Enhancing Few-Shot Image Classification with Unlabelled Examples

Peyman Bateni[1,6*†], Jarred Barber[2*‡], Jan-Willem van de Meent[3], Frank Wood[1,4,5]
University of British Columbia[1], Amazon[2], Northeastern University[3], Inverted AI[4], MILA[5], Beam AI[6]
pbateni@cs.ubc.ca, jarred.barber@gmail.com, j.vandemeent@northeastern.edu, fwood@cs.ubc.ca

Abstract

We develop a transductive meta-learning method that uses unlabelled instances to improve few-shot image classification performance. Our approach combines a regularized Mahalanobis-distance-based soft k-means clustering procedure with a modified state of the art neural adaptive feature extractor to achieve improved test-time classification accuracy using unlabelled data. We evaluate our method on transductive few-shot learning tasks, in which the goal is to jointly predict labels for query (test) examples given a set of support (training) examples. We achieve state of the art performance on the Meta-Dataset, mini-ImageNet and tiered-ImageNet benchmarks. All trained models and code have been made publicly available[1].

1. Introduction

Deep neural networks have transformed machine learning and computer vision [16, 50, 14, 11, 21, 12, 38, 40, 10, 46], enabled in part by the development of large and diverse sets of curated training data [52, 13, 22, 47, 50]. However, in many image classification tasks, millions of labelled examples are not available; therefore, techniques that can achieve sufficient classification performance with few labels are required. This has motivated research on few-shot learning [6, 57, 56, 3], which seeks to develop methods for developing classifiers with much smaller datasets. Given a few labelled "support" images per class, a few-shot image classifier is expected to produce labels for a given set of unlabelled "query" images. Typical approaches to few-shot learning adapt a base classifier network to a new support set through various means, such as learning new class embeddings [48, 54, 51], amortized [41, 34] or iterative [58] partial adaptation of the feature extractor, and complete fine-tuning of the entire network end-to-end [37, 7].

In addition to the standard fully supervised setting, techniques have been developed to exploit additional unlabeled

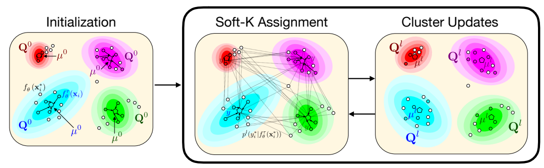


Figure 1: Transductive CNAPS' soft k-means Mahalanobis-distance based clustering procedure. First, cluster parameters are initialized using the support examples. Then, during cluster update iterations, query examples are assigned class probabilities as soft labels and subsequently, both soft-labelled query examples and labelled support examples are used to estimate new cluster parameters.

support data (semi-supervision) [39] as well as information present in the query set (transduction) [28, 18]. In our work, we focus on the transductive paradigm, where the entire query set is labeled at the same time. This allows us to exploit the additional unlabeled data, with the hopes of improving classification performance. Existing transductive few-shot methods reason about unlabelled examples by performing k-means clustering with Euclidean distance [39] or message passing in graph convolutional networks [28, 18].

Since few-shot classification requires handling a varying number of classes, an important architectural choice is the final feature to class mapping. Previous methods have used the Euclidean distance [39], the absolute difference [19], cosine similarity [54], linear classification [7, 41] or additional neural network layers [18, 51]. Bateni et al. [2] introduced a class-adaptive Mahalanobis distance. Their method, Simple CNAPS, uses a conditional neural-adaptive feature extractor, along with a regularized Mahalanobis-distance-based classifier. This modification to CNAPS [41] achieved improved performance on the Meta-Dataset benchmark [53], only recently surpassed by SUR [5] and URT [27]. However, its performance suffers when there are five or fewer support examples available per class.

Motivated by these observations, we explore the use of unlabelled examples through transductive learning within the same framework as Simple CNAPS. Our contributions are as follows. **(1)** We propose a transductive few-shot learner, namely Transductive CNAPS, that extends Simple CNAPS with a transductive two-step task encoder, as

*Authors contributed equally
†Work performed while at Inverted AI
‡Work performed while at Charles River Analytics
[1]Code available at github.com/plai-group/simple-cnaps

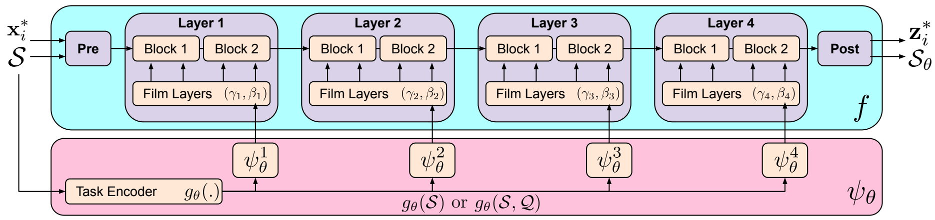


Figure 2: Overview of neural adaptive feature extraction in Transductive and Simple CNAPS. Figure adapted from [2].

well as an iterative soft k-means procedure for refining class parameter estimates (mean and covariance) using both labelled and unlabelled examples. **(2)** We demonstrate the efficacy of our approach by achieving new state of the art performance on Meta-Dataset [53]. **(3)** When deployed with a feature extractor trained on their respective training sets, Transductive CNAPS achieves state of the art performance on 4 out of 8 settings on mini-ImageNet [48] and tiered-Imagenet [39], while matching state of the art on another 2. **(4)** When additional non-overlapping classes from ImageNet [42] are used to train the feature extractor, Transductive CNAPS is able to leverage this example-rich feature extractor to achieve state of the art across the board on mini-ImageNet and tiered-ImageNet.

2. Related Work

2.1. Few-Shot Learning using Labelled Data

Early work on few-shot visual classification has focused on improving classification accuracy through the use of better classification metrics with a meta-learned non-adaptive feature extractor. Matching networks [54] use cosine similarities over feature vectors produced by independently learned feature extractors. Siamese networks [19] classify query images based on the nearest support example in feature space, under the L_1 metric. Relation networks [51] and variants [18, 44] learn their own similarity metric, parameterised through a Multi-Layer Perceptron. More recently, Prototypical Networks [48] learn a shared feature extractor that is used to produce class means in a feature space where the Euclidean distance is used for classification. ReMP[2] [59] extends this framework by incorporating self-attention for learning of prototypes in a rectified metric space, maintaining metric consistency between training and testing tasks.

Other work has focused on adapting the feature extractor for new few-shot tasks. Transfer learning by fine-tuning pretrained visual classifiers [58] was an early approach that proved limited in success due to issues arising from overfitting. MAML [7] and its variants [31, 32, 37] learn meta-parameters that allow fast task-adaptation with only a few gradient updates. Work has also been done on partial adaptation of feature extractors using conditional neural adaptive processes [34, 8, 41, 2]. These methods rely on channel-wise adaptation of pretrained convolutional layers by adjusting parameters of FiLM layers [35] inserted throughout the network. Our work builds on the most recent of these neural adaptive approaches, specifically Simple CNAPS [2]. SUR [5] and URT [27] are two very recent methods that employ universal representations stemming from multiple domain-specific feature extraction heads. URT [27], which was developed and released publicly in parallel to this work, achieves state of the art performance by using a universal transformation layer.

2.2. Few-Shot Learning using Unlabelled Data

Several approaches [18, 28, 39] have also explored the use of unlabelled instances for few-shot visual classification. EGNN [18] employs a graph convolutional edge-labelling network for iterative propagation of labels from support to query instances. Similarly, TPN [28] learns a graph construction module for neural propagation of soft labels between elements of the query set. These methods rely on a neural parameterization of distance within the feature space. TEAM [36] uses an episodic-wise transductive adaptable metric for performing inference on query examples using a task-specific metric. Song et al. [49] use a cross attention network with a transductive iterative approach for augmenting the support set using the query examples. TAFSSL[2] [25] improves few-shot learning accuracy in transductive and semi-supervised settings by performing a search for a compact feature sub-space that is discriminative for a given few-shot test-task.

The closest approach to our work is that of Ren et al. [39]. Their method extends prototypical networks [48] by performing a single additional soft-label weighted estimation of class prototypes. Our work, on the other hand, differs in three ways. First, we produce soft-labelled estimates of both class mean and covariance. Second, we use an expectation-maximization (EM) algorithm that performs a dynamic number of soft-label updates, depending on the task at hand. Lastly, we employ a neural-adaptive procedure for feature extraction that is conditioned on a two-step learned transductive task representation, as opposed to a fixed feature-extractor. As we discuss in Section 4.2, this novel task-representation encoder is responsible for substantial performance gains on out-of-domain tasks.

[2]Note that we do not directly compare to these methods as they are either unpublished (ArXiv) or were developed concurrent to our work.

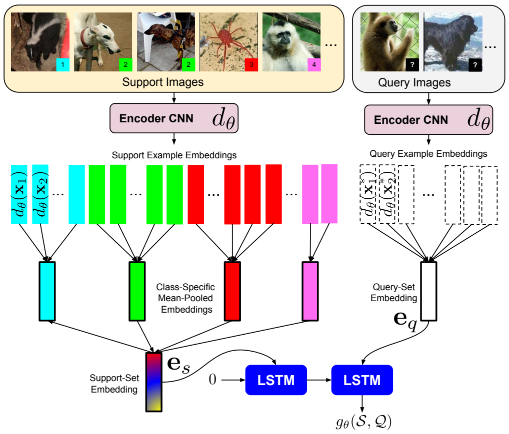


Figure 3: Overview of the transductive task-encoding procedure, $g_\theta(\mathcal{S}, \mathcal{Q})$, used in Transductive CNAPS.

3. Method

3.1. Problem Definition

Following [48, 2, 41, 7], we focus on a few-shot classification setting where a distribution D over image classification tasks $(\mathcal{S}, \mathcal{Q})$ is provided for training. Each task $(\mathcal{S}, \mathcal{Q}) \sim D$ consists of a support set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ of labelled images and a query set $\mathcal{Q} = \{\mathbf{x}_i^*\}_{i=1}^{m}$ of unlabelled images; the goal is to predict labels for these query examples, given the (typically small) support set. Each query image $\mathbf{x}_i^* \in \mathcal{Q}$ has a corresponding ground truth label y_i^* available at training time. A model will be trained by minimizing, over some parameters θ (which are shared across tasks), the expected query set classification loss over tasks: $\mathbb{E}_{(\mathcal{S},\mathcal{Q})\sim D}[\sum_{\mathbf{x}_i^* \in \mathcal{Q}} - \log p_\theta(y_i^* | \mathbf{x}_i^*, \mathcal{S}, \mathcal{Q})]$; the inclusion of the dependence on all of $\mathcal{Q}$ here allows for the model to be transductive. At test time, a separate distribution of tasks generated from previously unseen images and classes is used to evaluate performance. Let us also define *shot* as the number of support examples per class, and *way* as the number of classes within the task.

3.2. Simple CNAPS

Our method extends the Simple CNAPS [2] architecture for few-shot visual classification. Simple CNAPS performs few-shot classification in two steps.

First, it computes task-adapted features for every support and query example. This part of the architecture is the same as that in CNAPS [41], and is based on the FiLM meta-learning framework [35]. Second, it uses the support set to estimate a per-class Mahalanobis metric, which is used to assign query examples to classes. The architecture uses a ResNet18 [13] feature extractor. Within each residual block, Feature-wise Linear Modulation (FiLM) layers compute a scale factor γ and shift β for each output channel, using block-specific adaptation networks ψ_θ that are conditioned on a task encoding. The task encoding $g_\theta(\mathcal{S})$ consists of the mean-pooled feature vectors of support examples produced by d_θ, a separate but end-to-end learned Convolution Neural Network (CNN). This produces an adapted feature extractor f_θ (which implicitly depends on the support set $\mathcal{S}$) that maps support/query images onto the corresponding adapted feature space. We will denote by $\mathcal{S}_\theta, \mathcal{Q}_\theta$ versions of the support/query sets where each image is mapped into its feature representation $\mathbf{z} = f_\theta(\mathbf{x})$.

Simple CNAPS then computes a Mahalanobis distance relative to each class k by estimating a mean $\boldsymbol{\mu}_k$ and regularized covariance $\mathbf{Q}_k$ in the adapted feature space, using the support instances:

$$\boldsymbol{\mu}_k = \frac{1}{n_k} \sum_i \mathbb{I}[y_i = k] \, \mathbf{z}_i \tag{1}$$

$$\mathbf{Q}_k = \lambda_k \, \boldsymbol{\Sigma}_k + (1 - \lambda_k) \, \boldsymbol{\Sigma} + \beta I, \quad \lambda_k = \frac{n_k}{n_k + 1} \tag{2}$$

Here $\mathbb{I}[y_i = k]$ is the indicator function and $n_k = \sum_i \mathbb{I}[y_i = k]$ is the number of examples from class k in the support set $\mathcal{S}$. The ratio λ_k balances a task-conditional sample covariance $\boldsymbol{\Sigma}$ and a class-conditional sample covariance $\boldsymbol{\Sigma}_k$:

$$\boldsymbol{\Sigma} = \frac{1}{n} \sum_i \big(\mathbf{z}_i - \boldsymbol{\mu}\big) \big(\mathbf{z}_i - \boldsymbol{\mu}\big)^T \tag{3}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{n_k} \sum_i \mathbb{I}[y_i = k] \, \big(\mathbf{z}_i - \boldsymbol{\mu}_k\big) \big(\mathbf{z}_i - \boldsymbol{\mu}_k\big)^T \tag{4}$$

where $\boldsymbol{\mu} = \frac{1}{n} \sum_i \mathbf{z}_i$ is the task-level mean. When few support examples are available for a particular class, λ_k is small, and the estimate is regularized towards the task-level covariance $\boldsymbol{\Sigma}$. As the number of support examples for the class increases, the estimate tends towards the class-conditional covariance $\boldsymbol{\Sigma}_k$. Additionally, a regularizer βI (we set $\beta = 1$ in our experiments) is added to ensure invertibility. Given the class means and covariances, Simple CNAPS computes class probabilities for each query feature vector $\mathbf{z}_i^*$ through a softmax over the squared Mahalanobis distances with respect to each class:

$$p(y^* = k \mid \mathbf{z}^*) \propto \exp\big(-(\mathbf{z} - \boldsymbol{\mu}_k)^T \mathbf{Q}_k^{-1} (\mathbf{z} - \boldsymbol{\mu}_k)\big) \tag{5}$$

3.3. Transductive CNAPS

Transductive CNAPS extends Simple CNAPS by taking advantage of the query set, both in the feature adaptation step and the classification step. First, the task encoder g_θ is extended to incorporate both a support-set embedding $\mathbf{e}_s$ and a query-set embedding $\mathbf{e}_q$ such that,

$$\mathbf{e}_s = \frac{1}{K} \sum_k \frac{1}{n_k} \sum_i \mathbb{I}[y_i = k] \, d_\theta(\mathbf{x}_i), \tag{6}$$

$$\mathbf{e}_q = \frac{1}{n_q} \sum_{i*} d_\theta(\mathbf{x}_i^*), \tag{7}$$

where d_θ is a learned CNN. The support embedding $\mathbf{e}_s$ is formed by an average of (encoded) support examples, with weighting inversely proportional to their class counts to prevent bias from class imbalance. The query embedding $\mathbf{e}_q$ uses simple mean-pooling; both $\mathbf{e}_s$ and $\mathbf{e}_q$ are invariant to

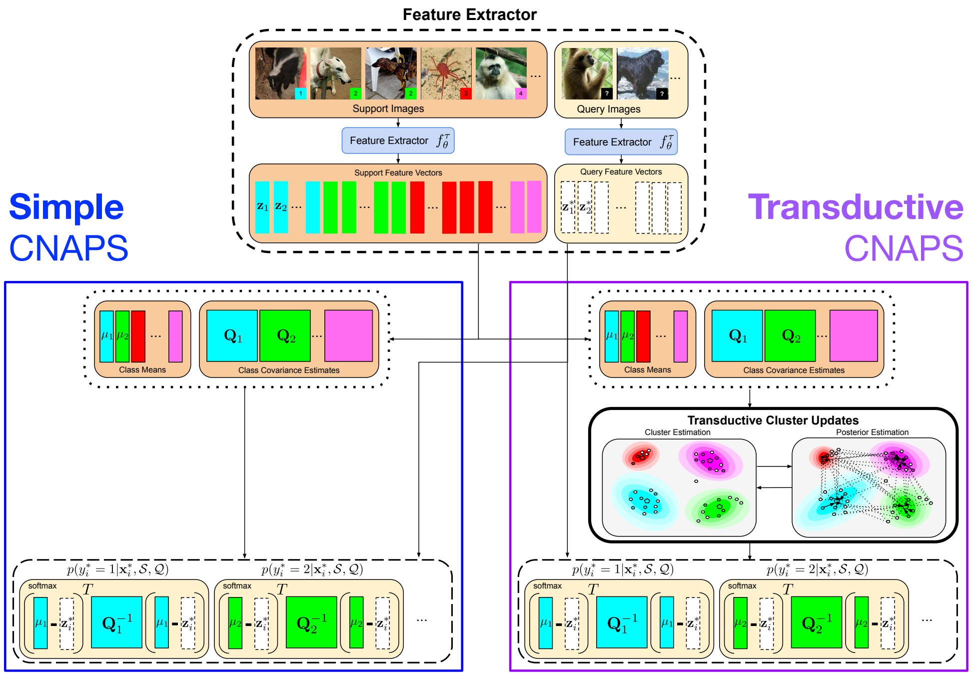
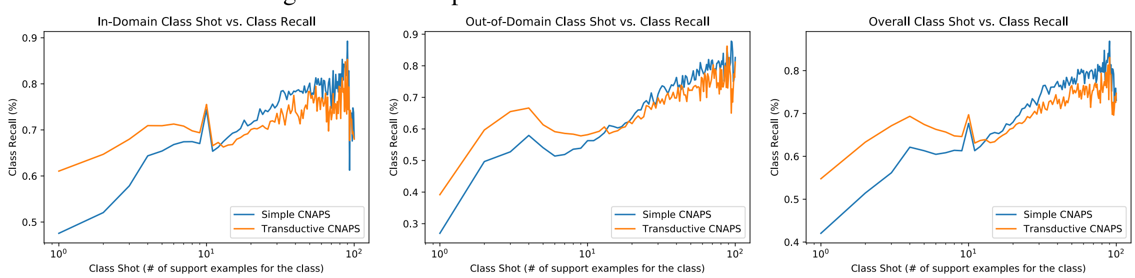


Figure 4: Transductive CNAPS (right) extends the Mahalanobis-distance based classifier in Simple CNAPS (left) through transductive soft k-means clustering of the visual space.



(a) In-Domain (b) Out-of-Domain (c) Overall

Figure 5: Class recall (otherwise noted as in-class query accuracy) averaged between classes across all tasks and (a: In-Domain, b: Out-of-domain, c: all) Meta-Dataset sub-datasets. Class recalls have been grouped together, averaged and plotted according to class shot in (a), (b), and (c).

permutations of the respective support/query instances. We then process $\mathbf{e}_s$ and $\mathbf{e}_q$ through two steps of a Long Short Term Memory (LSTM) network in the same order to generate the final transductive task-embedding $g_\theta(\mathcal{S}, \mathcal{Q})$ used for adaptation. This process is visualized in Figure 3.

Second, we can interpret Simple CNAPS as a form of "supervised clustering" in feature space; each cluster (corresponding to a class k) is parameterized with a centroid $\boldsymbol{\mu}_k$ and a metric $\mathbf{Q}_k^{-1}$, and we interpret (5) as class assignment probabilities based on the distance to each centroid. With this viewpoint in mind, a natural extension to consider is to use the estimates of the class assignment probabilities on unlabelled data to refine the class parameters $\boldsymbol{\mu}_k, \mathbf{Q}_k$ in a soft k-means framework based on per-cluster Mahalanobis distances [30]. In this framework, as shown in Figure 1, we alternate between computing updated assignment probabilities using (5) on the query set and using those assignment probabilities to compute updated class parameters.

We define $\mathcal{R}_\theta = \mathcal{S}_\theta \sqcup \mathcal{Q}_\theta$ as the disjoint union of the support set and the query set. For each element of $\mathcal{R}_\theta$, which we index by j, we define responsibilities w_{jk} in terms of their class predictions when it is part of the query set and in terms of the label when it is part of the support set,

$$w_{jk} = \begin{cases} p(y'_j = k \mid \mathbf{z}'_j) & \mathbf{z}'_j \in \mathcal{Q}_\theta, \\ \mathbb{I}[y'_j = k] & (\mathbf{z}'_j, y'_j) \in \mathcal{S}_\theta. \end{cases} \tag{8}$$

Using these responsibilities we can incorporate unlabelled samples from the support set by defining weighted estimates $\boldsymbol{\mu}'_k$ and $\mathbf{Q}'_k$:

$$\boldsymbol{\mu}'_k = \frac{1}{n'_k} \sum_j w_{jk}\, \mathbf{z}'_j, \tag{9}$$

$$\mathbf{Q}'_k = \lambda'_k \boldsymbol{\Sigma}'_k + (1 - \lambda'_k)\boldsymbol{\Sigma}' + \beta I, \tag{10}$$

where $n'_k = \sum_j w_{jk}$ defines $\lambda'_k = n'_k/(n'_k + 1)$. The covariance estimates $\boldsymbol{\Sigma}'$ and $\boldsymbol{\Sigma}'_k$ are

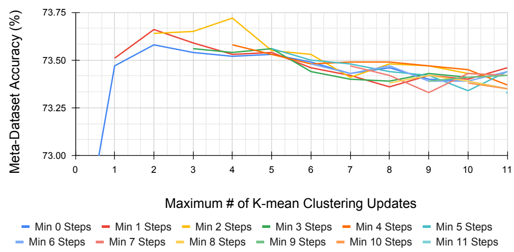


Figure 6: Evaluating Transductive CNAPS on Meta-Dataset with different minimum and maximum number of steps. Performances reported stem from five run averages.

$$\boldsymbol{\Sigma}' = \frac{1}{\sum_k n'_k} \sum_{jk} w_{jk} \left(\mathbf{z}'_j - \boldsymbol{\mu}'\right) \left(\mathbf{z}'_j - \boldsymbol{\mu}'\right)^T \tag{11}$$

$$\boldsymbol{\Sigma}'_k = \frac{1}{n'_k} \sum_{j} w_{jk} \left(\mathbf{z}'_j - \boldsymbol{\mu}'_k\right) \left(\mathbf{z}'_j - \boldsymbol{\mu}'_k\right)^T \tag{12}$$

where $\boldsymbol{\mu}' = \left(\sum_k n'_k\right)^{-1} \sum_{jk} w_{jk} \mathbf{z}'_j$ is the task mean.

These update equations are weighted versions of the original Simple CNAPS estimators from Section 3.2, and reduce to them exactly in the case of an empty query set.

Algorithm 1 summarizes the soft k-means procedure based on these updates. We initialize our weights using only the labelled support set. We use those weights to compute class parameters, then compute updated weights using both the support and query sets. At this point, the weights associated with the query set $\mathcal{Q}$ are the same class probabilities as estimated by Simple CNAPS. However, we continue this procedure iteratively until we reach either reach a maximum number of iterations, or until class assignments $\text{argmax}_k \, w_{jk}$ stop changing.

Unlike the transductive task-encoder, this second extension, namely the soft k-mean iterative estimation of class parameters, is used at test time only. During training, a single estimation is produced for both mean and covariance using only the support examples. This, as we discuss more in Section 4.2, was shown to empirically perform better. See Figure 4 for a high-level visual comparison of classification in Simple CNAPS vs. Transductive CNAPS.

3.4. Relationship to Bregman Soft Clustering

The procedure in Algorithm 1 resembles the Bregman clustering algorithms proposed by Banerjee et al. [1]. Specifically, the updates to soft assignments w_{jk} in Equation 5 are the semi-supervised equivalent of those in Bregman soft clustering, in which the divergence is based on the Mahalanobis distance $F(\mathbf{z}) = \mathbf{z}^T \mathbf{Q}^{-1} \mathbf{z}$,

$$D_F(\mathbf{z}, \mathbf{z}') = F(\mathbf{z}) - F(\mathbf{z}') - \nabla F(\mathbf{z}')^T (\mathbf{z} - \mathbf{z}'). \tag{13}$$

However, Algorithm 1 differs in that it updates both $\boldsymbol{\mu}'_k$ and $\mathbf{Q}'_k$ at each iteration, rather than just $\boldsymbol{\mu}'_k$.

In general, any (regular) exponential family can be associated with a Bregman divergence and vice versa, which gives rise to a correspondence between EM-based clustering and Bregman soft clustering algorithms [1]. Standard Bregman soft clustering corresponds to EM in which the likelihood is a Gaussian with unknown mean and a known covariance $\mathbf{Q}$ that is shared across clusters. The case where the covariance is unknown corresponds to Gaussian mixture models (GMMs), but the function $F(\mathbf{z})$ is not simply the Mahalanobis distance in this case.

The updates for $\boldsymbol{\mu}'_k$ and $\mathbf{Q}'_k$ in Algorithm 1 are equivalent to those in a GMM that incorporates regularization for the covariances. However, GMM clustering differs in the calculation of the assignment probabilities

$$p(y^* = k \mid \mathbf{z}^*) \propto \pi_k \exp\left(-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu}_k)^T \mathbf{Q}_k^{-1} (\mathbf{z} - \boldsymbol{\mu}_k) - \frac{1}{2} \log |\mathbf{Q}_k|\right).$$

These probabilities incorporate a term $\pi_k = p(y^* = k)$, which defines a prior probability of assignments to a cluster, and a term $\exp(-\log |\mathbf{Q}_k|)$, which reflects the fact that GMMs employ a likelihood with unknown covariance.

In short, our clustering procedure employs an update to soft assignments w_{jk} that is similar to the one in soft Bregman clustering, but employs updates to $\boldsymbol{\mu}'_k$ and $\mathbf{Q}'_k$ that are similar to those in a (regularized) GMM. In Section 4.2 we demonstrate through ablations that this combination of updates improves emperical performance relative to baselines that perform GMM-based clustering.

4. Experiments

4.1. Benchmarks

Meta-Dataset [53] is a few-shot visual classification benchmark that consists of 10 widely used datasets: ILSVRC-2012 (ImageNet) [42], Omniglot [23], FGVC-Aircraft (Aircraft) [29], CUB-200-2011 (Birds) [55], Describable Textures (DTD) [4], QuickDraw [17], FGVCx Fungi (Fungi) [45], VGG Flower (Flower) [33], Traffic Signs (Signs) [15] and MSCOCO [26]. Consistent with past work [41, 2], we train our model on the official training splits of the first 8 datasets and use the test splits to evaluate in-domain performance. We use the remaining two datasets as well as three external benchmarks, namely MNIST [24], CIFAR10 [20] and CIFAR100 [20], for out-of-domain evaluation.

Task generation in Meta-Dataset follows a complex procedure where tasks can be of different *ways* and individual classes can be of varying *shots* even within the same task. Specifically, for each task, the task *way* is first sampled uniformly between 5 and 50 and *way* classes are selected at random from the corresponding class/dataset split. Then, for each class, 10 instances are sampled at random and used as query examples for the class, while of the remaining images for the class, a *shot* is sampled uniformly from [1, 100] and *shot* number of images are selected at random as support examples with total support set size of 500.

Additional dataset-specific constraints are enforced, as discussed in Section 3.2 of [53], and since some datasets

Algorithm 1 Iterative Refinement in Transductive CNAPS

procedure COMPUTE_QUERY_LABELS($\mathcal{S}_\theta, \mathcal{Q}_\theta, N_{\text{iter}}$)

  For j ranging over support and query sets, $w_{jk} \leftarrow \begin{cases} 1 & \text{if } (\mathbf{z}'_j, y'_j) \in \mathcal{S}_\theta \text{ and } y_j = k \\ 0 & \text{otherwise} \end{cases}$

  for iter = $0 \cdots N_{\text{iter}}$ **do** ▷ The first iteration is equivalent to Simple CNAPS;

    Compute class parameters $\boldsymbol{\mu}_k, \mathbf{Q}_k$ according to update equations (9)-(11)

    Compute class weights using class parameters according to (8)

    break if the most probable class for each query example hasn't changed

  end for

  return class probabilities w_{jk} for j corresponding to $\mathcal{Q}_\theta$

end procedure

have fewer than 50 classes and fewer than 100 images per class, the overall *way* and *shot* distributions resemble Poisson distributions where most tasks have fewer than 10 classes and most classes have fewer than 10 support examples (see Appendix-A.1). Following [2] and [41], we first train our ResNet18 feature extractor on the Meta-Dataset defined training split of ImageNet following the procedure in Appendix-A.3. The ResNet18 parameters are then kept fixed while we train the adaptation network for a total of 110K sampled tasks using Episodic Training [48, 7] (see Appendix-A.3 for details).

Mini/tiered-ImageNet [54, 39] are two benchmarks for few-shot learning. Both datasets employ subsets of ImageNet [42] with a total of 100 classes and 60K images in mini-ImageNet and 608 classes and 779K images in tiered-ImageNet. Unlike Meta-Dataset, tasks across these datasets have pre-defined *shots* and *ways* that are uniform across every task generated in the specified setting.

Following [32, 28, 48], we report performance on the 1/5-*shot* 5/10-*way* settings across both datasets with 10 query examples per class. We first train the ResNet18 on the training set of the corresponding benchmark at hand following the procedure noted in Appendix-A.4. We also consider a more feature-rich ResNet18 trained on the larger ImageNet dataset. However, we exclude classes and examples from test sets of mini/tiered-ImageNet to address potential class/example overlap issues, resulting in 825 classes and 1,055,494 images remaining. Then, with the ResNet18 parameters fixed, we train episodically for 20K tasks (see Appendix-A.2 for details).

4.2. Results

Evaluation on Meta-Dataset: In-domain, out-of-domain and overall rankings on Meta-Dataset are shown in Table 1. Following [2] and [41], we pretrain the feature extractor on the training split of the ImageNet subset of Meta-Dataset. Transductive CNAPS sets new state of the art accuracy on 2 out of the 8 in-domain datasets, while matching other methods on 2 of the remaining domains. On out-of-domain tasks, it performs better with new state of the art performance on 4 out of the 5 out-of-domain datasets, Overall, it produces an average rank of 1.9 among all datasets, the best among the methods, with an average rank of 2.1 on in-domain tasks, only second to URT which was developed parallel to Transductive CNAPS, and 1.6 on out-of-domain tasks, the best among even the most recent methods.

Evaluation on mini/tiered-ImageNet: We consider two feature extractor training settings on these benchmarks. First, we use the feature extractor trained on the corresponding training split of the mini/tiered-ImageNet. As shown in Table 2, on tiered-ImageNet, Transductive CNAPS achieves state of art accuracy on both 10-way settings while matching state of the art performance of LEO [43] on the 5-way settings. On the mini-ImageNet, Transductive CNAPS outperforms other methods on 10-way settings while coming second to LEO [43] and TADAM [41, 34] on 5-way tasks.

We attribute this difference in performance between mini-ImageNet and tiered-ImageNet to the fact that mini-ImageNet only provides 38,400 training examples, compared to 448,695 examples provided by tiered-ImageNet. This results in a lower performing ResNet-18 feature extractor (which is trained in a traditional supervised manner). This hypothesis is further supported by the results provided in our second model (denoted by "FETI", for "Feature Extractor Trained with ImageNet", in Table 2). In this model, we train the feature extractor with a much larger subset of ImageNet, which has been carefully selected to prevent any possible overlap (in examples or classes) with the test sets of mini/tiered-ImageNet. Transductive CNAPS is able to take advantage of the more example-rich feature extractor, resulting in state-of-the-art performance across the board. Additionally, it outperforms the Simple CNAPS baseline by a large margin, even when using the same example-rich feature extractor; this demonstrates that leveraging additional query set information yields empirical gains.

Performance vs. Class Shot: In Figure 5, we examine the relationship between class recall (i.e. accuracy among query examples belonging to the class itself) and the num-

	In-Domain Accuracy (%)								Out-of-Domain Accuracy (%)					Avg Rank		
Model	ImageNet	Omniglot	Aircraft	Birds	DTD	QuickDraw	Fungi	Flower	Signs	MSCOCO	MNIST	CIFAR10	CIFAR100	In	Out	All
RelationNet	30.9±0.9	86.6±0.8	69.7±0.8	54.1±1.0	56.6±0.7	61.8±1.0	32.6±1.1	76.1±0.8	37.5±0.9	27.4±0.9	NA	NA	NA	10.5	11.0	10.6
MatchingNet	36.1±1.0	78.3±1.0	69.2±1.0	56.4±1.0	61.8±0.7	60.8±1.0	33.7±1.0	81.9±0.7	55.6±1.1	28.8±1.0	NA	NA	NA	10.1	8.5	9.8
MAML	37.8±1.0	83.9±1.0	76.4±0.7	62.4±1.1	64.1±0.8	59.7±1.1	33.5±1.1	79.9±0.8	42.9±1.3	29.4±1.1	NA	NA	NA	9.2	10.5	9.5
ProtoNet	44.5±1.1	79.6±1.1	71.1±0.9	67.0±1.0	65.2±0.8	64.9±0.9	40.3±1.1	86.9±0.7	46.5±1.0	39.9±1.1	NA	NA	NA	8.2	9.5	8.5
ProtoMAML	46.5±1.1	82.7±1.0	75.2±0.8	69.9±1.0	68.3±0.8	66.8±0.9	42.0±1.2	88.7±0.7	52.4±1.1	41.7±1.1	NA	NA	NA	7.1	8.0	7.3
CNAPS	52.3±1.0	88.4±0.7	80.5±0.6	72.2±0.9	58.3±0.7	72.5±0.8	47.4±1.0	86.0±0.5	60.2±0.9	42.6±1.1	92.7±0.4	61.5±0.7	50.1±1.0	6.6	6.0	6.4
BOHB-E	55.4±1.1	77.5±1.1	60.9±0.9	73.6±0.8	72.8±0.7	61.2±0.9	44.5±1.1	90.6±0.6	57.5±1.0	51.9±1.0	NA	NA	NA	6.4	4.0	5.9
TaskNorm	50.6±1.1	90.7±0.6	83.8±0.6	74.6±0.8	62.1±0.7	74.8±0.7	48.7±1.0	89.6±0.5	67.0±0.7	43.4±1.0	92.3±0.4	69.3±0.8	54.6±1.1	4.7	4.8	4.8
Simple CNAPS	**58.6±1.1**	91.7±0.6	82.4±0.7	74.9±0.8	67.8±0.8	77.7±0.7	46.9±1.0	90.7±0.5	73.5±0.7	46.2±1.1	93.9±0.4	74.3±0.7	60.5±1.0	3.4	3.0	3.2
SUR	56.3±1.1	93.1±0.5	**85.4±0.7**	71.4±1.0	**71.5±0.8**	81.3±0.6	**63.1±1.0**	82.8±0.7	70.4±0.8	**52.4±1.1**	94.3±0.4	66.8±0.9	56.6±1.0	3.1	2.6	2.9
URT	55.7±1.0	**94.4±0.4**	**85.8±0.6**	**76.3±0.8**	**71.8±0.7**	**82.5±0.6**	**63.5±1.0**	88.2±0.6	69.4±0.8	**52.2±1.1**	94.8±0.4	67.3±0.8	56.9±1.0	**1.7**	2.8	2.2
Our Method	**58.8±1.1**	**93.9±0.4**	84.1±0.6	**76.8±0.8**	69.0±0.8	78.6±0.7	48.8±1.1	**91.6±0.4**	**76.1±0.7**	48.7±1.0	**95.7±0.3**	**75.7±0.7**	**62.9±1.0**	2.1	**1.6**	**1.9**

Table 1: Few-shot classification on Meta-Dataset, MNIST, and CIFAR10/100. Error intervals correspond to 95% confidence intervals, and bold values indicate statistically significant state of the art performance. Average rank is obtained by ranking methods on each dataset and averaging the ranks.

		mini-ImageNet Accuracy (%)				tiered-ImageNet Accuracy (%)			
		5-*way*		10-*way*		5-*way*		10-*way*	
Model	Transductive	1-*shot*	5-*shot*	1-*shot*	5-*shot*	1-*shot*	5-*shot*	1-*shot*	5-*shot*
MAML [7]	BN	48.7±1.8	63.1±0.9	31.3±1.1	46.9±1.2	51.7±1.8	70.3±1.7	34.4±1.2	53.3±1.3
MAML+ [28]	Yes	50.8±1.8	66.2±1.8	31.8±0.4	48.2±1.3	53.2±1.8	70.8±1.8	34.8±1.2	54.7±1.3
Reptile [32]	No	47.1±0.3	62.7±0.4	31.1±0.3	44.7±0.3	49.0±0.2	66.5±0.2	33.7±0.3	48.0±0.3
Reptile+BN [32]	BN	49.9±0.3	66.0±0.6	32.0±0.3	47.6±0.3	52.4±0.2	71.0±0.2	35.3±0.3	52.0±0.3
ProtoNet [48]	No	46.1±0.8	65.8±0.7	32.9±0.5	49.3±0.4	48.6±0.9	69.6±0.7	37.3±0.6	57.8±0.5
RelationNet [51]	BN	51.4±0.8	67.0±0.7	34.9±0.5	47.9±0.4	54.5±0.9	71.3±0.8	36.3±0.6	58.0±0.6
TPN [28]	Yes	51.4±0.8	67.1±0.7	34.9±0.5	47.9±0.4	59.9±0.9	73.3±0.7	44.8±0.6	59.4±0.5
AttWeightGen [9]	BN	56.2±0.9	73.0±0.6	NA	NA	NA	NA	NA	NA
TADAM [34]	BN	58.5±0.3	76.7±0.3	NA	NA	NA	NA	NA	NA
Simple CNAPS [2]	BN	53.2±0.9	70.8±0.7	37.1±0.5	56.7±0.5	63.0±1.0	80.0±0.8	48.1±0.7	70.2±0.6
LEO [43]	BN	61.8±0.1	77.6±0.1	NA	NA	66.3±0.1	81.4±0.1	NA	NA
Transductive CNAPS	Yes	55.6±0.9	73.1±0.7	42.8±0.7	59.6±0.5	65.9±1.0	81.8±0.7	54.6±0.8	72.5±0.6
Simple CNAPS [2] + FETI	BN	77.4±0.8	90.3±0.4	63.5±0.6	83.1±0.4	71.4±1.0	86.0±0.6	57.1±0.7	78.5±0.5
Transductive CNAPS + FETI	Yes	**79.9±0.8**	**91.5±0.4**	**68.5±0.6**	**85.9±0.3**	**73.8±1.0**	**87.7±0.6**	**65.1±0.8**	**80.6±0.5**

Table 2: Few-shot visual classification results on 1/5-shot 5/10-way few-shot on mini/tiered-ImageNet. For CNAP-based models, "FETI" indicates that the feature extractor used has been trained on ImageNet [42] exluding classes within the test splits of mini/tiered-ImageNet (for more details see Appendix-A.4). "BN" indicates implicit transductive conditioning on the query set through the use of batch normalization. Error intervals denote 95% confidence interval.

ber of support examples in the class (shot). As shown, Transductive CNAPS is very effective when class shot is below 10, showing large average recall improvements, especially at the 1-shot level. However, as the class shot increases beyond 10, performance drops compared to Simple CNAPS. This suggests that soft k-means learning of cluster parameters can be effective when very few support examples are available. Conversely, in high-shot classes, transductive updates can act as distractors.

Classification-Time Soft K-means Clustering: We use soft k-means iterative updates of means and covariance at test-time only. It is natural to consider training the feature adaptation network end-to-end through the soft k-means transduction procedure. We provide this comparison in the bottom-half of Table 3, with "Transductive+ CNAPS" denoting this variation. Iterative updates during training result in an average accuracy decrease of 2.5%, which we conjecture to be due to training instabilities caused by applying this iterative algorithm early in training on noisy features.

Transductive Feature Extraction vs. Classification: Our approach extends Simple CNAPS in two ways: improved adaptation of the feature extractor using a transductive task-encoding, and the soft k-means iterative estimation of class means and covariances. We perform two ablations, "Feature Extraction Only Transductive" (FEOT) and "Classification Only Transductive" (COT), to independently assess the impact of these extensions. The results are presented in Table 3; both extensions outperform Simple CNAPS. The transductive task-encoding is especially effective on out-of-domain tasks, whereas the soft k-mean learning of class parameters boosts accuracy on in-domain tasks. Transductive CNAPS is able to leverage the best of both worlds, allowing it to achieve significant gains over Simple CNAPS.

Comparison to Gaussian Mixture Models: We consider five GMM-based ablations of our method where the log-determinant is introduced into the weight updates (using a uniform class prior $\pi_k = 1/K$). Results in Table 3 correspond to their soft k-means counterparts in the same order shown. The GMM-based variations of our method and Sim-

	In-Domain Accuracy (%)								Out-of-Domain Accuracy (%)					Avg Acc.		
CNAPS Model	ImageNet	Omniglot	Aircraft	Birds	DTD	QuickDraw	Fungi	Flower	Signs	MSCOCO	MNIST	CIFAR10	CIFAR100	In	Out	All
GMM-EM+	53.3±1.0	91.8±0.6	81.2±0.6	**75.8±0.7**	**71.8±0.6**	72.9±0.7	42.8±0.9	91.0±0.4	66.1±0.8	40.3±1.0	94.2±0.4	69.0±0.7	51.3±0.9	72.6	64.2	69.3
GMM	45.3±1.0	88.0±0.9	80.8±0.8	71.4±0.8	61.1±0.7	70.7±0.8	42.9±1.0	88.1±0.6	68.9±0.7	37.2±0.9	91.4±0.5	64.5±0.7	46.6±0.9	68.5	61.7	65.9
FEOT GMM	52.6±1.1	89.6±0.7	**84.0±0.6**	76.2±0.8	66.5±0.8	73.4±0.8	45.7±1.0	89.8±0.6	74.4±0.7	44.2±1.0	93.1±0.4	71.1±0.8	56.9±1.0	72.2	67.9	70.6
COT GMM	48.7±1.0	92.3±0.5	80.0±0.7	72.4±0.7	59.8±0.7	71.1±0.7	41.4±0.9	87.7±0.5	63.6±0.8	39.2±0.8	89.8±0.5	66.9±0.7	50.5±0.8	69.2	62.0	66.4
GMM-EM	52.3±1.0	92.0±0.5	**84.3±0.6**	75.2±0.8	64.3±0.7	72.6±0.8	44.6±1.0	90.8±0.5	71.4±0.7	44.7±0.9	93.0±0.4	71.1±0.7	56.4±0.9	72.0	67.3	70.2
Transductive+	53.3±1.1	92.3±0.5	81.2±0.7	75.0±0.8	**72.0±0.7**	74.8±0.8	45.1±1.0	92.1±0.4	71.0±0.8	44.0±1.1	95.9±0.3	71.1±0.7	57.3±1.1	73.2	67.9	71.2
Simple	**58.6±1.1**	91.7±0.6	82.4±0.7	74.9±0.8	67.8±0.8	77.7±0.7	46.9±1.0	90.7±0.5	73.5±0.7	46.2±1.1	93.9±0.4	74.3±0.7	60.5±1.0	73.8	69.7	72.2
FEOT	**57.3±1.1**	90.5±0.7	82.9±0.7	74.8±0.8	67.3±0.8	76.3±0.8	47.7±1.0	90.5±0.5	**75.8±0.7**	**47.1±1.1**	94.9±0.4	74.3±0.8	**61.2±1.0**	73.4	70.7	72.4
COT	**58.8±1.1**	**95.2±0.3**	**84.0±0.6**	**76.4±0.7**	68.5±0.8	**77.8±0.7**	**49.7±1.0**	**92.7±0.4**	70.8±0.7	**47.3±1.0**	94.2±0.4	**75.2±0.7**	**61.2±1.0**	**75.4**	69.7	**73.2**
Transductive	**58.8±1.1**	93.9±0.4	**84.1±0.6**	**76.8±0.8**	69.0±0.8	**78.6±0.7**	**48.8±1.1**	91.6±0.4	**76.1±0.7**	**48.7±1.0**	**95.7±0.3**	**75.7±0.7**	**62.9±1.0**	**75.2**	**71.8**	**73.9**

Table 3: Performance of various ablations of Tranductive and Simple CNAPS on Meta-Dataset. Error intervals indicate 95% confidence intervals, and bold values indicate statistically significant state of the art performance.

	In-Domain Accuracy (%)								Out-of-Domain Accuracy (%)					Avg Accuracy		
Model	ImageNet	Omniglot	Aircraft	Birds	DTD	QuickDraw	Fungi	Flower	Signs	MSCOCO	MNIST	CIFAR10	CIFAR100	In	Out	All
Simple CNAPS	**58.6±1.1**	91.7±0.6	82.4±0.7	74.9±0.8	**67.8±0.8**	**77.7±0.7**	46.9±1.0	90.7±0.5	73.5±0.7	46.2±1.1	93.9±0.4	74.3±0.7	60.5±1.0	73.8	69.7	72.2
No Refinements	**57.3±1.1**	90.5±0.7	82.9±0.7	74.8±0.8	**67.3±0.8**	76.3±0.8	47.7±1.0	90.5±0.5	**75.8±0.7**	**47.1±1.1**	94.9±0.4	74.3±0.8	**61.2±1.0**	73.4	70.7	72.4
No Min/Max	**58.7±1.1**	**94.0±0.4**	**84.0±0.6**	**76.4±0.8**	**68.9±0.8**	**77.9±0.7**	48.0±1.0	91.6±0.5	74.0±0.8	48.3±1.0	95.7±0.3	75.5±0.7	61.3±1.0	74.9	71.0	73.4
Min 2 Max 4	**58.8±1.1**	**93.9±0.4**	**84.1±0.6**	**76.8±0.8**	**69.0±0.8**	**78.6±0.7**	48.8±1.1	**91.6±0.4**	**76.1±0.7**	48.7±1.0	**95.7±0.3**	**75.7±0.7**	**62.9±1.0**	**75.2**	**71.8**	**73.9**

Table 4: Evaluating min/max refinement restrictions in Transductive CNAPS on Meta-Dataset, MNIST, and CIFAR10/100.

ple CNAPS result in a notable 4-8% loss in overall accuracy. It is also surprising to observe that the FEOT variation matches the performance of the full GMM-EM model.

Maximum and Minimum Number of Refinements: In our experiments, we use a minimum number of 2 refinement steps of class parameters, with the maximum set to 4 on the Meta-Dataset and 10 on the mini/tiered-ImageNet benchmarks. As shown in Table 4, the refinement criteria itself, without any step constraints, results in a significant performance gain as compared to performing no refinements. In fact, it accounts for the majority of the accuracy gain for Transductive CNAPS. We further explore the impact of these step-hyperparameters on the performance on Transductive CNAPS on the Meta-Dataset in Figure 6. As shown, requiring the same number of refinement steps for every task results in sub-optimal performance. This is demonstrated by the fact that the peak performance for each minimum number of steps is achieved with a larger number of maximum steps, showcasing the importance of allowing different numbers of refinement steps depending on the task. In addition, we observe that as the number of minimum refinement steps increases, the performance improves up to two steps while declining after. This suggests that, unlike [39] where only a single refinement step leads to the best performance, our Mahalanobis-based approach can leverage extra steps to further refine the class parameters. We do see a decline in performance with a higher number of steps; this suggests that while our refinement criteria can be effective at performing different number of steps depending on the task, it can potentially lead to over-fitting, justifying the need for a well chosen maximum number of steps.

5. Discussion

In this paper, we have presented a few-shot visual classification method that achieves new state of the art performance via a transductive clustering procedure for refining class parameters derived from a previous neural adaptive Mahalanobis-distance based approach. The resulting architecture, Transductive CNAPS, is more effective at producing useful estimates of class mean and covariance especially in low-shot settings, when used at test time. Even though we demonstrate the efficacy of our approach in the transductive domain where query examples themselves are used as unlabelled data, our soft k-means clustering procedure naturally extends to use other sources of unlabelled examples in a semi-supervised fashion.

Transductive CNAPS superficially resembles a transductive GMM stacked on top of a learned feature representation; however, when we try to make this connection exact (by including the log-determinant of the class covariances), we suffer substantial performance hits. Explaining why this happens will be the subject of future work.

6. Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Research Chairs (CRC) Program, the Canada CIFAR AI Chairs Program, Compute Canada, Intel, Inverted AI under their MITAC Accelerate internship program and DARPA under its D3M and LWLL programs. Additionally, this material is based upon work supported by the United States Air Force under Contract No. FA8750-19-C-0515.

References

[1] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6(Oct):1705–1749, 2005.

[2] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

[4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613, 2014.

[5] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Selecting relevant features from a multi-domain representation for few-shot classification, 2020.

[6] Abdur R Feyjie, Reza Azad, Marco Pedersoli, Claude Kauffman, Ismail Ben Ayed, and Jose Dolz. Semi-supervised few-shot learning for medical image segmentation, 2020.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

[8] Marta Garnelo, Dan Rosenbaum, Chris J. Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J. Rezende, and S. M. Ali Eslami. Conditional neural processes. *International Conference on Machine Learning (ICML)*, 2018.

[9] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, abs/1804.09458, 2018.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680, 2014.

[11] Grigorii Guz, Peyman Bateni, Darius Muglich, and Giuseppe Carenini. Neural RST-based evaluation of discourse coherence. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 664–671, 2020.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] MD. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Comput. Surv.*, 51(6), Feb. 2019.

[15] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *International Joint Conf. on Neural Networks (IJCNN)*, pages 1–8, 2013.

[16] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *CoRR*, abs/1907.09408, 2019.

[17] Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. The quick, draw!-ai experiment.(2016), 2016.

[18] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo. Edge-labeling graph neural network for few-shot learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[19] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 2015.

[20] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 2012.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[23] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[24] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[25] Moshe Lichtenstein, Prasanna Sattigeri, Rogério Feris, Raja Giryes, and Leonid Karlinsky. TAFSSL: task-adaptive feature sub-space learning for few-shot classification. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VII*, volume 12352 of *Lecture Notes in Computer Science*, pages 522–539. Springer, 2020.

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[27] Lu Liu, William Hamilton, Guodong Long, Jing Jiang, and Hugo Larochelle. A universal representation transformer layer for few-shot image classification, 2020.

[28] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. *International Conference on Learning Representations (ICLR)*, 2019.

[29] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[30] Igor Melnykov and Volodymyr Melnykov. On k-means algorithm with the use of mahalanobis distances. *Statistics & Probability Letters*, 84:88–95, 2014.

[31] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. Meta-learning with temporal convolutions. *CoRR*, abs/1707.03141, 2017.

[32] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018.

[33] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *IEEE Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008.

[34] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 721–731. 2018.

[35] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[36] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[37] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[38] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[39] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. *International Conference on Learning Representations (ICLR)*, 2018.

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39, June 2015.

[41] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. *Advances in Neural Information Processing Systems*, 2019.

[42] 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. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[43] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *International Conference on Learning Representations (ICLR)*, 2018.

[44] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[45] B. Schroeder and Y Cui. Fgvcx fungi classification challenge 2018, 2018.

[46] Adam Scibior, Vasileios Lioutas, Daniele Reda, Peyman Bateni, and Frank Wood. Imagining the road ahead: Multi-agent trajectory prediction via differentiable simulation, 2021.

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

[48] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

[49] Xibin Song, Yuchao Dai, Dingfu Zhou, Liu Liu, Wei Li, Hongdong Li, and Ruigang Yang. Channel attention based iterative residual learning for depth map super-resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[50] M. Sornam, K. Muthusubash, and V. Vanitha. A survey on image classification and activity recognition using deep convolutional neural network architecture. In *International Conference on Advanced Computing (ICoAC)*, 2017.

[51] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[52] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[53] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. *International Conference on Learning Representations (ICLR)*, 2020.

[54] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[55] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[56] Wei Wang, Vincent W. Zheng, Han Yu, and Chunyan Miao. A survey of zero-shot learning: Settings, methods, and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2):13:1–13:37, Jan. 2019.

[57] Yaqing Wang and Quanming Yao. Few-shot learning: A survey. *CoRR*, abs/1904.05046, 2019.

[58] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 2014.

[59] Yang Zhao, Chunyuan Li, Ping Yu, and Changyou Chen. Remp: Rectified metric propagation for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2581–2590, June 2021.