Transductive Few-shot Learning with Meta-Learned Confidence

Seong Min Kye 1  Hae Beom Lee 2  Hoirin Kim 1  Sung Ju Hwang 2 3 4

Abstract
We propose a novel transductive inference framework for metric-based meta-learning models, which updates the prototype of each class with the confidence-weighted average of all the support and query samples. However, a caveat here is that the model confidence may be unreliable, which could lead to incorrect prediction in the transductive setting. To tackle this issue, we further propose to meta-learn to assign correct confidence scores to unlabeled queries. Specifically, we meta-learn the parameters of the distance-metric, such that the model can improve its transductive inference performance on unseen tasks with the generated confidence scores. We also consider various types of uncertainties to further enhance the reliability of the meta-learned confidence. We combine our transductive meta-learning scheme, Meta-Confidence Transduction (MCT) with a novel dense classifier, Dense Feature Matching Network (DFMN), which performs both instance-level and feature-level classification without global average pooling, and validate it on four benchmark datasets. Our model achieves state-of-the-art results on all datasets, outperforming existing state-of-the-art models by 11.11% and 7.68% on miniImageNet and tieredImageNet dataset respectively. Further qualitative analysis confirms that this impressive performance gain is indeed due to its ability to assign high confidence to instances with the correct labels.

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. 1-shot learning). Some of the metric-based meta-learning approaches tackle this problem using transductive inference, by leveraging the structure of the unlabeled instances at the inference time (Hou et al., 2019; Ren et al., 2018). Popular transductive inference techniques includes leveraging nearest neighbor graph for propagating labels (Kim et al., 2019; Liu et al., 2018), or using predicted labels on unlabeled query samples to update the class prototype (Hou et al., 2019). However, all these transductive 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. Specifically, we update the class prototype by adding a weighted sum of the queries (see Figure 9(a)), where the weights are their predictive confidences (soft labels) that are output from the softmax function. In doing so, instead of selecting top-k examples for each class, we incorporate all of the query examples based on their confidence scores, in order to more robustly update the class prototypes (see Figure 9(b) and Figure 1(c)).

Yet, a crucial problem here is that the predictive confidence scores obtained from a usual inductive training method do not guarantee good performance on transductive inference. Therefore, we further propose to meta-learn the distance metric (see Figure 1(d)), such that the learned confidence could improve the transductive inference performance. Specifically, we simulate a single transduction step for each training episode, and let the the length-scaling function in the distance metric to output the optimal amount of scaling for the given transduction step.

Also, it is important to make the confidence scores robust and reliable. We consider the ensemble of them with various types of uncertainties, both in the model and the data. For example, we drop out the last ResNet block to obtain model uncertainties and perform random horizontal flipping to model data uncertainties. This allows the model to consider various scenarios during meta-learning and thus allows to obtain smoother and more robust confidence scores for
We validate our transductive inference scheme for metric-based meta-learning models against existing transductive approaches, which shows that our model obtains superior performance over them, with significantly higher gains using the meta-learned confidence. Moreover, we propose a novel dense classifier that leverages both the instance and feature-wise classification loss, which alone outperforms the previous state-of-the-art model. Our final model that combines both obtains state-of-the-art results on four benchmark datasets, largely outperforming all baselines. Further qualitative analysis shows that the good performance of our model comes from our model’s ability to obtain a prototype that is closer to the oracle point, such that it assigns high confidence only to well-discriminated samples that can be better trusted.

Our main contributions are as follows:

- We propose a novel transductive inference method for few-shot classification, that takes the confidence-weighted average of all query samples to update the prototype.
- To further enhance the performance of the confidence-based transductive inference, we propose to meta-learn the confidence assignment function such that it helps with transductive inference, by generating various types of uncertainties.
- We propose a new dense classifier for few-shot classification that performs dense feature-matching at test time and is trained with both the instance and pixel-wise classification loss, which alone outperforms the previous state-of-the-art inductive models.
- We validate our model on four benchmark datasets for few-shot classification and achieve new state-of-the-art results, largely outperforming all baselines.

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 (Na et al., 2020; Oreshkin et al., 2018; 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. More recent approaches propose to set the class prototypes to some global parameters (Gidaris & Komodakis, 2018; Qi et al., 2018), such that those global prototypes make the learning more stable than the local prototypes differently computed for each episode.

Transductive few-shot classification Since few-shot classification is intrinsically challenging, we may assume that we can access other unlabeled query examples, which is called transductive inference (Vapnik, 1998). Here we name a few recent works. Liu et al. (2018) construct a nearest-neighbor graph and propagate labels to pseudo-label the unlabeled query examples. 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.

**Dense classification** Dense classification (DC) (Lifchitz et al., 2019) is a pixel-wise classification method that makes the predictions coherent over the spatial dimensions of the last feature map. Lifchitz et al. (2019) train with global prototypes as with Qi et al. (2018) but use local prototypes for testing following Snell et al. (2017). Similarly, Hou et al. (2019) use DC, but with a fully-connected layer over all classes instead of using global prototypes. Our network uses DC with global prototypes as well, but we make the training and testing pipeline consistent by adding additional instance-wise classification loss at training time.

### 3. Method

In this work, we first describe the problem setting, then introduce our novel transductive inference algorithm for metric-based few-shot learning methods, which iteratively updates the class prototypes by taking a weighted sum of the query examples based on their meta-learned confidence score. We also introduce a simple yet effective pixel-wise dense classification loss that is coupled with the conventional instance-wise classification loss, which we empirically found to be superior over existing models.

#### 3.1. Problem Definition

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 \( S = \{ (x_1, y_1), (x_2, y_2), \ldots, (x_{C\times N}, y_{C\times N}) \} \) and query set \( Q = \{ (\tilde{x}_1, \tilde{y}_1), (\tilde{x}_2, \tilde{y}_2), \ldots, (\tilde{x}_{C\times M}, \tilde{y}_{C\times M}) \} \), where \( y, \tilde{y} \in \{1, \ldots, C\} \) are the class labels. 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 \( Q \) given the support set \( S \). Since \( S \) 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. We aim to learn an embedding function \( f_\theta(x) \in \mathbb{R}^l \) that maps an input \( x \) to an l-dimensional metric space. Support set and query examples are then mapped into this space, such that we can measure the distance between class prototypes and query embeddings. The penultimate layer of some convolutional network is usually set to this embedding space.

#### 3.2. Transduction with meta-learned confidence

We propose using the weighted average of all query examples, which is a more generalized transductive inference framework for metric-based models. This provides more robustness to the transductive inference, compared to the top-k selection approach which resorts to only a small fraction of examples.

The key to success is on deciding which query examples to incorporate into the support set \( S \) and how, such that we have a more accurate estimate of the class prototype. Hou et al. (2019) tackle this problem by selecting the top-k queries that are closest to the prototype. However, this model is too sensitive to the choice of the hyperparameter \( K \), since setting a suboptimal \( K \) may result in updating the class prototype with noisy queries that may adversely affect the model performance. Thus, hyperparameter \( K \) should be carefully tuned such that the model includes only the helpful queries into the supports, and exclude the ones that are harmful. Another issue is that all top-k queries are treated equally, without consideration of their relative importance. In the next subsection, we introduce a model that can overcome the limitation of this transductive inference scheme.

#### Transductive inference for metric-based meta-learning

Yet, even with the meta-learning strategy, few-shot learning remains very challenging, and some existing work has proposed to tackle the problem using transductive inference. In transductive inference, when we classify a query example \( \tilde{x}_1 \), we assume that we can access other unlabeled query examples \( \tilde{x}_2, \ldots, \tilde{x}_{C\times M} \), and then make use of the intrinsic structure of the data (e.g. clusters, top-k nearest neighbors, or nearest neighbor graphs) such that the prediction for each instance gets influenced by the prediction on the other instances that are related to it.

How should we set the weights then? One simple way to set the weight, is using the inverse of the distance of the queries to each support. However, when doing multi-class classification with metric-learning models, we should also consider their relative distance to other classes. Thus, instead of using the raw distance, we consider the prediction confidence for each sample as the weight for each query instance, that are output from the model, which lies between the value of 0 and 1.

Toward this goal, we define \( S_c \) as the set of support examples in class \( c \) and \( Q_c = \{ x_1, \ldots, x_{C\times M} \} \) as the set of all query instances. We first compute the initial prototype of each
Transductive Few-shot Learning with Meta-Learned Confidence

class \(c = 1, \ldots, C\):

\[
P_c^0 = \frac{1}{|S_c|} \sum_{x \in S_c} f_{\theta}(x).
\] (1)

Then, for each step \(t = 1, \ldots, T\), and for each query example \(\tilde{x} \in Q_x\), we compute the confidence scores of belonging to each class \(c = 1, \ldots, C\):

\[
q_c^{t-1}(\tilde{x}) = \frac{\exp(-d_{\phi}(f_{\theta}(\tilde{x}), P_{c}^{t-1}))}{\sum_{c' = 1}^{C} \exp(-d_{\phi}(f_{\theta}(\tilde{x}), P_{c'}^{t-1}))}
\] (2)

where \(d_{\phi}\) is some distance metric with learnable parameter \(\phi\). We then update the prototypes of class \(c\) based on the confidence scores \(q_c^{t-1}(\tilde{x})\) for all \(\tilde{x} \in Q_x\):

\[
P_c^t = \frac{\sum_{x \in S_c} 1 \cdot f_{\theta}(x) + \sum_{\tilde{x} \in Q_c} q_c^{t-1}(\tilde{x}) \cdot f_{\theta}(\tilde{x})}{\sum_{x \in S_c} 1 + \sum_{\tilde{x} \in Q_c} q_c^{t-1}(\tilde{x})}
\] (3)

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.

However, the proposed confidence-based transductive inference leads to a couple of new questions, that we address in the next few paragraphs.

- Is using the confidence of the model indeed helpful in transductive inference?
- Can we trust the model confidence that is output from the few-shot task?

Meta-learning confidence Our transductive inference method heavily depends on the confidence assigned to unlabeled queries. To obtain performance improvements from transductive inference, for each class, we want the network to assign high confidence to examples from the same class and low confidence to examples from the other classes. However, modern deep neural networks are known to be miscalibrated (Guo et al., 2017), in that predictive confidence does not match the actual accuracy. Thus, using the model confidence as is may lead to negative impact on the performance of the transductive inference. One way to fix this problem is to calibrate the network, such that the confidence score is well-aligned with the actual accuracy. One of the popular metrics of calibration is Expected Calibration Error (ECE) (Naeini et al., 2015), which divides the confidence values into multiple bins and average the gap between the actual accuracy and the confidence value over all the bins. A popular way to calibrate the model is histogram binning (Zadrozny & Elkan, 2001) and isotonic regression (Zadrozny & Elkan, 2002) that utilizes the cumulative distribution function (CDF), and temperature scaling that learns the temperature parameter of softmax function with a validation set.

Yet, having a well-calibrated model does not guarantee having optimal performance improvement with transductive inference, since even with perfect confidence assignment given to each example, Eq. (3) may not yield an optimal prototype in terms of class discrimination. Therefore, we instead propose a more direct approach; we meta-learn the calibration parameter by actually performing transductive inference during training with query instances, to obtain parameters that yield larger performance improvements on their transductive predictions.

More specifically, we meta-learn the distance metric \(d_{\phi}\) in Eq. (2), that we define as Euclidean distance with normalization and input-dependent length-scales:

\[
d_{\phi}(a_1, a_2) = \frac{\|a_1\|_2 \cdot g_{\phi}(a_1) - \|a_2\|_2 \cdot g_{\phi}(a_2)}{\|a_1\|_2 \cdot g_{\phi}(a_1) + \|a_2\|_2 \cdot g_{\phi}(a_2)}^2
\] (4)

for all \(a_1, a_2 \in \mathbb{R}^l\). In order to make the scaling function \(g_{\phi} : \mathbb{R}^l \rightarrow \mathbb{R}\) optimal for transduction, we first compute the query likelihoods after performing \(T\) transduction steps:

\[
p(\hat{y} = c | \tilde{x}, S; \theta, \phi) = \frac{\exp(-d_{\phi}(f_{\theta}(\tilde{x}), P_{c}^T))}{\sum_{c' = 1}^{C} \exp(-d_{\phi}(f_{\theta}(\tilde{x}), P_{c'}^T))} = q_c^T(\tilde{x}).
\] (5)

And then optimize \(\phi\), the parameter of the scaling function \(g_{\phi}\), by minimizing the following instance-wise loss:

\[
L_I(\theta, \phi) = \frac{1}{|Q|} \sum_{(\tilde{x}, \hat{y}) \in Q} - \log p(\hat{y} | \tilde{x}, S; \theta, \phi).
\] (6)

We empirically found that setting the number of transduction steps to \(T = 1\) works well for training. In other words, the quality of the initial step confidence \(q^0\) is more crucial than the remaining steps for updating class prototypes (See Figure 6(a)). We use \(T = 10\) for testing.

Ensemble confidence with diverse uncertainties The next question in performing confidence-based transductive inference using our framework, is whether we can trust the confidence output from the model. The model confidence from few-shot tasks is intrinsically unreliable due to the data scarcity, even if the model has been meta-learned over similar tasks. One way to output more reliable confidence scores, is to take the average confidence of the model while perturbing either the model or the data. In this work, we consider the following two sources of uncertainties:

- **Model uncertainty**: We consider two confidence scores, one from dropping the final ResNet block and the other from the full network path (Gal & Ghahramani, 2016; Veit et al., 2016; Wu et al., 2018).
- **Data uncertainty**: We also consider two confidence scores, one from horizontal flipping of input images and the other from the original images.
By jointly considering those two sources of randomness, we get a total of four \((2 \times 2)\) confidence scores. We use the average confidence of them for both training and testing. This allows the model to account for diverse uncertainties and help it obtain a more robust estimation of the confidence.

3.3. Dense Feature-Matching Networks (DFMN)

We then impose another constraint using the intuition that all the pixel-wise predictions of \(f_\theta(x)\) should be coherent across the spatial dimensions. We achieve this with dense classification (DC) (Lifchitz et al., 2019), which acts as a regularizer for the high-dimensional space. Specifically, for every pixel \(i \in \{ \{1, 1\}, \ldots, (H, W) \}\) on the spatial grid, we perform classification by indexing \(f_\theta(x)\) and extracting \(K\)-dimensional vector at the location \(i\), which we denote as \(f_\theta^i(x) \in \mathbb{R}^K\). Moreover, following Lifchitz et al. (2019), we assume a set of global prototypes for each class:

\[
\omega = \{ w_c \in \mathbb{R}^K | c = 1, \ldots, C \}
\]

and enforce each pixel vector to correctly cluster around each of the elements in \(\omega\). Specifically, we predict for each spatial location (i.e. pixel) \(i\) as follows:

\[
p^i(\tilde{y}|x; \theta, \omega) = \frac{\exp(-d(f_\theta^i(\tilde{x}), w_\tilde{y}))}{\sum_c \exp(-d(f_\theta^i(\tilde{x}), w_c))} \tag{9}
\]

1While these global prototypes include all the classes in the whole training dataset, for notational brevity we assume that we can select the correct subset for each given episode.

and compute the pixel-wise loss:

\[
L_{\tau}^D(\theta, \omega) = \frac{1}{|Q|} \sum_{(\tilde{x}, \tilde{y}) \in Q} -\log p^i(\tilde{y}|\tilde{x}, S; \theta, \omega). \tag{10}
\]

We set the distance metric \(d\) to Euclidean distance without normalization and re-scaling, following Snell et al. (2017).

**Final training objective and testing** Our final learning objective combines the instance-wise loss in Eq. (7) with the sum of all pixel-wise losses in Eq. (10):

\[
L(\theta, \phi, \omega) = \mathbb{E}_{p(\tau)} \left[ \lambda L_1^I(\theta, \phi) + \sum_i L_{\tau}^D(\theta, \omega) \right] \tag{11}
\]

where \(\lambda\) is a hyperparameter that can be found with a validation set. We evaluate the expectation over task distribution \(p(\tau)\) via Monte-Carlo (MC) approximation with a single sample during training. This final objective allows us to perform instance-wise classification in a high-dimensional space, while each of the pixels remains coherent across the spatial dimensions. Note that the pixel-wise prediction in Eq. (9) is used for training, but not for test (See Figure 3). At test time, we use the instance-wise prediction in Eq. (6) after repeating the full \(T = 10\) transduction steps.

4. Experiments

**Dataset** 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 \(84 \times 84\) 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 super classes. 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.

Experimental setting Here we mention a few important experimental settings of our model. During training, we apply weight decay of 0.0005 and perform random cropping and horizontal flipping of the images for data augmentation, following the standard procedure. When the image size is 32 × 32, we apply max-pooling only to the second and the fourth layer to increase the dimensionality of the final embedding space. Note that the number of additional parameters incurred by our DFMN is very marginal. For all experiments, we set λ = 0.2 in Eq. (11) which we found with a validation set. Please see the supplementary file for more details (e.g. the number of training epochs and learning rate scheduling). We will also submit the codes2 for reproduction of the experimental results.

4.1 Main Results

Inductive inference We first examine the results of inductive inference to analyze the advantage of using our classifier. The top rows of Table 1 and Table 2 show the results of our Dense Feature-Matching Networks (DFMN) and existing inductive methods for few-shot classification. Our model achieves new state-of-the-art results on all four benchmark datasets with significant margins. We observe that the models with dense classification generally perform better than models without, demonstrating the importance of coherent predictions across the spatial dimensions of the embedding space. Further, DFMN significantly outperforms existing dense classifiers, such as DC (Lifchitz et al., 2019) and CAN (Hou et al., 2019). This performance gain is due to two reasons. First, our model computes the distance using dense matching of spatial features, while existing models use global average pooling before computing distances which results in the loss of spatial information. Secondly, existing dense classifiers train pixel-wise classifiers during training and instance-wisely predict at test time, which results in inconsistency across the two phases. DFMN on the other hand, has a consistent framework as it

2https://github.com/seongmin-kye/MCT_DFMN
Transductive Few-shot Learning with Meta-Learned Confidence

Table 3. Comparison with other transductive models on same backbone networks for fair comparison.

| Model                  | Transduction | Backbone | miniImageNet 1-shot | 5-shot | tieredImageNet 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 |
| DFMN + MCT             | ✓            | ConvNet-64 | 64.65 ± 0.89 | 75.96 ± 0.54 | 63.85 ± 0.95 | 75.72 ± 0.61 |
| EGNN (Kim et al., 2019)| ×            | ConvNet-256 | 58.57 ± 0.64 | 75.25 ± 0.47 | 57.56 ± 0.72 | 77.09 ± 0.57 |
| DFMN                  | ×            | ConvNet-256 | 69.73 ± 0.93 | 80.49 ± 0.50 | 70.21 ± 0.97 | 80.22 ± 0.56 |
| EGNN (Kim et al., 2019)| ✓            | ConvNet-256 | 76.37           |          | 80.15                 |        |

Figure 4. Visualization of incorrectly classified query examples, on a miniImageNet 5-way 1-shot task. Best viewed in color.

Transductive inference Now we evaluate the performance of our full model, which performs transductive inference with meta-learned confidence. The bottom rows of Table 1 and Table 2 show the results of our models against baselines with transductive inference; we again achieve new state-of-the-art results on all the datasets. Notably, DFMN with our Meta-Confidence Transduction (MCT) largely outperforms the same model with the top-k transductive inference proposed in Hou et al. (2019) (denoted as Top-k) (10.20% on miniImageNet and 8.00% on tieredImageNet), which is the second-best performing model. MCT yields relatively larger performance gains on one-shot cases. We next compare MCT against the recent nearest neighbor graph approaches, TPN (Liu et al., 2018) and EGNN (Kim et al., 2019) with the same backbone network architectures used in their experiments, for fair comparison. Table 3 shows the results of this experiment. MCT significantly outperforms both baselines using the same backbone networks, which demonstrates the effectiveness of our transduction method.

Qualitative Analysis Figure 4 visualizes the effect of transductive inference using top-k method introduced in CAN (Hou et al., 2019) and our MCT. We observe that if the initial prototype is far from the class center, the top-k strategy is likely to add incorrect queries from other classes to the support set for each class, causing the prototypes to move to an incorrect direction. For example, in Figure 4(a), the initial prototype of the green class is far from the oracle prototype, and thus it cannot escape from the initial point. However, our MCT in Figure 4(b) relocates the prototype close to the oracle prototype, greatly reducing the number of incorrect predictions of the green class. This explains why our model has significantly larger performance gains on 1-shot cases where the initial prototypes are highly biased.

4.2. Ablation Study

We next perform ablation studies to better understand the effect of the different key components of our model. We experiment with miniImageNet and use ResNet-12 backbone.

Effect of distance metric We first study the effect of the distance metric in Table 4. The first row shows the results using Euclidean distance with normalization. We see that the confidence is initially too underconfident and the expected calibration error (ECE) is very high (39% for 1-shot and 55% for 5-shot), such that the gain using the transductive inference is marginal (1.87% for 1-shot) or even negative (-0.65% for 5-shot). This is because the normalization makes the space too compact. In the second and third row, we see that lowering the entropy by learning a scalar parameter s or setting it to a fixed constant of 10 improves the transductive inference by sharpening the confidence scores (See Figure 1(c) and 1(d)). Finally, the proposed meta-learning of the scaling function in Eq. (12) obtains the highest accuracy improvements among all distance measures, since its input-wise scaling provides more flexibility than other temperature-scaling methods. Note that the correlation between ECE and the gain from transductive inference is less clear. This suggests that the calibration performance itself is not a good measure of transductive inference performance.

Effect of diverse uncertainties We next analyze the contribution of each type of uncertainty, on obtaining more reliable confidence scores for query examples, in Table 5. We observe that the transductive performance improves as we add in each of the two types of uncertainties. Note that entropy increases as well, which is opposite to Table 12, because overconfident scores are penalized as we average confidence scores from different sources of uncertainties. Thus the effect of uncertainties on confidence is different from that of the distance metric. Without the uncertainties, the confidence is determined solely by the distance, in which case the model cannot assign high weights to true queries far from the class prototypes. However, by averaging confidence with various uncertainties, we can upweight the queries that are far from the prototype (See Figure 5).
Table 4. Ablation study on distance metric. We do not consider uncertainties here for clear comparison. ECE and mean Entropy of confidence scores are computed just before taking the initial transduction step. Red color: overconfident and blue color: underconfident. Inductive*: the results are from inductive inference with the transductively trained model. d(\cdot, \cdot) denotes euclidean distance, and we let \( n := a/\|a\|_2 \). s \( \in \mathbb{R} \) is a learnable parameter initialized to 10, following Gidaris & Komodakis (2018) and Lifchitz et al. (2019).

| Distance Metric | Entropy | ECE | Inductive* | Transductive | Gain |
|-----------------|---------|-----|------------|--------------|------|
| \( d(A_1, A_2) \) | 1.586   | 9.6 | 65.43±0.63 | 74.30±0.81  | 8.73 |
| \( s \cdot d(A_1, A_2) \) | 0.732   | 6.2 | 65.57±0.63 | 74.30±0.81  | 8.73 |
| 10 \( - d(A_1, A_2) \) | 0.586   | 11.1| 65.92±0.65 | 74.11±0.85  | 8.19 |
| \( d_2(a_1, a_2) \) (12) | 0.466   | 17.9| 64.03±0.63 | 74.78±0.84  | 10.75|

Table 5. Ablation study on the source of uncertainties.

| Flip Image | Drop Layer | minImageNet 1-shot | Entropy | ECE | Inductive* | Transductive | Gain |
|------------|------------|---------------------|---------|-----|------------|--------------|------|
| \( \times \times \) | 0.466   | 17.9 | 64.03±0.63 | 74.78±0.84  | 10.75 |
| \( \checkmark \times \) | 0.494   | 14.9 | 65.76±0.64 | 76.75±0.86  | 10.99 |
| \( \times \checkmark \) | 0.924   | 2.4  | 65.33±0.63 | 77.98±0.78  | 12.65 |
| \( \checkmark \checkmark \) | 0.972   | 2.7  | 65.54±0.64 | 78.30±0.81  | 12.76 |

Figure 5. Confidence with/without uncertainties. Numbers indicate the queries incorrectly classified by (a) but correctly by (b).

The number of transduction steps We next see the effect of the number of transduction steps used for training and testing. Figure 6(a) suggests that we can simply set the number of transduction steps for training to \( T = 1 \), considering computational efficiency as well. It implies that this very first step is important for finding the correct direction of updating the class prototypes. Figure 6(b) shows that the test performance keep increasing monotonically as we repeat the transduction steps up to around \( T = 10 \). We found those configurations with a validation set.

Classification method Lastly, we perform an ablation study of different types of classifiers in Table 6. Global Prototypes (G) denotes whether we perform classification with global prototypes. Dense classification (D) denotes whether we apply pixel-wise classification in Eq. (10). We observe that the combination of those two (G and D) leads to significant improvements (Lifchitz et al., 2019). Next, Instance-wise classification (I) denotes whether we use dense feature matching (Figure 2(b)) instead of global average pooling (Figure 2(a)), and additionally perform instance-wise classification in Eq. (7) on that high-dimensional space. Normalization (N) denotes whether we apply normalization on the distance metric in Eq. (12). We see that whereas instance-wise classification (I) alone deteriorates the performance due to the excessive dimensionality, normalization (N) solves the problem and largely improves the classification performance. Further, increase in the dimensionality \((512\rightarrow 18432)\) of the features considered for distance computation results in marginal increase of inference time (+0.6 ms). Overall, the Table 6 justifies our design of combining two different classifiers in Eq. (11).

Table 6. Ablation study on classification method.

| G | D | I | N | Dim. | IT | minImageNet (Inductive) |
|---|---|---|---|-----|----|-------------------------|
| \( \times \times \times \times \) | 512 | 2.2 | 55.41±0.65 | 74.63±0.49 |
| \( \checkmark \times \times \times \times \) | 512 | 2.2 | 52.81±0.64 | 76.71±0.47 |
| \( \checkmark \times \times \times \times \) | 512 | 2.2 | 58.96±0.63 | 79.89±0.46 |
| \( \checkmark \times \times \times \times \) | 18432 | 2.8 | 56.49±0.63 | 73.51±0.52 |
| \( \checkmark \times \times \times \checkmark \checkmark \) | 18432 | 2.8 | 65.78±0.63 | 82.05±0.44 |

5. Conclusion

We proposed a novel transductive inference scheme for metric-based few-shot learning models. Our method updates the class prototype by taking a weighted combination of queries, whose weights are determined by their predictive confidence. In order to directly make the confidence scores to improve the transductive inference, we propose to \( meta-learn \) the parameter of the length-scaling function such that proper distance metric for the confidence scores be automatically determined. We also consider various types of uncertainties to obtain more robust confidence estimation for unseen examples. We experimentally validate our transductive inference model on four benchmark datasets and obtain state-of-the-art performances, by coupling it with a
novel dense classifier. Further analysis confirms that the impressive performance of our model owes to its ability to correctly upweight or downweight samples, that allows it to obtain prototypes that are closer to oracles.

References

Bertinetto, L., Henriques, J. F., Torr, P., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. In ICLR, 2019.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In ICML, pp. 1126–1135. JMLR.org, 2017.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In ICML, 2016.

Gidaris, S. and Komodakis, N. Dynamic few-shot visual learning without forgetting. In CVPR, 2018.

Gordon, J., Bronskill, J., Bauer, M., Nowozin, S., and Turner, R. E. Meta-learning probabilistic inference for prediction. In ICLR, 2018.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. In ICLR, 2018.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In ICML, pp. 1321–1330, 2017.

Hou, R., Chang, H., Bingpeng, M., Shan, S., and Chen, X. Cross attention network for few-shot classification. In NeurIPS, pp. 4005–4016, 2019.

Kim, J., Kim, T., Kim, S., and Yoo, C. D. Edge-labeling graph neural network for few-shot learning. In CVPR, pp. 11–20, 2019.

Lee, K., Maji, S., Ravichandran, A., and Soatto, S. Meta-learning with differentiable convex optimization. In CVPR, pp. 10657–10665, 2019.

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

Li, X., Sun, Q., Liu, Y., Zhou, Q., Zheng, S., Chua, T.-S., and Schiele, B. Learning to self-train for semi-supervised few-shot classification. In NeurIPS, pp. 10276–10286, 2019.

Lifchitz, Y., Avrithis, Y., Picard, S., and Bursuc, A. Dense classification and implanting for few-shot learning. In CVPR, pp. 9258–9267, 2019.

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. Sphereface: Deep hypersphere embedding for face recognition. In CVPR, pp. 212–220, 2017.

Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S. J., and Yang, Y. Learning to propagate labels: Transductive propagation network for few-shot learning. In ICLR, 2018.

Na, D., Lee, H. B., Lee, H., Kim, S., Park, M., Yang, E., and Hwang, S. J. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. In ICLR, 2020.

Naeini, M. P., Cooper, G. F., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In AAAI, 2015.

Oreshkin, B., López, P. R., and Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. In NeurIPS, pp. 721–731, 2018.

Qi, H., Brown, M., and Lowe, D. G. Low-shot learning with imprinted weights. In CVPR, pp. 5822–5830, 2018.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In ICLR, 2017.

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. In ICLR, 2018.

Requeima, J., Gordon, J., Bronskill, J., Nowozin, S., and Turner, R. E. Fast and flexible multi-task classification using conditional neural adaptive processes. In NeurIPS, pp. 7957–7968, 2019.

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascaru, R., Osindero, S., and Hadsell, R. Meta-learning with latent embedding optimization. In ICLR, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. IJCV, 115(3):211–252, 2015.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In NeurIPS, pp. 4077–4087, 2017.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In CVPR, pp. 1199–1208, 2018.

Thrun, S. and Pratt, L. Learning to Learn. Kluwer Academic Publishers, Norwell, MA, USA, 1998. ISBN 0-7923-8047-9.
Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P.-A., and Larochelle, H. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.

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

Veit, A., Wilber, M. J., and Belongie, S. Residual networks behave like ensembles of relatively shallow networks. In *NeurIPS*, pp. 550–558, 2016.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *NeurIPS*, pp. 3630–3638, 2016.

Wu, Z., Nagarajan, T., Kumar, A., Rennie, S., Davis, L. S., Grauman, K., and Feris, R. BlockDrop: Dynamic Inference Paths in Residual Networks. In *CVPR*, 2018.

Yoon, S. W., Seo, J., and Moon, J. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. *arXiv preprint arXiv:1905.06549*, 2019.

Zadrozny, B. and Elkan, C. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, 2001.

Zadrozny, B. and Elkan, C. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, 2002.

Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., and Whiteson, S. Fast context adaptation via meta-learning. In *ICML*, 2019.
Supplementary File for
Transductive Few-shot Learning with Meta-Learned Confidence

A. Experimental Setup

Here we describe the detailed experimental setup. We consider ResNet-12 backbone and conventional 4-block convolutional networks with 64-64-64-64 or 64-96-128-256 channels for each layer. We apply dropout to each layer with the ratio of 0.1. We use SGD optimizer with the Nesterov momentum of 0.9 and set the weight decay to 0.0005. Following Snell et al. (2017); Yoon et al. (2019), we use higher way (15-way) for training and 5-way for test. The number of query examples for each class is set to 8 for training and 15 for testing. 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 each episode, we use the global prototypes of the whole classes in the dataset for training. 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 use global prototypes only for the classes corresponding to each episode in order to reduce memory consumption.

B. Architecture of Length-scale Function

Figure 7 shows the specific network architecture which implements the length-scale function $g_\phi$ we introduced in the main paper. It is basically a CNN with one convolutional block followed by one fully-connected layer. The convolutional block consists of $3 \times 3$ convolution, batch normalization, and ReLU activation. The final fully connected layer is followed by SoftPlus activation, which enforces the length scale $g_\phi(a)$ to be non-negative.

![Figure 7. Length scale function](image)

C. More Ablation Study

C.1. Comparison with DC with shallow network

We further observe in Table 7 that while existing dense classifier often underperforms prototypical networks with shallow backbone networks due to excessive spatial coherency (Lifchitz et al., 2019), our DFMN outperforms both Prototypical Networks and dense classifiers, since it utilizes both instance-wise and pixel-wise predictions. More specifically, we not only apply dense classification for spatial coherency, but also eliminate the global average pooling to make use of all the spatial dimensions (instance-wise classification), effectively removing unnecessary bottlenecks.

| Model          | Dim. | miniImageNet 1-shot | miniImageNet 5-shot |
|----------------|------|---------------------|---------------------|
| PN (Snell et al., 2017) | 1600 | 45.33 ± 0.65        | 69.37 ± 0.53        |
| DC (Lifchitz et al., 2019) | 64   | 49.98 ± 0.67        | 67.80 ± 0.57        |
| DFMN           | 1600 | **52.73 ± 0.65**    | **70.89 ± 0.51**    |

*Table 7. The effectiveness of DFMN on a shallow network (ConvNet-64). All the results are re-implemented by us.*
C.2. Effect of uncertainties with shallow ConvNet

In Table 8 and 9, we observe that ensemble model with layer-dropout obtains worse performance with shallow networks (ConvNet). This is because each layer of ConvNet has significant impact on accuracy for shallow networks, unlike deep networks with shortcut connections (e.g. ResNet-12). However, the ensemble with horizontal flipping achieves consistent performance gains on those shallow networks compared to the one without uncertainties. However, for the ConvNet-256, the effect of the uncertainties seems marginal.

D. Detail of Transductive Inference

As shown in Algorithm 1, 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 queries, we update prototypes of each space and class respectively. With the confidence scores obtained from various space and queries, we update prototypes of each example. Moreover, we find that true queries far from the confidence of the embeddings depend on the source of uncertainty. We notice that ensemble of multiple techniques that generate uncertainties can yield more accurate confidence for each example. Furthermore, we find that true queries far from the prototypes can be upweighted by ensemble of various uncertainties. This explains why our MCT with diverse uncertainties yields more accurate prototypes.

### Table 8. Ablation study on the source of uncertainties for miniImageNet.

| Drop | Flip | Backbone | miniImageNet 1-shot | miniImageNet 5-shot |
|------|------|----------|---------------------|---------------------|
|      |      |          | Inductive* Transductive Gain | Inductive* Transductive Gain |
| ×    | ×    | ConvNet-64 | 54.00±0.65 | 63.10±0.93 | 9.10 | 71.83±0.51 | 74.94±0.57 | 3.11 |
| ✔    | ×    | ConvNet-64 | 52.53±0.64 | 60.52±0.93 | 7.99 | 70.43±0.54 | 73.43±0.59 | 3.00 |
| ×    | ✔    | ConvNet-64 | 54.90±0.62 | 64.65±0.89 | 9.75 | 72.73±0.49 | 75.96±0.54 | 3.23 |
| ×    | ×    | ConvNet-256 | 56.98±0.60 | 69.35±0.85 | 12.37 | 75.35±0.47 | 79.70±0.50 | 4.35 |
| ✔    | ×    | ConvNet-256 | 54.95±0.62 | 66.74±0.86 | 11.79 | 74.24±0.50 | 78.17±0.53 | 3.93 |
| ×    | ✔    | ConvNet-256 | 57.30±0.60 | 69.73±0.93 | 12.43 | 75.81±0.49 | 80.49±0.50 | 4.68 |

### Table 9. Ablation study on the source of uncertainties for tieredImageNet.

| Drop | Flip | Backbone | tieredImageNet 1-shot | tieredImageNet 5-shot |
|------|------|----------|------------------------|------------------------|
|      |      |          | Inductive* Transductive Gain | Inductive* Transductive Gain |
| ×    | ×    | ConvNet-64 | 55.62±0.68 | 65.12±0.98 | 9.50 | 73.03±0.38 | 75.47±0.60 | 2.44 |
| ✔    | ×    | ConvNet-64 | 53.44±0.69 | 64.50±0.98 | 11.06 | 72.62±0.59 | 75.03±0.63 | 2.41 |
| ×    | ✔    | ConvNet-64 | 55.83±0.73 | 65.66±0.98 | 9.83 | 73.34±0.60 | 75.72±0.61 | 2.38 |
| ×    | ×    | ConvNet-256 | 57.29±0.71 | 70.34±0.96 | 13.05 | 76.66±0.55 | 80.01±0.57 | 3.35 |
| ✔    | ×    | ConvNet-256 | 57.56±0.70 | 69.71±0.90 | 12.15 | 75.91±0.56 | 78.48±0.61 | 2.57 |
| ×    | ✔    | ConvNet-256 | 59.13±0.70 | 70.21±0.97 | 11.08 | 77.01±0.54 | 80.22±0.56 | 3.21 |

E. Qualitative Analysis

Figure 8 visualizes the example embeddings in the space of the last path, along with their confidence. The figure shows that the confidence of the embeddings depend on the source of uncertainty. We notice that ensemble of multiple techniques that generate uncertainties can yield more accurate confidence for each example. Moreover, we find that true queries far from the prototypes can be upweighted by ensemble of various uncertainties. This explains why our MCT with diverse uncertainties yields more accurate prototypes.
Algorithm 1 Meta-Confidence Transduction (MCT)

**Input:** The number of classes \( C \), and the number of transduction steps \( T \).
**Input:** The set of support examples \( S_c \), for each class \( c = 1, \ldots, C \).
**Input:** The set of all query examples \( Q_x \).
**Input:** Full-path embedding function \( f_\theta \) and drop-layer embedding function \( f_\theta^{\text{drop}} \).
**Input:** Horizontal-flipping operation \( \text{Flip()} \).

**Output:** Confidence score \( q_c^x(\tilde{x}) \) obtained after \( T \) transduction steps, for all \( c = 1, \ldots, C \) and \( \tilde{x} \in Q_x \).

\[
\begin{align*}
1: & \quad \text{for } c = 1, \ldots, C \text{ do} \\
2: & \quad \text{Compute initial prototype: } P_{0,c} \leftarrow \frac{1}{|S_c|} \sum_{x \in S_c} f_\theta(x). \\
3: & \quad \text{Repeat with drop-layer (D) embedding function } f_\theta^{\text{drop}} \text{ and flipping (F) function } \text{Flip()} \text{ in turn to collect } P_{0,c}, P_{0,c}^{D,F}, P_{0,c}^{D,F}. \\
4: & \quad \text{for } c = 1, \ldots, C \text{ do} \\
5: & \quad \text{Compute initial confidence score: } \sigma_{0,c} \leftarrow \frac{\exp(-d_\theta(f_\theta(\tilde{x}), P_{0,c}))}{\sum_{c'} \exp(-d_\theta(f_\theta(\tilde{x}), P_{0,c}'))}. \\
6: & \quad \text{Repeat with } \{P_{0,c}, P_{0,c}^{D,F}\}, \{P_{0,c}^{D,F}\}, \{P_{0,c}^{D,F}\} \text{ to collect } \sigma_{0,c}, \sigma_{0,c}^{D,F}, \sigma_{0,c}^{D,F}. \\
7: & \quad q_c^x(\tilde{x}) \leftarrow \frac{1}{2} (\sigma_{0,c} + \sigma_{0,c}^{D,F} + \sigma_{0,c}^{D,F}). \\
8: & \quad \text{for } t = 1, \ldots, T \text{ do} \\
9: & \quad \text{Update class } c \text{ prototype: } P_{t,c} \leftarrow \frac{\sum_{x \in S_c} \|f_\theta(x)\| + \sum_{\tilde{x} \in Q_x} q_{t-1}^x f_\theta(\tilde{x})}{\sum_{x \in S_c} \|f_\theta(x)\| + \sum_{\tilde{x} \in Q_x} q_{t-1}^x}. \\
10: & \quad \text{Repeat with } f_\theta^{\text{drop}} \text{ and } \text{Flip()} \text{ in turn to collect } \{P_{t,c}, P_{t,c}^{D,F}, P_{t,c}^{D,F}\}. \\
11: & \quad \text{for } c = 1, \ldots, C \text{ do} \\
12: & \quad \text{Compute } t\text{-step confidence score: } \sigma_{t,c} \leftarrow \frac{\exp(-d_\theta(f_\theta(\tilde{x}), P_{t,c}))}{\sum_{c'} \exp(-d_\theta(f_\theta(\tilde{x}), P_{t,c}'))}. \\
13: & \quad \text{Repeat with } \{P_{t,c}, P_{t,c}^{D,F}\}, \{P_{t,c}^{D,F}\}, \{P_{t,c}^{D,F}\} \text{ to collect } \sigma_{t,c}, \sigma_{t,c}^{D,F}, \sigma_{t,c}^{D,F}. \\
14: & \quad q_c^x(\tilde{x}) \leftarrow \frac{1}{2} (\sigma_{t,c} + \sigma_{t,c}^{D,F} + \sigma_{t,c}^{D,F}) \text{ for all } \tilde{x} \in Q_x.
\end{align*}
\]

Figure 8. Visualization of incorrectly classified query examples, on a miniImageNet 3-way 1-shot task. The size of circles shows the confidence scores for the red class. Each row is visualized by same episode. Best viewed in color.
Figure 9. Reliability plots. For (a) and (b), c=1, s=10, c=10 and $d_\phi$ denotes $d(\hat{a}_1, \hat{a}_2)$, $s \cdot d(\hat{a}_1, \hat{a}_2)$, $10 \cdot d(\hat{a}_1, \hat{a}_2)$ and $d_\phi(a_1, a_2)$ respectively. If plot is above(under) dotted line that denotes perfectly calibrated one, it means underconfident(overconfident). All results are conducted on ResNet-12.