28±0.61**

bone network (i.e. ResNet-12 vs WRN-28-10). For fair comparison against TPN (Liu et al., 2018) and EGNN (Kim et al., 2019) that use different backbone networks, we further perform an additional experiments using shallow backbone networks in Table 3. Again, our model largely outperforms all baselines. Here, we use MCT without model perturbation (block drop) since ConvNet-64 and ConvNet-256 do not have skip connections.

Semi-supervised inference We also perform experiments on semi-supervised classification in Table 4 to further validate the effectiveness and generality of our MCT. We follow the same experimental setting described in Li et al. (2019). In the semi-supervised setting, instead of computing the confidence scores of query examples, we compute the confidence scores of unlabeled support examples in order to update the class prototype. Again, our MCT largely outperforms all the baselines including the recent LST model. The results demonstrate the effectiveness of various perturbations with the distance metric scaling for correctly assigning confidence scores to unlabeled examples.

5.5 Ablation Studies

We next perform ablation studies of our model on miniImageNet dataset to identify where the performance improvements come from. We use Prototypical Networks (PN) with ResNet-12 backbone networks for these experiments without auxiliary dense classifier.

Effect of the distance metrics We first study the effect of the distance metric in Table 5. The performance in the transductive inference columns correspond to each of the models with the transductive inference with naive soft k-means algorithm without model and data perturbations. We see that the ProtoNets (PN) with metric scaling from TADAM (Oreshkin et al., 2018) underperforms the plain PN with Euclidean distance. On the other hand, the proposed instance-wise and pair-wise metric significantly outperform both distance metrics in both inductive and transductive inference settings, demonstrating the effectiveness of our input-dependent metric scaling methods over globally shared distance metric. In Figure 3, we observe that instance-wise metric scaling assigns various scales to different inputs. On the other hands, the pair-wise metric scaling assigns low (high) values between the samples from the same (different) classes to allow the model to obtain accurate confidence.

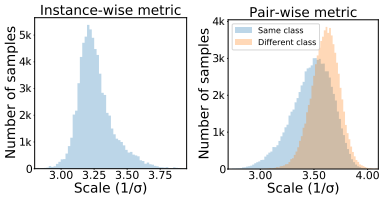


Figure 3: Histogram of metric scale, on a miniImageNet 5-way 5-shot task. σ corresponds to g_ϕ.

Table 4: **Semi-supervised few-shot classification performance**. We consider 5-way classification on miniImageNet ('mini') and tieredImageNet ('tiered'). The baseline results are drawn from Li et al. (2019). All results are based on pre-trained ResNet-12 with full dataset in conventional supervised manner. "w/$\mathcal{D}$" means that unlabeled set includes 3 distracting classes, which does not overlap the label space of the support set (Li et al., 2019; Liu et al., 2018; Ren et al., 2018).

Model	mini		tiered		mini w/$\mathcal{D}$		tiered w/$\mathcal{D}$	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Masked Soft k-Means (Ren et al., 2018)	62.1	73.6	68.6	81.0	61.0	72.0	66.9	80.2
TPN (Liu et al., 2018)	62.7	74.2	72.1	83.3	61.3	72.4	71.5	82.7
LST (Li et al., 2019)	70.1	78.7	**77.7**	85.2	64.1	77.4	73.5	83.4
MCT (Instance)	**73.8**$\pm$**0.7**	**84.4**$\pm$**0.5**	76.9$\pm$0.7	**86.3**$\pm$**0.5**	**69.6**$\pm$**0.7**	**81.3**$\pm$**0.5**	**74.5**$\pm$**0.7**	**84.0**$\pm$**0.5**

Table 5: **Average classification performance** over 1000 randomly generated episodes, with 95% confidence intervals. $d(\cdot, \cdot)$ denotes Euclidean distance. $s \in \mathbb{R}$ is a learnable parameter initialized to 7.5, following Oreshkin et al. (2018).

Model	Distance Metric	Inductive		Transductive	
		1-shot	5-shot	1-shot	5-shot
ProtoNets (PN)	$d(\mathbf{a}_1, \mathbf{a}_2)$	57.36$\pm$0.66	75.59$\pm$0.51	68.58$\pm$0.92	78.71$\pm$0.53
PN + metric scaling	$s \cdot d(\mathbf{a}_1, \mathbf{a}_2)$	55.43$\pm$0.67	74.52$\pm$0.49	68.34$\pm$0.87	78.57$\pm$0.51
PN + Instance-wise metric (Eq. 3)	$d^I_\phi(\mathbf{a}_1, \mathbf{a}_2)$	61.08$\pm$0.66	77.26$\pm$0.46	70.34$\pm$0.87	79.54$\pm$0.54
PN + Pair-wise metric (Eq. 4)	$d^P_\phi(\mathbf{a}_1, \mathbf{a}_2)$	**61.81**$\pm$**0.58**	**77.67**$\pm$**0.50**	**71.95**$\pm$**0.81**	**81.06**$\pm$**0.51**

Effect of the model / data perturbation In Table 6, We analyze the contribution of each type of uncertainty to the reliability of confidence. We observe that the performance of transductive inference improves as we add in each type of uncertainties. We use negative log-likelihood (NLL) as the quality measure for the confidence scores: the lower the NLL, the closer the confidence scores to the target label. We observe that both types of uncertainties are helpful in improving the reliability of the output confidence.

Data Perturb	Model Perturb	miniImageNet 1-shot		miniImageNet 5-shot	
		NLL	Transduction	NLL	Transduction
✗	✗	1.11	71.95$\pm$0.81	0.82	81.06$\pm$0.51
✓	✗	1.09	73.93$\pm$0.85	0.68	81.93$\pm$0.49
✗	✓	**1.04**	74.07$\pm$0.85	**0.60**	82.62$\pm$0.47
✓	✓	1.09	**74.73**$\pm$**0.86**	**0.60**	**83.36**$\pm$**0.45**

Table 6: Test NLL vs. performance of transductive inference with pair-wise distance metric. NLL is computed just before taking the initial transductive step.

Effect of auxiliary dense classifier We use Prototypical Networks with auxiliary dense classifier proposed in Hou et al. (2019) as a baseline. To show how it affects the performance, we do ablation study on miniImageNet. In Table 7, we see that instance-wise distance metric (PN+Inst) improves ProtoNets. In addition, auxiliary dense classifier helps further improve the performance (PN+Inst+AD). Finally, we can see that our full model (PN+AD+MCT), which uses meta-learned confidence for transduction against third row (PN+Inst+AD), achieves the superior performance, improving the performance of the baseline model by 4.72%.

Model	miniImageNet	
	1-shot	5-shot
ProtoNets (PN)	57.36$\pm$0.66	75.59$\pm$0.51
PN+Inst	61.08$\pm$0.66	77.26$\pm$0.46
PN+Inst+AD	65.34$\pm$0.63	82.15$\pm$0.45
PN+AD+MCT(Inst)	**78.55**$\pm$**0.86**	**86.03**$\pm$**0.42**

Table 7: Ablation study on miniImageNet. Inst: Instance-wise distane metric (Eq. 3); AD: Auxiliary dense classifier.

6 Conclusion

Using unlabeled data for few-shot learning, either test instances themselves (transductive) or others (semi-supervised) could help with predictions. Yet, they should be assigned correct confidence scores for optimal performance gains. In this work, we proposed to tackle them by meta-learning confidence scores, such that the prototypes updated with meta-learend scores optimize for the transductive inference performance. Specifically, we first propose perturbations that can simulate model and data-level uncertainties for unseen examples, for more robust confidence estimation. Then, we *meta-learn* the parameter of the length-scaling function on the perturbed samples, such that the proper *distance metric* for the confidence scores can be automatically determined. We experimentally validate our transductive inference model on four benchmark datasets and obtain state-of-the-art performances on both transductive and semi-supervised few-shot classification tasks. Further ablation studies confirm the effectiveness of each component.

REFERENCES

Antreas Antoniou and Amos J Storkey. Learning to learn by self-critique. In *NeurIPS*, pp. 9936–9946, 2019.

Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019.

Malik Boudiaf, Ziko Imtiaz Masud, Jérôme Rony, José Dolz, Pablo Piantanida, and Ismail Ben Ayed. Transductive information maximization for few-shot learning. In *NeurIPS*, 2020.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019.

Debasmit Das and CS George Lee. A two-stage approach to few-shot learning for image recognition. *IEEE Transactions on Image Processing*, 2019.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

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

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pp. 1126–1135. JMLR. org, 2017.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E Turner. Meta-learning probabilistic inference for prediction. In *ICLR*, 2018.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *ICLR*, 2018.

Ruibing Hou, Hong Chang, MA Bingpeng, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *NeurIPS*, pp. 4005–4016, 2019.

Shell Xu Hu, Pablo G Moreno, Yang Xiao, Xi Shen, Guillaume Obozinski, Neil D Lawrence, and Andreas Damianou. Empirical bayes transductive meta-learning with synthetic gradients. In *ICLR*, 2020.

Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *CVPR*, pp. 11–20, 2019.

Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *ICML*, pp. 2933–2942, 2018.

Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In *NeurIPS*, pp. 10276–10286, 2019.

Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. In *CVPR*, pp. 9258–9267, 2019.

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*, 2018.

Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *The IEEE Winter Conference on Applications of Computer Vision*, pp. 2218–2227, 2020.

Donghyun Na, Hae Beom Lee, Hayeon Lee, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. In *ICLR*, 2020.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, pp. 721–731, 2018.

Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *ICCV*, pp. 3603–3612, 2019.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

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.

James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *NeurIPS*, pp. 7957–7968, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *ICJV*, 115(3):211–252, 2015.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, pp. 4077–4087, 2017.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, pp. 403–412, 2019.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, pp. 1199–1208, 2018.

Sebastian Thrun and Lorien Pratt. *Learning to Learn*. Kluwer Academic Publishers, Norwell, MA, USA, 1998. ISBN 0-7923-8047-9.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *NeurIPS*, pp. 550–558, 2016.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, pp. 3630–3638, 2016.

Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S. Davis, Kristen Grauman, and Rogerio Feris. BlockDrop: Dynamic Inference Paths in Residual Networks. In *CVPR*, 2018.

Ling Yang, Liangliang Li, Zilun Zhang, Erjin Zhou, Yu Liu, et al. Dpgn: Distribution propagation graph network for few-shot learning. In *CVPR*, 2020.

Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *ICML*, 2019.

A TRANSDUCTIVE INFERENCE FOR MCT

Algorithm 2 Meta-Confidence Transduction (MCT)

Require: The number of classes C, and the number of transduction steps T.
Require: The set of support examples $\mathcal{S}_c$, for each class $c = 1, \ldots, C$.
Require: The set of all query examples $\mathcal{Q}_x$.
Require: Full-path embedding function f_θ and block-dropped embedding function f_θ^D.
Require: Flip augmentation $\text{Aug}(\cdot)$ and define f_θ^A as $f_\theta(\text{Aug}(\cdot))$.
Require: Embedding function set $F = \{f_\theta, f_\theta^D, f_\theta^A, f_\theta^{A,D}\}$
Output: Confidence score $q_c^T(\tilde{\mathbf{x}})$ obtained after T transduction steps, for all $c = 1, \ldots, C$ and $\tilde{\mathbf{x}} \in \mathcal{Q}_x$.

for $c = 1, \ldots, C$ **do**
  $P_{0,c}^{h_\theta} \leftarrow \frac{1}{|\mathcal{S}_c|} \sum_{\mathbf{x} \in \mathcal{S}_c} h_\theta(\mathbf{x})$ for all $h_\theta \in F$ ▷ Compute initial prototype for each space
for $t = 0, \ldots, T$ **do**
  $q_c^{(t)}(\tilde{\mathbf{x}}) \leftarrow 0$ ▷ Initialize confidence score
  for $c = 1, \ldots, C$ **do**
    for h_θ in F **do**
      $\sigma_{t,c}^{h_\theta}(\tilde{\mathbf{x}}) \leftarrow \frac{\exp(-d_\phi(h_\theta(\tilde{\mathbf{x}}), P_{t,c}^{h_\theta}))}{\sum_{c'}^{C} \exp(-d_\phi(h_\theta(\tilde{\mathbf{x}}), P_{t,c'}^{h_\theta}))}$ for all $\tilde{\mathbf{x}} \in \mathcal{Q}_x$ ▷ Compute local confidence
      $q_c^{(t)}(\tilde{\mathbf{x}}) \leftarrow q_c^{(t)}(\tilde{\mathbf{x}}) + \frac{1}{|F|} \cdot \sigma_{t,c}^{h_\theta}(\tilde{\mathbf{x}})$ for all $\tilde{\mathbf{x}} \in \mathcal{Q}_x$ ▷ Obtain ensemble confidence score
    for h_θ in F **do**
      $P_{t+1,c}^{h_\theta} \leftarrow \frac{\sum_{\mathbf{x} \in \mathcal{S}_c} 1 \cdot h_\theta(\mathbf{x}) + \sum_{\tilde{\mathbf{x}} \in \mathcal{Q}_x} q_c^{(t)}(\tilde{\mathbf{x}}) \cdot h_\theta(\tilde{\mathbf{x}})}{\sum_{\mathbf{x} \in \mathcal{S}_c} 1 + \sum_{\tilde{\mathbf{x}} \in \mathcal{Q}_x} q_c^{(t)}(\tilde{\mathbf{x}})}$
      ▷ Update class c prototype for each space

As shown in Algorithm 2, we update the class prototypes by considering various types of uncertainties. Given an episode consisting of raw images, we generate another episode by flipping the original images. First, prototypes of full-path and drop-path are obtained by averaging embedding of support set. By using these prototypes, we compute the confidence scores for each space and class, respectively. With the ensemble confidence score obtained from various spaces and queries, we update prototypes of each space. Then, we repeatedly update the prototype T times by using an averaged confidence. Finally, $q^{(T)}(\tilde{\mathbf{x}})$ is used for inference.

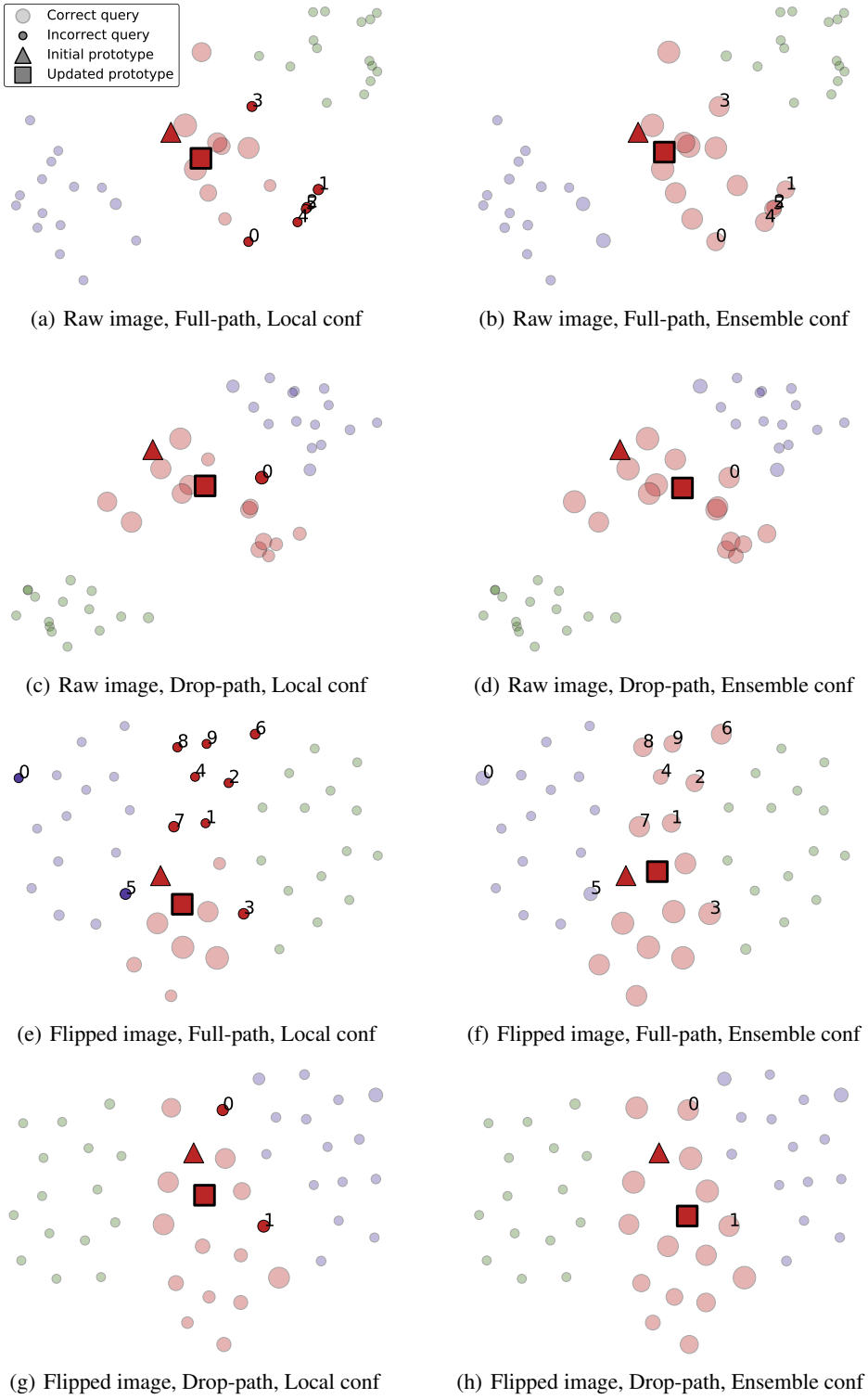
B DETAILED EXPLANATION FOR AUXILIARY DENSE CLASSIFIER

Auxiliary dense classifier (AD) is firstly proposed in Lifchitz et al. (2019), and achieves successful performance improvement in few-shot classification. However, they apply spatial pooling to feature maps, in order to make embeddings at testing. This causes unnecessary bottlenecks, making it difficult to completely use the learned spatial information. To alleviate this problem, we reinterpret AD as a regularizer on the high dimensional embedding being learned. In other words, we do not apply spatial pooling at both training and testing, and then use flattened feature map as the embedding for each instance. We found that computing the distance with densely matching the spatial embeddings improves performance, without any additional parameters. When training with AD, we compute dimension-wise loss L_D^τ, the average classification loss for each dimension of embedding (e.g. 64-way classification for miniImageNet). Hence, final learning objective is $L = E_{p(\tau)}[\lambda L_E^\tau + L_D^\tau]$, where L_I^τ is the episodic-wise loss in Eq. 6 and λ is set to 0.5. For our full models, we evaluate the expectation over task distribution $p(\tau)$ via Monte-Carlo (MC) approximation with a single sample during training to obtain the learning objective.

Model (DC +)	Pooling	miniImageNet 1-shot	miniImageNet 5-shot
Instance-wise metric (d_ϕ^I)	GAP	64.99±0.63	81.22±0.44
Instance-wise metric (d_ϕ^I)	None	**65.34±0.63**	**82.15±0.45**
Pair-wise metric (d_ϕ^P)	GAP	62.66±0.62	80.22±0.47
Pair-wise metric (d_ϕ^P)	None	**64.49±0.64**	**81.63±0.44**

Table 8: The inductive inference performance with various dimension-wise classification methods.

C QUALITATIVE ANALYSIS



(a) Raw image, Full-path, Local conf

(b) Raw image, Full-path, Ensemble conf

(c) Raw image, Drop-path, Local conf

(d) Raw image, Drop-path, Ensemble conf

(e) Flipped image, Full-path, Local conf

(f) Flipped image, Full-path, Ensemble conf

(g) Flipped image, Drop-path, Local conf

(h) Flipped image, Drop-path, Ensemble conf

Figure 4: Visualization of incorrectly classified query examples, on a miniImageNet 3-way 1-shot task. The size of circles shows the confidence score for the red class. Every figure is visualized by same task. conf denotes confidence. In each row, we show the transduction with local confidence and the transduction with ensemble confidence, where local confidence is derived from each space. Best viewed in color.