A Dual Attention Network with Semantic Embedding for Few-shot Learning

Shipeng Yan*, Songyang Zhang*, Xuming He†
School of Information Science and Technology, ShanghaiTech University
{yanshp, zhangsy1, hexm}@shanghaitech.edu.cn

Abstract

Despite recent success of deep neural networks, it remains challenging to efficiently learn new visual concepts from limited training data. To address this problem, a prevailing strategy is to build a meta-learner that learns prior knowledge on learning from a small set of annotated data. However, most of existing meta-learning approaches rely on a global representation of images and a meta-learner with complex model structures, which are sensitive to background clutter and difficult to interpret. We propose a novel meta-learning method for few-shot classification based on two simple attention mechanisms: one is a spatial attention to localize relevant object regions and the other is a task attention to select similar training data for label prediction. We implement our method via a dual-attention network and design a semantic-aware meta-learning loss to train the meta-learner network in an end-to-end manner. We validate our model on three few-shot image classification datasets with extensive ablative study, and our approach shows competitive performances over these datasets with fewer parameters.

1. Introduction

A particular intriguing property of human cognition is being able to learn a new concept from only a few examples, which, despite recent success of deep learning, remains a challenging task for machine learning systems [14]. Such a few-shot learning problem setting has attracted much attention recently, and in particular, for the task of classification [13, 33, 31]. To tackle the issue of data deficiency, a prevailing strategy of few-shot classification is to formulate it as a meta-learning problem, aiming to learn a prior on the few-shot classifiers from a set of similar classification tasks [33, 17]. Typically, a meta-learner learns an embedding that maps the input into a feature space and a predictor that transfers the label information from the training set of each task to its test instance.

While this learning framework is capable of extracting effective meta-level prediction strategy, it suffers several limitations in the task of image classification. First, the i.i.d assumption on tasks tends to ignore the semantic relations between image classes that reflects the intrinsic similarity between individual tasks. This can lead to inefficient embedding feature learning. Second, most of existing work rely on an off-the-shelf deep network to compute a holistic feature of each input image, which is sensitive to nuisance variations, e.g, background clutter. This makes it challenging to learn an effective meta-learner, particularly for the methods based on feature similarity. Moreover, recent attempts typically resort to learning complex prediction strategies to incorporate the context of training set in each task [23, 17], which are difficult to interpret in terms of the prior knowledge that has been learned.

In this work, we aim to address the aforementioned weaknesses by a semantic-aware meta-learning framework, in which we explicitly incorporates class sharing across tasks and focuses on only semantically informative parts of input images in each task. To this end, we make use of attention mechanisms [32] to develop a novel modularized deep network for the problem of few-shot classification. Our deep network consists of two main modules: an embedding network that computes a semantic-aware feature map for each image, and an meta-learning network that learns a similarity-based classification strategy to transfer the training label cues to a test example.

Specifically, given a few-shot classification task, our embedding network first generates a convolutional feature map for each image. Taking as input all these feature maps, the meta-learning network then extracts a task-specific representation of input data with a dual-attention mechanism, which is used for few-shot class prediction. To achieve this, the meta-learning network first infers a spatial attention map for each image to capture relevant regions on the feature maps and produces a selectively pooled feature vector for every image [34]. Given these image features, the network employs a second attention module, referred as task attention, to compute an attention map over the training set of the task. This attention encodes the relevance of each

*Authors contributed equally and are listed in alphabetical order
†In part supported by the NSFC Grant No. 61703195.

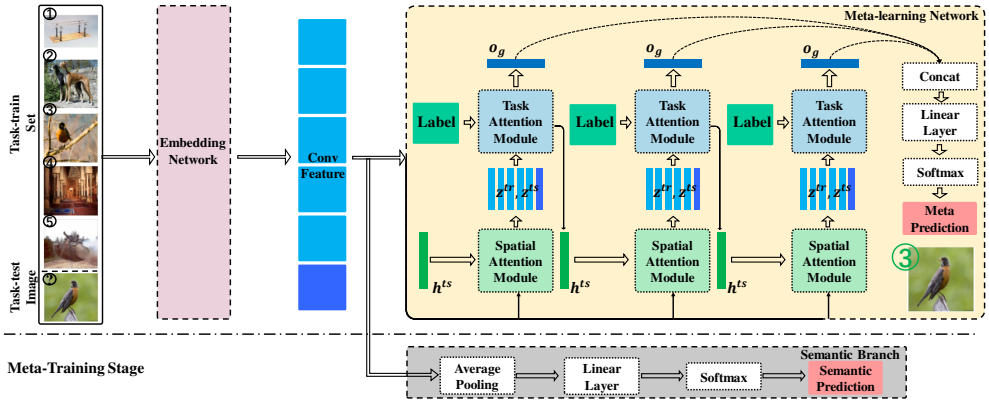

Figure 1: An illustration of few-shot classification via our attention-based network. See text for more details.

training example to the test image class in the task and is used to calculate a context-aware representation of the test instance [33] for its class prediction. To improve its discriminative power, we can further refine the context-aware representation by stacking multiple layers of two attention modules and the resulting deep network is referred as the Spatial-Task Attention Network (STANet). Figure 1 shows an overview of few-shot classification via our dual attention network.

For training our STANet, we design a multi-task loss to incorporate the shared class semantics of all tasks and to learn a meta-level classification strategy. To this end, we introduce a semantic branch and integrate a semantic prediction loss for the embedding network with a meta-classification loss for the overall network.

We evaluate our STANet extensively on three challenging few-shot image classification benchmarks, including the Omniglot dataset, MiniImageNet and a new dataset based on the CIFAR-100. The empirical results and ablative study demonstrate the superior or comparable performance of our method over the prior state-of-the-art approaches. The main contributions of our work are two-fold:

- We develop an efficient attention-based deep network for one-shot prediction, which is also easy to interpret in terms of learned knowledge for task-level generalization. Our network achieves the state-of-the-art accuracies with much fewer parameters and simpler network structure.

- We propose to learn a semantic-aware representation for few-shot classification, which exploits the label correlation across tasks and location of objects. Moreover, we build a new benchmark of few-shot classification based on CIFAR-100 to study the impact of task similarity and benefits of shared representations.

2. Related Work

Few-shot learning: Inspired by data-efficient learning in human cognition [12], few-shot learning aims to learn a new concept representation from only a few training examples. Such a learning paradigm has attracted much attention in the literature [5, 33, 21] as the traditional data-driven deep learning approaches, despite their recent success, have difficulty in handling new classes with limited data annotation [15]. Existing few-shot learning approaches can be largely categorized into three main groups: Bayesian learning based, metric learning based and meta-learner based methods.

Early works on few-shot learning aim to build a Bayesian prior model that can be transferred to new classes. Fei-Fei et al. [4, 5] utilized a hierarchical Bayesian model to represent the prior knowledge on visual classes for one-shot image classification. More recently, Lake et al. [13] proposed a hierarchical Bayesian program learning (HBPL) to effectively learn the prior knowledge on object categories.

A second strategy in few-shot learning learns to predict class-agnostic similarity between data instances. In particular, deep siamese network [9] trains a convolutional network to embed data samples so that samples in the same class are close while samples in different classes are far away. Recent works [33, 26, 27, 31] refine this idea by introducing recurrent network structure, attention mechanisms, or novel learning objective to improve the similarity learning. Sung et al. [29] propose to use relation networks to compare the images within episodes. However, these methods typically rely on a global feature representation of images and thus lack the capacity to choose relevant regions when embedding the images. In contrast, our approach employs dual attention mechanism to focus on object features in images.

Meta-learning, or learning-to-learn strategies [19, 30, 25], have been applied to few-shot learning and made significant progresses recently [1, 21, 6, 18, 16]. Ravi and

Larochelle [21] proposed an LSTM meta-learner to learn the exact optimization algorithm used to train the neural network classifier in the few-shot regime. MAML [6] learns an update step that a learner can take to successfully adapt to a new task. Other work [3] combines metric learning and meta-learning to learn task-specific learners.

In addition to the meta-optimizers, other complex deep network models have been adopted as meta-learners for few-shot learning, such as memory augmented neural networks [23], graph neural networks [24] and Meta networks [18], which encode a meta-level inductive biases across tasks. Temporal convolution network [17] models each classification task as a sequence prediction problem. Recently, [20] propose to generate the parameters of the last network layer from the activations of a pre-trained feature embedding. [7] use an attention kernel to produce a mixing of pre-trained linear weights for novel categories. In contrast to these models, our method is based on a simple attention-based neural network that has a compact structure and is easy to interpret in terms of learned prior knowledge.

Attention-based representation: Attention mechanism enables a deep network to attend relevant parts of input data, which is important for learning an object representation robust toward cluttered background. Additive attention [2, 34] and multiplicative attention [32, 28] are the two most commonly used attention mechanism. We exploit the self-attention [32] to capture the data similarity in our method. The Matching Network [33] also uses an attention mechanism over a learned embedding of training examples to predict classes for the test data. By contrast, our dual attention network further incorporates the spatial attention [34] to learn a better representation of input images.

3. Problem Setting and Overview

We aim to learn an image classification model that can predict object classes using only a few annotated images per class as training data. To this end, we formulate the few-shot image classification as a meta-learning problem [33]. Formally, we consider each instance of few-shot classification as a task T (also called *an episode*) sampled from a task distribution $\mathcal{T}$. Each task T is defined by a class label set $\mathbf{L}_T$, a task-train set $\mathbf{D}_T^{tr}$ (also called *support set*) consisting of N annotated images, and a task-test example $\mathbf{x}_T^{ts}$ with its groundtruth class label $y_T^{ts} \in \mathbf{L}_T$. The task-train set $\mathbf{D}_T^{tr} = \{(\mathbf{x}_T^{(1)}, y_T^{(1)}), \cdots, (\mathbf{x}_T^{(N)}, y_T^{(N)})\}$, where $\mathbf{x}_T^{(i)}$ denotes the i-th input image in task T, and $y_T^{(i)} \in \mathbf{L}_T$ is its class labels, $1 \leq i \leq N$.

The problem of meta-learning is to build a meta-learner $\mathcal{M}$, or a mapping from the task-train set $\mathbf{D}_T^{tr}$ and the task-test data $\mathbf{x}_T^{ts}$ to the task-test label y_T^{ts}. Its learning framework typically consists of two phases: meta-training and meta-test. In the meta-training phase, we train the meta-learner $\mathcal{M}$ on a set of tasks $\{T = (\mathbf{L}_T, \mathbf{D}_T^{tr}, \mathbf{x}_T^{ts}, y_T^{ts})\}$ sampled from $\mathcal{T}$, denoted as $\mathcal{S}_{tr}^{meta}$. The entire label set used in meta-training is denoted by $\mathcal{L}_{tr} = \cup_{T \in \mathcal{S}_{tr}^{meta}} \mathbf{L}_T$. In the meta-test phase, we evaluate the meta-learner by testing it on a separate set of task $\mathcal{S}_{ts}^{meta}$ with new classes only. In other words, let the label set in the meta-test be $\mathcal{L}_{ts} = \cup_{T \in \mathcal{S}_{ts}^{meta}} \mathbf{L}_T$, and we have $\mathcal{L}_{tr} \cap \mathcal{L}_{ts} = \emptyset$. We use the one-shot learning setting throughout the model sections for notation clarity.

In this work, we address the meta-learning problem in the context of image classification by explicitly incorporating spatial and semantic cues of object categories and develop an easy-to-interpret deep network architecture for few-shot classification. Our approach is motivated by three key observations: 1) Object categories are mostly localized in the images and using only relevant features allows us to learn an object representation robust toward background clutters; 2) A simple attention mechanism can be used to find semantically similar images and encode the context of task-train set $\mathbf{D}_T^{tr}$ for label prediction; and 3) A good image representation is critical for building an effective meta-learner and can be learned by incorporating the semantic class information across the individual tasks (i.e., $\mathcal{L}_{tr}$) in the meta-learning setting. We instantiate these ideas by designing a deep dual-attention neural network for the few-shot image classification problem, which is detailed in the following section.

4. Model Architecture

We now introduce our deep neural network based meta-learner, which learns a class-relevant feature representation of images based on a spatial attention and a context-aware representation of test instances using a task attention. To effectively train the dual-attention network, we also propose a meta-learning loss with novel semantic regularization.

Specifically, our deep network consists of two main network modules: an **embedding network** module that computes convolutional feature maps for all the images in the input $(\mathbf{D}_T^{tr}, \mathbf{x}_T^{ts})$ and a **meta-learning network** that uses a dual spatial-task attention mechanism to predict the task-test label y_T^{ts}. To facilitate the network learning, we also introduce an auxiliary **semantic branch** in the meta-training stage. We refer to our deep meta-learner as the Spatial-Task Attention Network (STANet). An overview of our entire model pipeline with two attention layers is shown in Figure 1 and we will describe each module in details below.

4.1. Embedding network

Given a task (or episode), the first stage of our STANet is an embedding network module that extracts convolutional feature maps of every training and test image in this task. The embedding module consists of a series of convolution

layers with residual connections and multiple strides [8]. Unlike prior work, we do not collapse the image feature maps into a feature vector with full-connected layers. By maintaining the spatial structure of the feature maps, we are able to select regions relevant to the image categories, and ignore the background clutters in the next stage.

Formally, denoting the embedding network as $\mathcal{F}$ and a task $T = (\mathbf{L}, \mathbf{D}^{tr}, \mathbf{x}^{ts}, y^{ts})$, we compute the feature maps of images in T as,

$$\mathbf{C}_i^{tr} = \mathcal{F}(\mathbf{x}^{(i)}), \quad \mathbf{x}^{(i)} \in \mathbf{D}^{tr},\ 1 \le i \le N, \tag{1}$$

$$\mathbf{C}^{ts} = \mathcal{F}(\mathbf{x}^{ts}) \tag{2}$$

where $\mathbf{C}_i^{tr}$ and $\mathbf{C}^{ts}$ are feature representations of task-train and task-test images respectively. Here we omit the task index T for clarity. Let (H_f, W_f) be the height and width of the feature maps. We represent the features as a matrix in $\mathbb{R}^{n_{ch} \times n_{loc}}$, where n_{ch} is the number of feature channels and $n_{loc} = H_f \times W_f$.

4.2. Meta-learning network

Taking as input the conv features and task-train labels $(\mathbf{D}^{tr}, \mathbf{x}^{ts})$, the second component of the STANet is a meta-learning network that builds a classifier to predict the class label y^{ts}. To achieve this, we introduce a dual-attention mechanism to locate the relevant image regions and produce a context-aware representation of task-test image for transferring task-train labels.

We implement the dual-attention mechanism as a spatial-task attention (STA) layer, composed of a spatial attention and a task attention module. The STA layer takes the image features and task-train labels in T to produce a context-aware representation of the test example $\mathbf{x}^{ts}$, which also allows us to stack multiple STA layers to generate a set of refined task-test image features for classification. Below we will introduce those two attention modules and an one-layer STA network first, followed by the multi-layer STA network.

4.2.1 Spatial attention module

To focus on the regions related to their semantic class, we introduce a spatial attention module to generate object-centric representations of images in a task by exploiting spatial information in the conv features. Specifically, we derive a spatial attention map $\mathbf{a}_s$ for each conv feature map $\mathbf{C} \in \{\mathbf{C}_i^{tr}\}_{i=1}^N \cup \{\mathbf{C}^{ts}\}$ based on the task-test feature, which will be detailed below. Here the attention $\mathbf{a}_s \in \Delta^{n_{loc}}, \Delta^{n_{loc}} = \{\mathbf{a}_s \in \mathbb{R}^{n_{loc}}, \mathbf{a}_s \succeq \mathbf{0}, \mathbf{1}^\intercal \mathbf{a}_s = 1\}$, indicating the relevance of each spatial site w.r.t. the target class y^{ts}. Given the attention map $\mathbf{a}_s$, we can take a weighted average of the feature map to obtain an object-centric representation $\mathbf{z} = \mathbf{a}_s^\intercal \mathbf{C}$, where $\mathbf{z} \in \mathbb{R}^{n_{ch}}$.

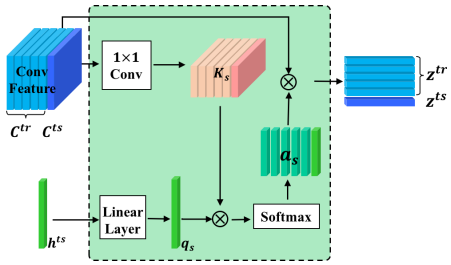

Figure 2: Spatial attention module of meta-learning branch, which generates object-aware representations of images.

To compute the attention maps, we first estimate a task-test representation $\mathbf{h}^{ts} \in \mathbb{R}^{d_h}$ that captures distinctive features of the test class of the task. We initialize $\mathbf{h}^{ts}$ by taking an average pooling of the test feature map $\mathbf{C}^{ts}$, which typically generates a global semantic feature descriptor for the test image. We use the task-test representation $\mathbf{h}^{ts}$ as a query and search for the relevant spatial sites on the conv feature maps. Formally, we adopt the attention mechanism proposed in [32], which maps the query feature ($\mathbf{h}^{ts}$) and the conv features ($\{\mathbf{C}_i^{tr}\}_{i=1}^N \cup \{\mathbf{C}^{ts}\}$) into a key space, and measures the key similarities based on their inner product. To this end, we apply an 1×1 convolution to each conv feature map $\mathbf{C}$ to compute its key representation $\mathbf{K}_s$, and a linear transform to the query $\mathbf{h}^{ts}$ to compute its key $\mathbf{q}_s$:

$$\mathbf{q}_s = \mathbf{W}_{\mathbf{q}_s} \mathbf{h}^{ts}, \quad \mathbf{K}_s = \mathbf{W}_{\mathbf{K}_s} \mathbf{C} \tag{3}$$

where $\mathbf{W}_{\mathbf{q}_s} \in \mathbb{R}^{d_{k_s} \times d_h}$, $\mathbf{W}_{\mathbf{K}_s} \in \mathbb{R}^{d_{k_s} \times n_{ch}}$, and the dimensions of the resulting keys are $\mathbf{q}_s \in \mathbb{R}^{d_{k_s}}$ and $\mathbf{K}_s \in \mathbb{R}^{d_{k_s} \times n^{loc}}$. The spatial attention map on $\mathbf{C}$ is then derived by a weighted inner product between the query and the conv feature keys, followed by a softmax function:

$$\mathbf{a}_s = \text{softmax}\left(\frac{\mathbf{q}_s \mathbf{K}_s}{\sqrt{d_{k_s}}}\right) \tag{4}$$

Such an attention map will have larger weights on the locations sharing similar features as the task-test representation.

For each task, the spatial attention module generates an attention map for every task-train and task-test image, and we denote them as $\{\mathbf{a}_s^{(i)}\}_{i=1}^N$ and $\mathbf{a}_s^{ts}$ respectively. Given those attention maps, we compute an object-aware feature representation for each image as follows:

$$\mathbf{z}^{ts} = \mathbf{a}_s^{ts\intercal} \mathbf{C}^{ts}, \quad \mathbf{z}_i^{tr} = \mathbf{a}_s^{(i)\intercal} \mathbf{C}_i^{tr},\ 1 \le i \le N, \tag{5}$$

where $\mathbf{z}^{ts}, \mathbf{z}_i^{tr}$'s $\in \mathbb{R}^{n_{ch}}$. Figure 2 shows the structure of the spatial attention module.

4.2.2 Task attention module

Given the object-centric image features $\mathbf{z}^{ts}, \{\mathbf{z}_i^{tr}\}_{i=1}^N$ of a task, the second module of the meta-learning network aims

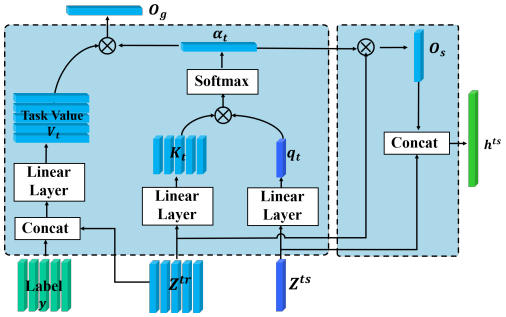

Figure 3: Task attention module produces context-aware representations for few-shot classification via task attention.

to produce a context-aware test feature for predicting the test class y^{ts}. Our goal is to find the task-train instances that are semantically similar to the test one for label transfer. To this end, we implement a task attention mechanism inspired by [17] to compute the context-aware test feature.

Concretely, we use the task-test image feature $\mathbf{z}^{ts}$ as a query and produce a task attention vector $\mathbf{a}_t \in \Delta^N$ (here Δ denotes a simplex). Each element of the task attention vector encodes the similarity between the task-test feature and the corresponding training feature. We adopt a similar strategy as in the spatial attention module: First, we compute the key representations as $\mathbf{q}_t = \mathbf{W}_{\mathbf{q}_t}\mathbf{z}_{ts}$, $\mathbf{K}_t = \mathbf{W}_{\mathbf{K}_t}\left[\mathbf{z}_1^{tr}, \mathbf{z}_2^{tr}, \cdots, \mathbf{z}_N^{tr}\right]$, where $\mathbf{W}_{\mathbf{q}_t} \in \mathbb{R}^{d_{k_t} \times n_{ch}}$, $\mathbf{W}_{\mathbf{K}_t} \in \mathbb{R}^{d_{k_t} \times n_{ch}}$, and the dimensions of the resulting keys are $\mathbf{q}_t \in \mathbb{R}^{d_{k_t}}$ and $\mathbf{K}_t \in \mathbb{R}^{d_{k_t} \times N}$. The task attention on the task-train set is then calculated by $\mathbf{a}_t = \text{softmax}\left(\frac{\mathbf{q}_t\mathbf{K}_t}{\sqrt{d_{k_t}}}\right)$.

Given the task attention, we compute a context-aware representation to encode the task-train examples that are similar to the test instance, which is then used to predict the test label. Specifically, we compute a linear embedding of the training features and labels, and take the weighted average of the embedded features based on the task attention $\mathbf{a}_t$:

$$\mathbf{V}_t = \mathbf{W}_{\mathbf{v}_t}\left[\begin{pmatrix}\mathbf{z}_1^{tr}\\ \mathbf{y}_1\end{pmatrix}, \begin{pmatrix}\mathbf{z}_2^{tr}\\ \mathbf{y}_2\end{pmatrix}, \cdots, \begin{pmatrix}\mathbf{z}_N^{tr}\\ \mathbf{y}_N\end{pmatrix}\right] \quad (6)$$

$$\mathbf{o}_g = \mathbf{a}_t^\intercal \mathbf{V}_t, \quad (7)$$

where $\mathbf{y}_i$ are the one-hot encoding of label $y^{(i)}$ for a given task, and $y^{(i)} \in \{1, \ldots, |\mathbf{L}|\}$. The embedding matrix $\mathbf{W}_{\mathbf{v}_t} \in \mathbb{R}^{d_{v_t} \times (n_{ch}+|\mathbf{L}|)}$, the embedded feature $\mathbf{V}_t \in \mathbb{R}^{d_{v_t} \times N}$, and the context-aware representation $\mathbf{o}_g \in \mathbb{R}^{d_{v_t}}$. Here $\mathbf{o}_g$ can be viewed as a summary of the task-train information relevant for predicting the test class in the task. The computation flow of this module is illustrated in Figure 3.

Based on the context-aware feature, we predict the label of the task-test instance by a fully connected layer followed by a softmax function as follows,

$$P_{meta} = \text{softmax}(\mathbf{W}_o\mathbf{o}_g) \quad (8)$$

where $P_{meta} \in \Delta^{|\mathbf{L}|}$ is the probability vector over the label space of the task, $\mathbf{W}_o \in \mathbb{R}^{|\mathbf{L}| \times d_{v_t}}$ is the weight for the fully connected layer, and $\hat{y}^{ts}_{meta} = \text{argmax}_{l \in \{1,\ldots,|\mathbf{L}|\}} P_{meta}(l)$ is the predicted label.

4.2.3 Multi-layer STA Net

Our one-layer STA network relies on an estimated task-test representation $\mathbf{h}^{ts}$ to initialize the first spatial attention module and hence the efficacy of our dual attention mechanism depends on the quality of $\mathbf{h}^{ts}$. While the average pooling provides a good initial estimate, our task context feature $\mathbf{o}_g$ can be further improved given a better task-test representation. Specifically, we re-estimate the task-test representation $\mathbf{h}^{ts}$ at the end of the task attention module based on the features $\mathbf{z}^{ts}$ and an attention-weighted average of $\{\mathbf{z}_i^{tr}\}_{i=1}^N$:

$$\mathbf{o}_s = \mathbf{a}_t^\intercal\left[\mathbf{z}_1^{tr}, \mathbf{z}_2^{tr}, \ldots, \mathbf{z}_N^{tr}\right], \quad \mathbf{h}^{ts} = [\mathbf{z}^{ts\intercal}, \mathbf{o}_s^\intercal]^\intercal \quad (9)$$

where $\mathbf{o}_s \in \mathbb{R}^{n_{ch}}$ encodes the task context and is used to enrich the test feature $\mathbf{z}^{ts}$.

Using the new $\mathbf{h}^{ts}$, we stack a second STA-layer into the meta-learning module and generate a new task context presentation $\mathbf{o}_g$. Such process can be repeated and produce M outputs $\{\mathbf{o}_g^{(0)}, \mathbf{o}_g^{(1)}, \cdots, \mathbf{o}_g^{(M-1)}\}$ with an M-layer STA network. We concatenate $\mathbf{o}_g^{(m)}$s from all the STA layers to form a multi-level task-context representation, which is then pass through a logistic regressor to predict the final label.

4.3. Meta-training with semantic regularization

To estimate the model parameters, we train our STANet in the meta-learning framework. Specifically, we assume a meta-train dataset $\mathcal{S}_{tr}^{meta} = \{T = (\mathbf{L}_T, \mathbf{D}_T^{tr}, \mathbf{x}_T^{ts}, y_T^{ts})\}$ is provided in the meta-train stage, which is sampled from the task distribution $\mathcal{T}$. Our goal is to minimize the expected task loss by learning an image representation through the embedding network and a spatial-task attention mechanism through the meta-learning network. To this end, we propose a novel meta-learning loss that consists of the empirical task loss on the meta-train dataset and a semantic loss that exploits the class correlation between different tasks.

The empirical task loss is the average log-loss of the network predictions on the task-test instances, defined as,

$$L_{task}(\Theta) = \sum_{T \in \mathcal{S}_{tr}^{meta}} \frac{-\log P_{meta}(y_T^{ts}|\mathbf{D}_T^{tr}, \mathbf{x}_T^{ts}; \Theta)}{|\mathcal{S}_{tr}^{meta}|} \quad (10)$$

where Θ denotes all the model parameters.

To learn a better embedded feature representation, we further introduce a semantic loss defined on a shared class space across different tasks. To achieve this, we augment the embedding network with a Semantic Branch that predicts a label distribution in the global label space $\mathcal{L}_{tr}$. This allows us to inject additional supervisory signal into the embedding network training. Specifically, our semantic branch takes average pooling on each convolutional feature map $\mathbf{C}$ and passes the resulting features through a logistic regressor, which predicts a label distribution $P_{sem} \in \Delta^{|\mathcal{L}_{tr}|}$over the global label set,

$$P_{sem} = \text{softmax}(\mathbf{W}_{sem}\mathbf{1}^{\intercal}\mathbf{C}) \tag{11}$$

where $\mathbf{W}_{sem}$ is the weight for the logistic regression. The semantic loss is defined by the average log-loss of the semantic branch prediction:

$$L_{sem}(\Theta) = \sum_{T \in \mathcal{S}_{tr}^{meta}} \sum_{(\mathbf{x}_T, y_T)} \frac{-\log P_{sem}(y_T|\mathbf{x}_T;\Theta)}{|\mathcal{S}_{tr}^{meta}|(|\mathbf{D}^{tr}|+1)} \tag{12}$$

where the dataset size equals to the number of task-train data $|D_T^{tr}|$ plus the number of task-test examples. And the overall meta-train loss is defined as,

$$L_{full} = L_{task}(\Theta) + \lambda L_{sem}(\Theta) \tag{13}$$

where λ is a weight balancing the regularization from the semantic loss. As the semantic loss is closer to the embedding network and shared across different training tasks, it enables us to significantly speedup the feature learning, and learn a better convolutional feature representation for the meta-learning network.

5. Experiments

We evaluate our STANet method on the task of few-shot image classification by conducting a set of experiments on three datasets. In addition to two publicly-available datasets, MiniImageNet [11] and Omniglot [13], we propose a new few-shot learning benchmark using real-world images from CIFAR100 [10], which is referred to as Meta-CIFAR100 dataset. In this section, we introduce the datasets and report detailed experimental results. We perform different N-way m-shot experiments on the three datasets, with 95% confidence interval in the meta-test phase[1].

5.1. MiniImageNet

Dataset. MiniImageNet is a subset of the ILSVRC-12 dataset [22], consisting of 84×84 RGB images from 100 different classes with 600 examples per class. We adopted the splits proposed by [33, 21] with 64 classes for training, 16 for validation, 20 for testing in the meta-learning setting.

[1]Details of network architecture and experiments configuration are listed in the supplementary material.

Table 1: MiniImageNet Performance. STANet-S refers to shallow embedding network.

Method	# Params	Feature Extractor	5-way Accuracy	
			1-shot	5-shot
Matching Net [33]	0.1M	Conv64	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
Prototypical Net(Snell et al. 2017)	0.1M	Conv64	$49.42 \pm 0.78\%$	$68.20 \pm 0.66\%$
MAML (Finn et al. 2017)	0.1M	Conv64	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
RelationNet[29]	0.23M	Conv64	$50.44 \pm 0.82\%$	$65.32 \pm 0.70\%$
(Gidaris et al. 2018)	0.24M	Conv64	$56.20 \pm 0.86\%$	$72.81 \pm 0.62\%$
GNN (Satorras et al. 2018)	1.6M	Conv64	$50.33 \pm 0.36\%$	$66.41 \pm 0.63\%$
STANet-S(1-Layer)	0.24M	Conv64	$50.38 \pm 0.65\%$	$65.67 \pm 0.66\%$
STANet-S(3-Layer)	0.24M	Conv64	$\mathbf{53.11 \pm 0.60\%}$	$\mathbf{67.16 \pm 0.66\%}$
SNAIL [17]	6.1M	ResNet-12	$55.71 \pm 0.99\%$	$68.88 \pm 0.92\%$
(Gidaris et al. 2018)	2.6M	ResNet-12	$55.45 \pm 0.86\%$	$70.13 \pm 0.68\%$
[20]	40.5M	WRN-28	$\mathbf{59.60 \pm 0.41\%}$	$\mathbf{73.74 \pm 0.19\%}$
STANet(1-Layer)	2.6M	ResNet-12	$57.25 \pm 0.40\%$	$69.45 \pm 0.50\%$
STANet(3-Layer)	2.6M	ResNet-12	$\mathbf{58.35 \pm 0.57\%}$	$\mathbf{71.07 \pm 0.39\%}$

Table 2: Ablation study for STANet on MiniImageNet using 3 layers dual-attention. SR-Semantic Regularization, SA-Spatial Attention, TA-Task Attention.

Components			5-way(Normal)	
SR.	SA.	TA.	1-shot	5-shot
✗	Uniform	✓	$53.41 \pm 0.61\%$	$64.32 \pm 0.57\%$
✗	Gaussian	✓	$54.29 \pm 0.66\%$	$65.41 \pm 0.55\%$
✗	✓	✓	$55.52 \pm 0.64\%$	$66.75 \pm 0.62\%$
✓	✓	✓	$\mathbf{58.35 \pm 0.57\%}$	$\mathbf{71.07 \pm 0.39\%}$

Quantitative Results. We compare the performance of our STANet with previous state-of-the-art meta-learning methods in Table 1. The top section compares our networks with other methods using the same shallow embedding network, while the bottom section shows comparison results with deeper embedding networks. In both settings, our 3-Layer STANet outperforms the previous approaches that use the same type of embedding networks by a sizable margin. Moreover, our network achieves comparable accuracies to [20]'s method but has a much simpler architecture: only 6% of their model in parameter size.

Visualizing Results. To understand the dual attention mechanism, we visualize the spatial attentions of the 3-Layer STANet by overlaying them on the images and the task attentions in Figure 4. We can see that spatial attention helps the model locate salient region of task-test image (e.g., the foreground objects), and the matching regions in the task-train set. Based on the localized features, the task attention weight for each task-train image indicates how relevant an image is for predicting the label of the task-test image. Moreover, Figure 4 shows that spatial attention generates sharper focuses on salient regions with increasing number of layers, while task attention also concentrates more on the task-train image in the same class as the task-test ones.

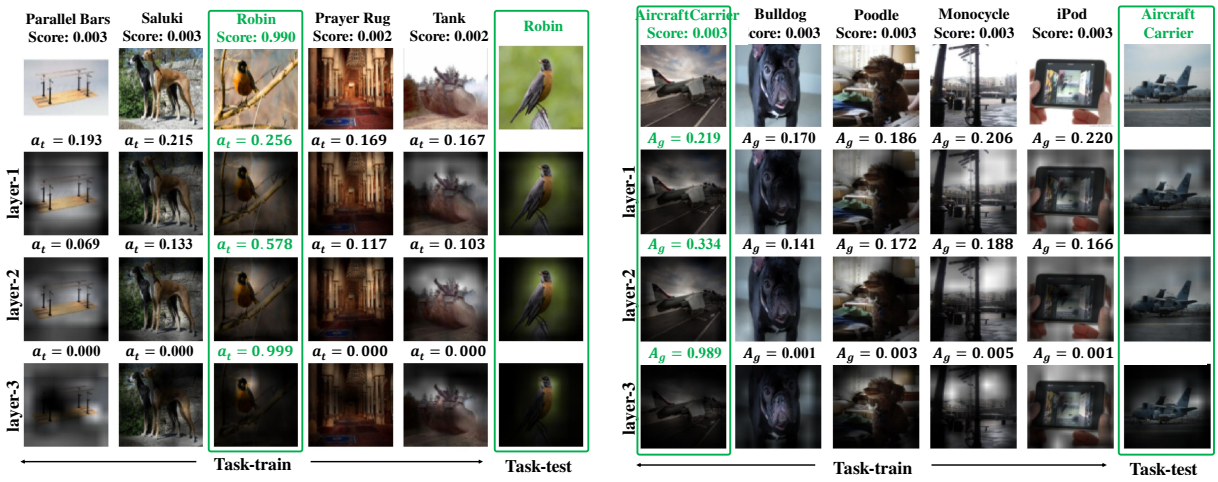

Figure 4: Visualization of attention maps in the 3-Layer STANet on one-shot classification (MiniImageNet). Green indicates the correct class for the test image. First row: input images, classes and predicted scores. Second-Fourth row: spatial attention-masked images. $\mathbf{a}_t$ denotes the task attention values.

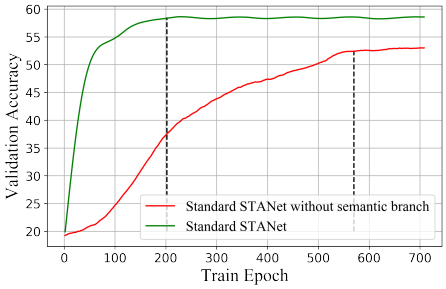

Figure 5: Validation curves of the STANet with and without the semantic branch. Green curve shows the training of the full STA-Net and Red curve is for the STA-Net without semantic branch.

Ablation Study. We conduct a series of ablation studies to evaluate the importance of each component used in our STANet model. Table 1 and Table 2 summarize the results of our ablative experiments, in which we compare our full model with several partial model settings.

First, we compare the single-layer STANet with the multi-layer model in Table 1 (last two rows). We can see that the multi-layer STANet further promotes the accuracies in both few-shot settings. While the improvement seems mild, it demonstrates that more STA layers can refine the image representations to achieve better performance.

Second, in Table 2, we create three baseline models by removing the semantic branch or the spatial attention. By comparing with the full model, we show that adding the semantic branch improves the performance of the model from 55.52% to 58.35% with a 3% gain. We also compare our spatial attention with uniform and Gaussian attention, and our learned attention achieves favorable performance. We note that the MiniImageNet has a strong center-bias, which may cause the mild improvements.

Finally, we compare the training process of the full STANet model with the STANet without semantic regularization. We plot the validation performance curves of these two STANets during training in Figure 5. It is evident that the semantic branch is able to improve the convergence and the final performance significantly, which indicates that the full STANet exploits the semantic information efficiently.

5.2. Meta-CIFAR100

Dataset. To investigate the impact of task distributions in the few-shot learning, we design a new few-shot classification benchmark, Meta-CIFAR100, based on the CIFAR-100[10] dataset. We use all the classes from the CIFAR-100 in our dataset, which contains 32×32 RGB images from 100 classes with 600 images per class. The label classes of CIFAR100 have a balanced hierarchical structure: they are included in 20 parent categories and each parent category comprises 5 base categories. This allows us to design different types of task distributions when building training and test splits in the meta-learning setting.

Specifically, we introduce three kinds of dataset splits: **Easy**, **Moderate** and **Hard** as follows[2], which indicates how related the test tasks are to the training tasks, and how much semantic knowledge can be transferred.

Easy: We choose one base category from each parent category to construct the meta-test set, which represents the

[2]We include more split details in the supplementary material.

Table 3: Classification on CIFAR-100

Method	# Params	Feature Extractor	Easy		Moderate		Hard	
			1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MAML (Finn et al. 2017)	0.1M	Conv64	$55.17 \pm 1.90\%$	$74.12 \pm 0.86\%$	$46.30 \pm 1.90\%$	$58.49 \pm 0.94\%$	$38.46 \pm 1.83\%$	$52.17 \pm 0.89\%$
STANet(1-layer)	0.1M	Conv64	$\mathbf{65.51 \pm 0.44\%}$	$\mathbf{78.31 \pm 0.38\%}$	$\mathbf{56.62 \pm 0.46\%}$	$\mathbf{67.38 \pm 0.51\%}$	$\mathbf{42.94 \pm 0.41\%}$	$\mathbf{54.91 \pm 0.38\%}$
STANet(3-layer)	0.1M	Conv64	$\mathbf{66.11 \pm 0.42\%}$	$\mathbf{78.54 \pm 0.51\%}$	$\mathbf{57.31 \pm 0.39\%}$	$\mathbf{68.71 \pm 0.43\%}$	$\mathbf{42.98 \pm 0.43\%}$	$\mathbf{55.23 \pm 0.42\%}$

Table 4: Omniglot Performance

Method	# Params	Feature Extractor	5-way Accuracy	
			1-shot	5-shot
Matching Net [33]	0.1M	Conv64	98.1%	98.9%
Prototypical Net (Snell et al. 2017)	0.1M	Conv64	97.4%	99.3%
GNN (Satorras et al. 2018)	0.4M	Conv64	99.2%	99.7%
MAML (Finn et al. 2017)	0.1M	Conv64	$98.7 \pm 0.4\%$	$99.9 \pm 0.3\%$
SNAIL [17]	2.7M	Conv64	$98.96 \pm 0.20\%$	$99.75 \pm 0.11\%$
STANet(1-Layer)	0.1M	Conv64	$\mathbf{98.10 \pm 0.11\%}$	$\mathbf{99.41 \pm 0.09\%}$
STANet(3-Layer)	0.1M	Conv64	$\mathbf{98.69 \pm 0.22\%}$	$\mathbf{99.59 \pm 0.33\%}$

case that at the meta level, the test and train set are from different categories but share the same parents.
Moderate: We select 2 or 3 base categories from 7 parent categories to build the meta-test set with 20 base classes, and use the remaining 80 classes for the meta-train set.
Hard: We choose 4 parent categories randomly, and use their 20 base categories as the meta-test set. The remaining 16 parent categories and 80 base categories are employed for the meta-train set.

Experimental Results. Our experiments on the Meta-CIFAR100 dataset aim to investigate the impact of the task distribution in the few-shot classification. We include the MAML method [6], which has released its code, for comparison with the state of the art.

From results in Table 3, we can see that our approach outperforms the baseline method by a large margin in all three different settings. In addition, as the STANet is able to exploit the semantic similarity during feature learning, it achieves the largest performance gain for the **Easy** case where the meta-train and meta-test sets have the highest similarity in semantic features. When the meta-train and test are less similar, the meta-learning task becomes more difficult but the performance gap only decreases mildly.

5.3. Omniglot

Dataset. The Omniglot dataset [12] consists of 1623 characters (classes) from multiple alphabet vocabularies. We follow the setting in [33] to split the dataset into 1200 classes for training and the remaining 423 for testing, and augment the dataset by rotation proposed by [23].

Experimental Results The Omniglot dataset [12] has been widely used for testing few-shot learning methods and most recent methods achieve strong performances. Here we use it as a sanity check to validate our method. For semantic regularization, we choose the parent level of the base categories, which include 39 classes, as a coarse-level supervision in training the embedding network.

The overall comparison results are shown in Table 4. We can see that our STANet achieves competitive performance on the Omniglot dataset in comparison with the state-of-the-art methods. This indicates that our approach performs well on different types of image data.

5.4. Implementation Details

Task Sampling and Evaluation We focus on few-shot image recognition task and conduct all our experiments in the N-way m-shot setting. Specifically, we build our meta-training and meta-test dataset by sampling tasks as follows: for each task, we first randomly sample N image classes and then sample $Nm+1$ examples from N classes, including m images per class for the task-train and one for the task-test set.

We adopt a mini-batch learning strategy and in each training epoch, we use sample-without-replacement to select N classes for each training task. This makes sure that each training class occurs once in every epoch. For evaluation, we report the average accuracy on the meta-test set, which consists of tasks sampled from the test data.

Training strategy For MiniImageNet, We use *adam* optimizer with learning rate at $3e^{-4}$, weight decay at $5e^{-4}$ and a meta batch size of 16 and 8 for 1-shot and 5-shot in the training phrase respectively. The λ of multi-loss is 0.5.

6. Conclusion

In this work, we have proposed a simple and yet effective meta-learning method based on a dual attention deep network. Our approach has several advantages over the prior works. First, by exploiting the spatial attention and shared semantics, we are able to learn a robust semantic-aware image representation. In addition, our attention mechanism is easy to interpret in terms of the prior knowledge learned by the meta-learner. Furthermore, we demonstrate the efficacy of our approach by extensive experiments on the MiniImageNet, Omniglot and a new Meta-CIFAR100 benchmark, which clearly show that our network has achieved competitive or the state-of-the-art performances.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *NIPS*, 2016. 2

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*, 2015. 3

[3] Yu Cheng, Mo Yu, Xiaoxiao Guo, and Bowen Zhou. Few-shot learning with meta metric learners. In *NIPS 2017 Workshop on Meta-Learning*, 2017. 3

[4] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, 2003. 2

[5] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *TPAMI*, 2006. 2

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2, 3, 8

[7] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[9] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, 2015. 2

[10] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009. 6, 7

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 6

[12] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *CogSci*, 2011. 2, 8

[13] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 2015. 1, 2, 6

[14] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 2017. 1

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 2015. 2

[16] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-SGD: Learning to Learn Quickly for Few Shot Learning. *arXiv:1707.09835*, 2017. 2

[17] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018. 1, 3, 5, 6, 8

[18] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *ICML*, 2017. 2, 3

[19] Devang K Naik and RJ Mammone. Meta-neural networks that learn by learning. In *IJCNN*, 1992. 2

[20] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L. Yuille. Few-shot image recognition by predicting parameters from activations. In *CVPR*, June 2018. 3, 6

[21] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2, 3, 6

[22] 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. *ICCV*, 2015. 6

[23] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 1, 3, 8

[24] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *ICLR*, 2018. 3

[25] Jurgen Schmidhuber. Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook. Diploma thesis, TUM*, 1987. 2

[26] Pranav Shyam, Shubham Gupta, and Ambedkar Dukkipati. Attentive recurrent comparators. In *ICML*, 2017. 2

[27] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 2

[28] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *NIPS*, 2015. 3

[29] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, June 2018. 2, 6

[30] Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*. 1998. 2

[31] Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. In *NIPS*, 2017. 1, 2

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*. 2017. 1, 3, 4

[33] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, 2016. 1, 2, 3, 6, 8

[34] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 1, 3