# RepMet: Representative-based metric learning for classification and few-shot object detection

Leonid Karlinsky*, Joseph Shtok*, Sivan Harary*, Eli Schwartz*, Amit Aides, Rogerio Feris
IBM Research AI

Raja Giryes
Tel-Aviv University

Alex M. Bronstein
Technion

## Abstract

*Distance metric learning (DML) has been successfully applied to object classification, both in the standard regime of rich training data and in the few-shot scenario, where each category is represented by only a few examples. In this work, we propose a new method for DML that simultaneously learns the backbone network parameters, the embedding space, and the multi-modal distribution of each of the training categories in that space, in a single end-to-end training process. Our approach outperforms state-of-the-art methods for DML-based object classification on a variety of standard fine-grained datasets. Furthermore, we demonstrate the effectiveness of our approach on the problem of few-shot object detection, by incorporating the proposed DML architecture as a classification head into a standard object detection model. We achieve the best results on the ImageNet-LOC dataset compared to strong baselines, when only a few training examples are available. We also offer the community a new episodic benchmark based on the ImageNet dataset for the few-shot object detection task.*

## 1. Introduction

Due to the great success of deep neural networks (DNNs) in the tasks of image classification and detection [7, 11, 12, 14, 32, 45], they are now widely accepted as the 'feature extractors of choice' for almost all computer vision applications, mainly for their ability to learn good features from the data. It is well-known that training a regular DNN model from scratch requires a significant amount of training data [26]. Yet, in many practical applications, one may be given only a few training samples per class to learn a classifier. This is known as the few-shot learning problem.

Recent studies have achieved significant advances in using DNNs for few-shot learning. This has been demonstrated for domain-specific tasks, such as face recognition [28] and for the classification of general categories [6, 10, 33, 38, 40, 44]. However, very few works have investigated the problem of few-shot object *detection*, where the task of recognizing instances of a category, represented by a few examples, is complicated by the presence of the image background and the need to accurately localize the objects. Recently, several interesting papers demonstrated preliminary results for the zero-shot object detection case [1, 23] and for the few-shot transfer learning [5] scenario.

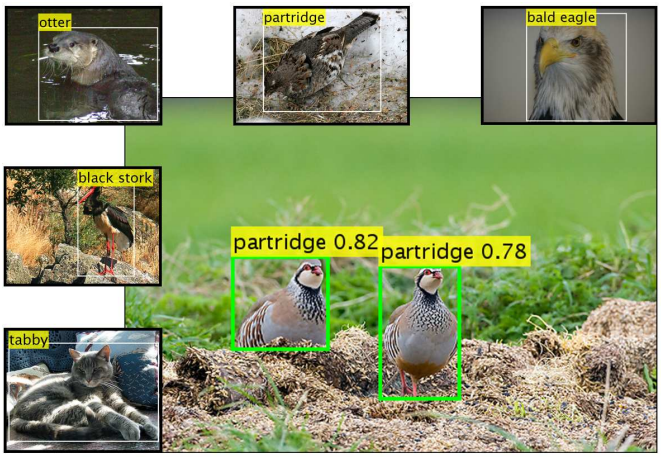


Figure 1. **One-shot detection example**. Surrounding images: examples of new categories unseen in training. Center image: detection result for the one-shot detector on an image containing instances of partridge, which is one of the new categories.

In this work, we propose a novel approach for Distance Metric Learning (DML) and demonstrate its effectiveness on both few-shot object detection and object classification. We represent each class by a mixture model with multiple modes, and consider the centers of these modes as the *representative* vectors for the class. Unlike previous methods, we *simultaneously* learn the embedding space, backbone network parameters, and the representative vectors of the training categories, in a single end-to-end training process.

For few-shot object detection, we build upon modern approaches (e.g., the deformable-FPN variant of the Faster-

*The authors have contributed equally to this work

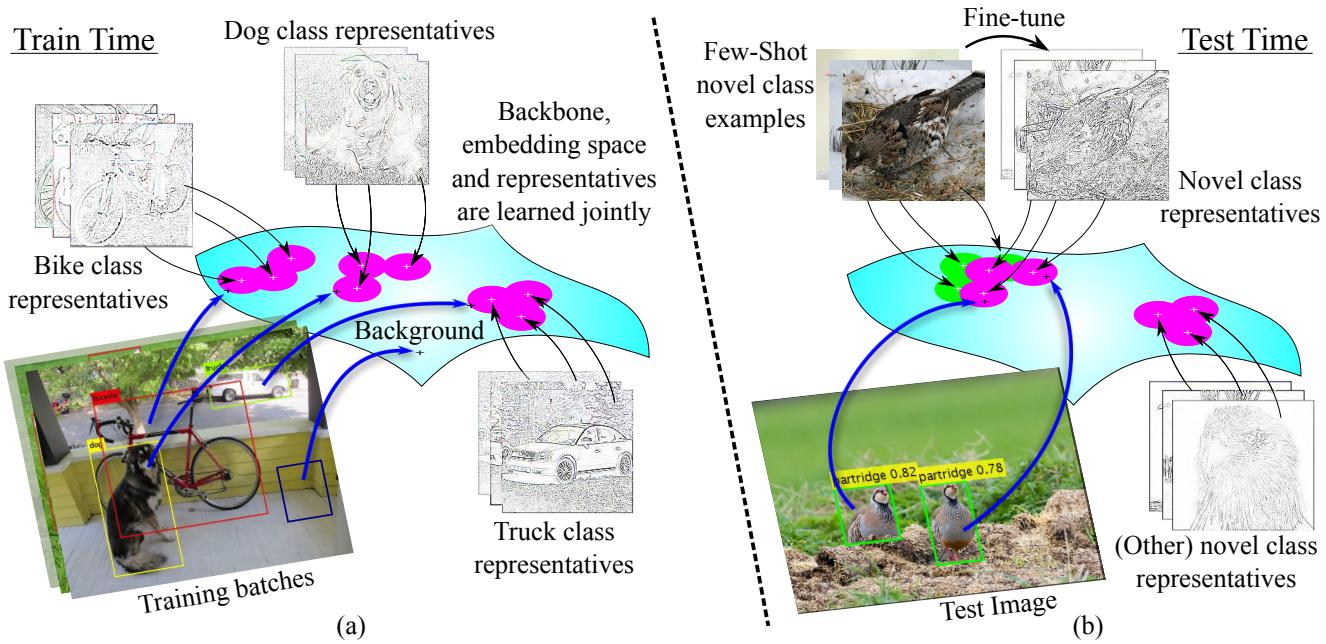


Figure 2. **Overview of our approach.** (a) *Train time:* backbone, embedding space and mixture models for the classes are learned jointly, class representatives are mixture mode centers in the embedding space; (b) *Test time:* new (unseen during training) classes are introduced to the detector in the learned embedding space using just one or a few examples. Fine tuning the representatives and the embedding (on the episode train data) can be used to further improve performance (Section 5). For brevity, only two novel classes are illustrated in the test. The class posteriors are computed by measuring the distances of the input features to the representatives of each of the classes.

RCNN [7, 11]) that rely on a Region Proposal Network (RPN) to generate regions of interest, and a classifier 'head' that classifies these ROIs into one of the object categories or a background region. In order to learn a robust detector with just a few training examples (see Figure 1 for a one-shot detection example), we propose to replace the classifier head with a subnet that learns to compute class posteriors for each ROI, using our proposed DML approach. The input to this subnet are the feature vectors pooled from the ROIs, and the class posteriors for a given ROI are computed by comparing its embedding vector to the set of representatives for each category. The detection task requires solving 'an open set recognition problem', namely to classify ROIs into both the structured foreground categories and the unstructured background category. In this context, the joint end-to-end training is important, since sampling background ROIs for separate training of the DML is very inefficient (Section 5).

In the few-shot detection experiments, we introduce *new categories* into the detector. This is done by replacing the learned representatives (corresponding to old categories) with embedding vectors computed from the foreground ROIs of the few training examples given for these categories ($k$ examples for $k$-shot detection). We also investigate the effects of fine-tuning our proposed model and the baselines for few-shot learning. Promising results, compared to baselines and the previous work, are reported on the few-shot detection task (Section 5.2) underlining the effectiveness of jointly optimizing the backbone and the embedding for DML. Figure 2 schematically illustrates an overview of our approach to few-shot detection.

We also demonstrate the use of our approach for general DML-based classification by comparing to the Magnet Loss [25] and other state-of-the-art DML-based approaches [43, 22]. Instead of the alternating training of embedding and clustering used in [25], our proposed approach end-to-end trains a single (monolithic) network architecture capable of learning the DML embedding together with the representatives (modes of the mixture distributions). Effectively, this brings the clustering inside the end-to-end network training. Using this method, we were able to improve upon the state-of-the-art classification results of [22, 25, 43] on a variety of fine-grained classification datasets (Sec. 5.1).

Our contributions are threefold. **First**, we propose a novel sub-net architecture for jointly training an embedding space together with the set of mixture distributions in this space, having one (multi-modal) mixture for each of the categories. This architecture is shown to improve the current state of the art for both DML-based object classification and few-shot object detection. **Second**, we propose a method to equip an object detector with a DML classifier head that can admit new categories, and thus transform it into a few-shot detector. To the best of our knowledge, this has not been done before. This is probably due to detector training batches being usually limited to one image per-GPU,

not allowing for batch control in terms of category content. This control is needed by any of the current few-shot learners that use episode-based training. This, in turn, makes it challenging to use those approaches within an end-to-end trained detector. In our approach, the set of representatives serves as an 'internal memory' to pass information between training batches. **Third**, in the few-shot classification literature, it is a common practice to evaluate the approaches by averaging the performance on multiple instances of the few-shot task, called episodes. We offer such an episodic benchmark for the few-shot detection problem, built on a challenging fine-grained few-shot detection task.

## 2. Related work

**Distance Metric Learning.** The use of metric learning for computer vision tasks has a long history (see [15] for a survey). In a growing body of work, the methods for image classification and retrieval, based on deep DML, have achieved state-of-the-art results on various tasks [22, 25, 34, 43]. Rippel et al. [25] showed that if the embedding and clustering of the category instances are alternated during training, then on a variety of challenging fine-grained datasets [13, 20, 21, 27] the DML-based classification improves the state-of-the-art even with respect to the non-DML methods. In DML, the metric being learned is usually implemented as an $L_2$ distance between the samples in an embedding space generated by a neural network. The basic loss function for training such an embedding is the triplet loss [41], or one of its recent generalizations [34, 35, 39]. These losses are designed to make the embedding space semantically meaningful, such that objects from the same category are close under the $L_2$ distance, and objects from different categories are far apart. This makes DML a natural choice for few-shot visual recognition. Following the DML, a discriminative class posterior is computed at test time. To that end, a non-parametric approach such as $k$-Nearest-Neighbors ($k$-NN) is commonly used to model the class distributions in the learned embedding space [38, 33, 41], though in some cases parametric models are also used [4]. In addition, in many approaches such as [33, 41] there is an inherent assumption of the category distributions being uni-modal in the embedding space. Our approach instead learns a multi-modal mixture for each category, while *simultaneously* learning the embedding space in which the distances to these representatives are computed.

**Few-shot Learning.** An important recent work in few-shot classification has introduced Matching Networks [38], where both train and test data are organized in 'episodes'. An $N$-way, $M$-shot episode is an instance of the few-shot task represented by a set of $M$ training examples from each of the $N$ categories, and one query image of an object from one of the categories. The goal is to determine the correct category for the query. In [38], the algorithm learns to produce a dedicated DML embedding specific to the episode. In [33], each class is represented by a Prototype - a centroid of the batch elements corresponding to that category. Recently, even more compelling results were obtained on the standard few-shot classification benchmarks using meta-learning methods [9, 19, 24, 44] and synthesis methods [6, 10, 29, 40, 44]. Although great progress was made towards few-shot classification, it is still difficult to apply these methods to few-shot detection. The reason is that a detector training batch typically consists of just one image, with a highly unbalanced foreground to background ROI ratio (somewhat balanced using OHEM [31] and alike). This is problematic for existing few-shot learners, which usually require a balanced set of examples from multiple categories in each batch and commonly have difficulty coping with unstructured noise (background ROIs in our case).

There are only a handful of existing works on few-shot detection. An interesting recent work by Chen at al. [5] proposed using regularized fine-tuning on the few given examples in order to transfer a pre-trained detector to the few-shot task. The authors show that using their proposed regularization, fine-tuning of the standard detectors, such as FRCNN [30] and SSD [18], can be significantly improved in the few-shot training scenario. A different approach by Dong et al. [8] uses additional unlabeled data in a semi-supervised setting. By using the classical method of enriching the training data with high-confidence sample selection, the method of [8] produces results comparable to weakly supervised methods with lots of training examples. Unlike previous methods, we propose a DML-based approach for few-shot object detection, which yields superior performance compared to existing techniques.

## 3. RepMet Architecture

We propose a subnet architecture and corresponding losses that allow us to train a DML embedding jointly with the multi-modal mixture distribution used for computing the class posterior in the resulting embedding space. This subnet then becomes a DML-based classifier head, which can be attached on top of a classification or a detection backbone. It is important to note that our DML-subnet is trained jointly with the feature producing backbone. The architecture of the proposed subnet is depicted in Figure 3.

Batch-training is used, but for simplicity we will refer to the input of the subnet as a single (pooled) feature vector $X \in \mathbb{R}^f$ computed by the backbone for the given image (or ROI). Examples for a backbone are InceptionV3 [36] or an FPN [16] (without RCNN). We first employ a DML embedding module, which consists of a few fully connected (FC) layers with batch normalization (BN) and ReLU nonlinearity (2-3 such layers in our experiments). The output of the embedding module is a vector $E = E(X) \in \mathbb{R}^e$, where commonly embedding size $e \ll f$. As an additional set of trained parameters, we hold a set of 'representatives'

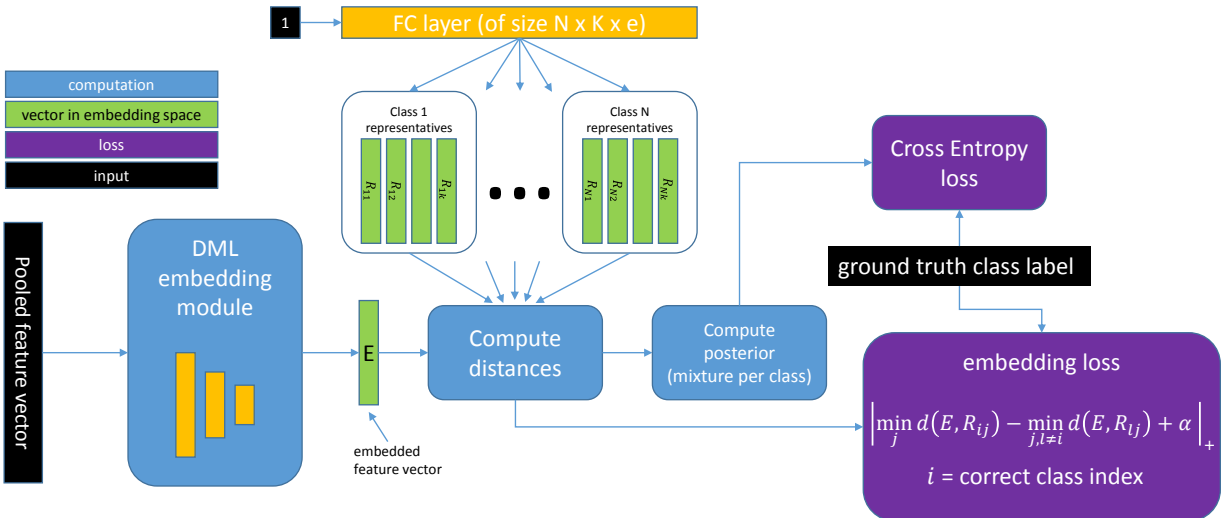


Figure 3. **The proposed RepMet DML sub-net architecture** performs joint end-to-end training of the DML embedding together with the modes of the class posterior distribution. For the detailed description of the notation and the different components please refer to section 3.

$R_{ij} \in \mathbb{R}^e$. Each vector $R_{ij}$ represents the center of the $j$-th mode of the learned discriminative mixture distribution in the embedding space, for the $i$-th class out of the total of $N$ classes. We assume a fixed number of $K$ modes (peaks) in the distribution of each class, so $1 \leq j \leq K$.

In our implementation, the representatives are realized as weights of an FC layer of size $N \cdot K \cdot e$ receiving a fixed scalar input 1. The output of this layer is reshaped to an $N \times K \times e$ tensor. During training, this simple construction flows the gradients to the weights of the FC layer and learns the representatives. For a given image (or detector ROI) and its corresponding embedding vector $E$, our network computes a matrix of $N \times K$ distances $d_{ij}(E) = d(E, R_{ij})$ between $E$ and the representatives $R_{ij}$. These distances are used to compute the probability of the given image (or ROI) in each mode $j$ of each class $i$:

$$p_{ij}(E) \propto \exp\left(-\frac{d_{ij}^2(E)}{2\sigma^2}\right) \tag{1}$$

Here we assume that all the class distributions are mixtures of isotropic multi-variate Gaussians with variance $\sigma^2$. In our current implementation, we do not learn the mixing coefficients and set the discriminative class posterior to be:

$$\mathbb{P}(\mathcal{C} = i|X) = \mathbb{P}(\mathcal{C} = i|E) \equiv \max_{j=1,\ldots,K} p_{ij}(E) \tag{2}$$

where $\mathcal{C} = i$ denotes class $i$ and the maximum is taken over all the modes of its mixture. This conditional probability is an upper bound on the actual class posterior. The reason for using this approximation is that for one-shot detection, at test time, the representatives are replaced with embedded examples of novel classes, unseen during training (more details are found in Section 5). Mixture coefficients are associated with specific modes, and since the modes change at test time, learning the mixture coefficients becomes highly non-trivial. Therefore, the use of the upper bound in Eq. 2 eliminates the need to estimate the mixture coefficients. An interesting future extension to our approach would be to predict the mixture coefficients and the covariance of the modes as a function of $E$ or $X$.

Having computed the class posterior, we also estimate a (discriminative) posterior for the 'open' background ($\mathcal{B}$) class. Following [2], we do not model the background probability, but instead it is estimated via its lower bound using the foreground (class) probabilities:

$$\mathbb{P}(\mathcal{B}|X) = \mathbb{P}(\mathcal{B}|E) = 1 - \max_{ij} p_{ij}(E) \tag{3}$$

Having $\mathbb{P}(\mathcal{C} = i|X)$ and $\mathbb{P}(\mathcal{B}|X)$ computed in the network, we use a sum of two losses to train our model (DML subnet + backbone). The first loss is the regular cross-entropy (CE) with the ground truth labels given for the image (or ROI) corresponding to $X$. The other is intended to ensure there is at least $\alpha$ margin between the distance of $E$ to the closest representative of the correct class, and the distance of $E$ to the closest representative of a wrong class:

$$L(E,R) = \left|\min_j d_{i^*j}(E) - \min_{j,i\neq i^*} d_{ij}(E) + \alpha\right|_+ \tag{4}$$

where $i^*$ is the correct class index for the current example and $|\cdot|_+$ is the ReLU function. Figure 4 illustrates how the proposed DML sub-net is integrated within the full network architectures used for the DML-based classification and the few-shot detection experiments.

## 4. Implementation details

In this section we list additional details of our implementation of the proposed approach for the DML-based classification (Section 4.1) and few-shot detection (Section 4.2) tasks. Our code is available here.

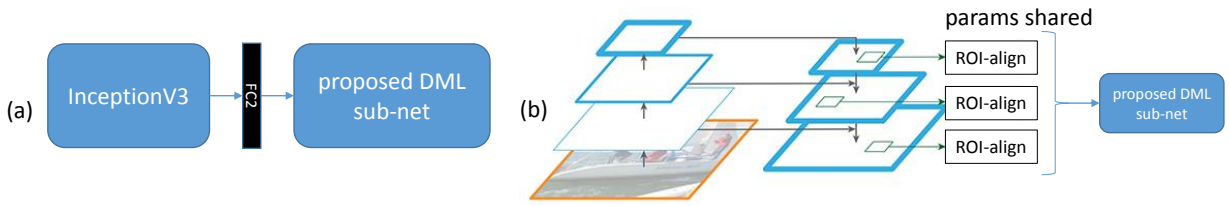


Figure 4. **Network architectures used.** (a) Network for DML based classification. (b) Network for few-shot detection; its backbone is FPN+DCN with deformable ROI-align [7].

### 4.1. DML-based classification

For the DML-based classification experiments, we used the InceptionV3 [36] backbone, attaching the proposed DML subnet to the layer before its last FC layer. The embedding module of the subnet consists of two FC layers of sizes 2048 and 1024, the first with BN and ReLU, and the second just with linear activation. This is followed by an $L_2$ normalization of the embedding vectors. All layers are initialized randomly. In all of our DML-based classification experiments, we set $\sigma = 0.5$ and use $K = 3$ representatives per category. Varying of $K$ on the validation set (Fig. 5(d)) shows that going from $K = 1$ to $K = 3$ improves accuracy by $15\%$ and for $K > 5$ accuracy degrades gracefully by $5\%$. Learning optimal $K$ for each category is an interesting future direction. Each training batch was constructed by randomly sampling $M = 12$ categories and sampling $D = 4$ random instances from each of those categories.

In our DML-based classification experiments on standard benchmarks, there is no background category $\mathcal{B}$, hence we do not need our class mixtures to handle points that are outliers to all of the mixtures. Therefore, we resort to a more classical mixture model variant with equaly weighted modes, replacing the class posterior in Eq. 2 with its softer normalized version, which we have experimentally verified as more beneficial for DML-based classification:

$$\mathbb{P}(\mathcal{C} = i|X) = \mathbb{P}(\mathcal{C} = i|E) = \frac{\sum_{j=1}^{K} p_{ij}(E)}{\sum_{i=1}^{N}\sum_{j=1}^{K} p_{ij}(E)} \qquad (5)$$

### 4.2. DML-based few-shot detection

For few-shot detection, we used our DML sub-net instead of the RCNN (the classification 'head') on top of the FPN backbone [16] in its Deformable Convolutions (DCN) variant [7]. Our code is based on the original MXNet implemetation of [7]. The backbone was pre-trained on MS-COCO [17]. Our DML subnet, including the representatives, was initialized randomly. The entire network was trained end-to-end using OHEM [31] and SoftNMS [3]. The embedding module in the DML subnet for one-shot detection consisted of two FC layers of width 1024 with BN and ReLU, and a final FC layer of width 256 with linear activation, followed by $L_2$ normalization. We trained using $K = 5$ representatives per class, and $\sigma = 0.5$. Figure 6(d) shows examples of the learned representatives. As in [7], each training batch contained one random training image.

## 5. Results

We have evaluated the utility of our proposed DML subnet on a series of classification and few-shot detection tasks.

### 5.1. DML-based classification

**Fine-grained classification.** We tested our approach on a set of fine-grained classification datasets, widely adopted in the state-of-the-art DML classification works [22, 25, 43]: Stanford Dogs [13], Oxford-IIIT Pet [21], Oxford 102 Flowers [20], and ImageNet Attributes [25]. The results reported in Table 1 show that our approach outperforms the state-of-the-art *DML* classification methods [22, 25, 43] on all datasets[1] except Oxford Flowers. Figure 5 shows the evolution of the t-SNE [37] plot of the training instances in the embedding space over the training iterations.

**Attribute distribution.** We verified that following DML training for classification, images with similar attributes are closer to each other in the embedding space (even though attribute annotations were not used during training). We used the same experimental protocol as [25]. Specifically, we trained our DML classifier on the ImageNet Attributes dataset defined in [25], which contains 116236 images from 90 classes. Next, we measured the attribute distribution on the Object Attributes dataset [27], which provides 25 attributes annotations for about 25 images per class for these 90 classes. For each image in this dataset, and for each attribute, we compute the fraction of neighbors also featuring this attribute, over different neighborhood cardinalities. Figure 6(a) shows improved results obtained by our approach as compared to [25] and to other methods.

**Hyperparameter robustness – ablation study.** We evaluated different values of representatives per class ($1 \leq K \leq 8$), and 9 different architectures of the embedding network (varying the number of dense layers between 1 and 3 and using three different widths for each). Same robustness tests were also repeated for our implementation (reproducing the results) of [25] (original code is not available).

[1] Non-DML [42] achieves 3.3% error on Stn.-Dogs using external data

| | method | | | |
|---|---|---|---|---|
| dataset | MsML [22] | Magnet [25] | VMF [43] | Ours |
| Stanford Dogs | 29.7 | 24.9 | 24.0 | **13.7** |
| Oxford Flowers | 10.5 | 8.6 | **4.4** | 11 |
| Oxford Pet | 18.8 | 10.6 | 9.9 | **6.9** |
| ImageNet Attributes | – | 15.9 | — | **13.2** |

Table 1. Comparison of **test error** (in %) with the state-of-the-art DML classifier approaches on different fine-grained classification datasets (lower is better). For our method, same hyper-parameters were used for all datasets. Specific tuning may improve flowers results furhter.

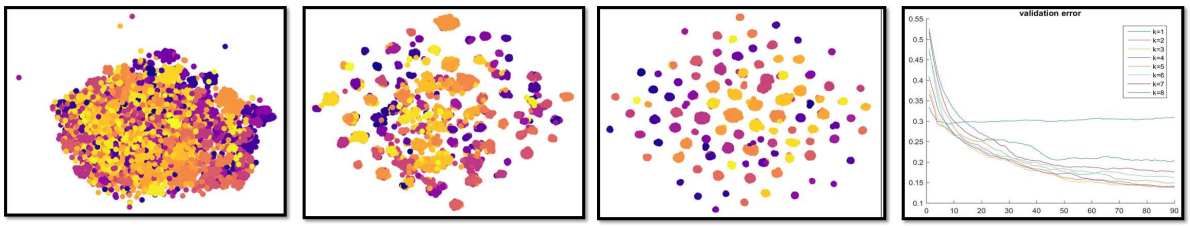


Figure 5. Evolution of the t-SNE visualization of the embedding space while training on the Oxford Flowers. Different colors correspond to different mixture modes. (a) initial; (b) 1200 iterations; (c) 4200 iterations; (d) perfromance for different $K$ = number of representatives

Figures 6(b) and 6(c) show that our method is more robust to hyperparameter changes compared to [25]. We noticed that every time [25] does a k-means step, a significant loss increase occurs causing slower and less stable convergence. In our method this is addressed by doing joint updates to both the embedding and the mixture models.

### 5.2. Few-shot object detection

To the best of our knowledge, the only few-shot detection benchmark available to-date is reported in the LSTD work [5] by Chen et al., who proposed to approach few-shot detection by a regularized fine-tuning. In Table 2, we compare our approach to the results of LSTD [5] on 'Task 1', which is their most challenging ImageNet based 50-way few-shot detection scenario.

| | 1-shot | 5-shot | 10-shot |
|---|---|---|---|
| LSTD [5] | 19.2 | 37.4 | 44.3 |
| ours | **24.1** | **39.6** | **49.2** |

Table 2. Comparison to LSTD [5] on their Task 1 experiment: 50-way detection on 50 ImageNet categories (as mAP %).

Since for all of their proposed tasks, the benchmarks of [5] consist of just one episode (train/test images selection) per task, we created an additional benchmark for few-shot detection. Our proposed benchmark is based on ImageNet-LOC data. The benchmark contains multiple random episodes (instances of the few-shot detection tasks); we used 500 random episodes in our benchmark. This format is borrowed from the few-shot classification literature. Each episode, for the case of the $n$-shot, $m$-way few-shot detection task, contains random $n$ training examples for each of the $m$ randomly chosen classes, and $10 \cdot m$ random query images containing one or more instances belonging to these classes (thus at least 10 instances per class). The goal is to detect and correctly classify these instances. For consistency, for each $n \in \{1, 5, 10\}$ the same 500 random episodes are used in all of the $n$-shot experiments. Please see Figure 1 for an illustration of a 1-shot, 5-way episode.

On the proposed few-shot detection benchmark, we have compared our approach to three baselines. For the first, denoted as **'baseline-FT'**, we fine-tune a standard detector network on just the few $(n \cdot m)$ available samples of the $(m)$ novel categories in each ($n$-shot, $m$-way) test episode. Specifically, we fine-tuned the linear decision layer of the classifier head of the FPN-DCN detector [7], the same detector we use as a backbone for our approach. For the second baseline, denoted as **'baseline-DML'**, we attach our DML sub-net without the embedding module to the regular (pre-trained) FPN-DCN detector, effectively using the FPN-DCN two last FC layers as the embedding module. The FPN-DCN detector used for this baseline is pre-trained as a regular FPN-DCN on the same data as our approach, hence without being optimized for DML based classification as opposed to our full method. For the third baseline, denoted as **'baseline-DML-external'**, we trained the DML sub-net embedding module separately from the detector, in an offline training process. The embedding was trained on sampled foreground and background ROIs using the triplet loss [41]. Training the embedding using Prototypical Networks [33] obtained similar performance for this baseline.

All the baselines were pre-trained on the same training set as our model and tested on the same collections

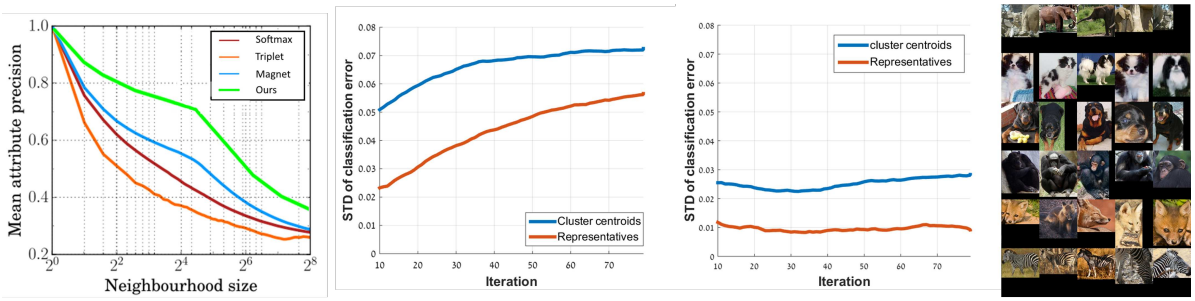


Figure 6. (a) Mean attribute precision as a function of neighborhood size on the ImageNet Attributes dataset. The 'Softmax', 'Triplet' and 'Magnet' graphs are borrowed from [25]. (b) Testing performance stability to hyperparameter change of our method and the Magnet loss [25]. We plot the STD of the classification error, measured accross various depth and width sizes of the embedding model, as a function of the iteration number. Lower is better. (c) same as (b) for various number of modes in the learned mixture. (d) Examples of learned representatives, for each representative train RPN crop with the closest embedding is shown (please use zoom).

| | | no episode fine-tuning | | | with episode fine-tuning | | |
|---|---|---|---|---|---|---|---|
| dataset | method | 1-shot | 5-shot | 10-shot | 1-shot | 5-shot | 10-shot |
| **ImageNet-LOC** (214 unseen animal classes) | baseline-FT (FPN-DCN [7]) | — | — | — | 35.0 | 51.0 | 59.7 |
| | baseline-DML | 41.3 | 58.2 | 61.6 | 41.3 | 59.7 | 66.5 |
| | baseline-DML-external | 19.0 | 30.2 | 30.4 | 32.1 | 37.2 | 38.1 |
| | Ours | **56.9** | **68.8** | **71.5** | **59.2** | **73.9** | **79.2** |
| **ImageNet-LOC** (100 seen animal classes) | Ours - trained representatives | — | 86.3 | — | — | — | — |
| | Ours - episode representatives | 64.5 | 79.4 | 82.6 | — | — | — |

Table 3. Few-shot 5-way detection test performance on ImageNet-LOC. Reported as mAP in %.

or random episodes. To train the models we used the 100 first categories from ImageNet-LOC (mostly animals and birds species). For testing, we used all the remaining 214 ImageNet-LOC animal and bird species categories (unseen at training) to ensure that the train and the test categories belonged to the same concept domain. For our model and all the DML-baselines, in each episode, the set of categories being detected was reset to the $m$ new ones by replacing the set of representatives $R$ in the DML subnet with the embedding vectors computed from the ROIs corresponding to the training objects of the episode. These ROIs were selected among the $2K$ ROIs per image returned by RPN by checking which ROIs passed the IoU$\geq 0.7$ requirement with the training objects bounding boxes. In our approach, the embedding and the backbone are jointly optimized to be used with the representatives-based class posterior. This offers an advantage compared to the baselines, as suggested by the performance comparison reported in Table 3.

The evaluation of our approach and the baselines on the set of unseen classes is reported in Table 3 (in its unseen classes section). The mean average precision (mAP) in % is calculated on 5-way detection tasks (500 such tasks). The mAP is computed by collecting and evaluating jointly (in terms of score threshold for computing precision and recall) the entire set of bounding boxes detected in all the 500 test episodes with 50 query images each.

In addition, for each of the tested methods (ours and the baselines), we repeated the experiments while fine-tuning the last layer of the network just on the episode training images (for our model and the baselines using DML, the last embedding layer and the representatives were fine-tuned). The results with fine-tuning are also reported in Table 3. Figure 7 shows examples of 1-shot detection test results.

From the relatively low performance of 'baseline-DML-external', we can conclude that, as stated in the introduction, joint training of the embedding space with the DML classifier is crucial for the performance. From our close examination, the reduction in mAP of 'baseline-DML-external' is mostly attributed to significantly higher False Positives rates than in the other methods. Although the external embedding was trained on the same training images as our method and the other baselines, it was infeasible to sample the entire collection of possible background ROIs that are being processed by our method when training as a detector end-to-end. Therefore, we had to resort to sampling 200 ROIs per image, which reduced the baseline's ability to reject the background.

To test the inter-dependence of the learned embedding on the specific representatives learned jointly with it during training, we repeated the episode-based testing on the set of classes seen during training (using only validation images not used for training). The results of this evaluation are also reported in Table 3 in the seen classes section. We repeated the seen classes testing twice: once using the rep-

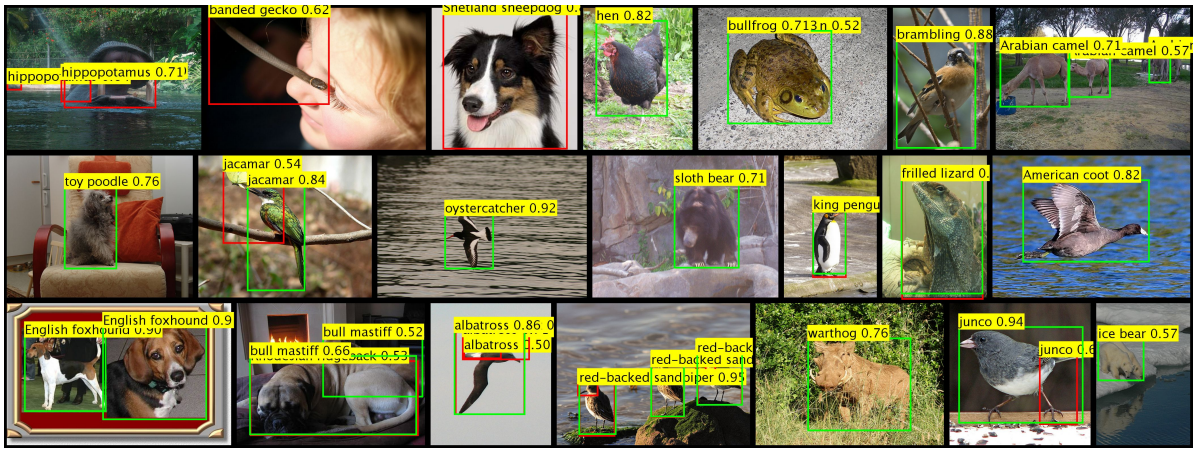


Figure 7. **Example one-shot detection results.** Green frames indicate correctly detected objects and red frames indicate wrong detections. A threshold of 0.5 on the detection score is used throughout. Detections with higher scores are drawn on top of those with lower scores.

resentatives taken from the training objects of each episode (same as for unseen classes) and once using the originally trained representatives (as they correspond to the set of seen classes). Since during training, we learn $K = 5$ representatives per class, we report the result of the second test in the 5-shot column. We can see that (i) the trained representatives perform better than embedding of random class examples, underlining again the benefits of joint training; (ii) the performance drop from trained representatives to random class members is not that big ($\sim 7$ points), hinting that the learned embedding is robust to change of representatives and is likely to perform well on the new unseen categories (as was verified above in our few-shot experiments).

In [1] Recall@100 was used as their performance measure (Recall % taking 100 top detections in each test image). We also implemented this measure in our 1-shot test, achieving 88.2% Recall@100 and 65.9% Recall@10 calculated over our entire set of 500 test episodes. This demonstrates that our approach works well on an individual image basis, and illustrates the importance of considering all the boxes from all the test images simultaneously when computing the AP, as we did in our benchmark.

In order to check if the modification introduced by replacing the RCNN classifier with our DML sub-net hinders the detection performance on the seen classes, we tested the detection performance of our model and the vanilla FPN-DCN model (using their original code) on the validation sets of the 100 first Imagenet-LOC training categories and of PASCAL VOC. As shown in Table 4, our detector is slightly inferior to the original FPN-DCN model on the Pascal VOC, but compares favorably on the 100 first Imagenet-LOC (more fine-grained) categories.

## 6. Summary & Conclusions

In this work, we proposed a new method for DML, achieving state-of-the-art performance for object classification compared to other DML-based approaches. Using this method, we designed one of the first few-shot detection approaches, which compares favorably to the current few-shot detection state-of-the-art. We also proposed a benchmark for the few-shot object detection, based on the Imagenet-LOC dataset, in the hopes that it will encourage researchers to further investigate into this problem, which has so far been almost untouched. Future work directions include predicting the mixing coefficients and covariances for the class mixtures learned within our DML sub-net as a function of the input. High RPN recall is important (for any two stage detector) and is clearly harder to achieve for the few-shot caegories. Additional interesting future direction is using our proposed DML classifer also for the RPN. This will allow improving the RPN sensitivity to the new categories and potentially better handle cases where few-shot categories appear alongside training categories in the intial training. That said, class-agnostic RPN (as in our approach) is trained for 'general objectness' and in many cases its recall is quite high (above 90%) on the unseen categories.

| | PASCAL VOC | | | ImageNet (LOC) | | |
|---|---|---|---|---|---|---|
| method / IoU | 0.7 | 0.5 | 0.3 | 0.7 | 0.5 | 0.3 |
| FPN-DCN [7] | **74.6** | **83.5** | **85.3** | 46.9 | 55.2 | 60.2 |
| Ours | 73.7 | 82.9 | 84.9 | **60.7** | **61.7** | **70.7** |

Table 4. Regular detection performance (in mAP [%]) per different acceptance IoU. FPN-DCN evaluated using their original code.

**Acknowledgments:** This research was partially supported by ERC-StG SPADE PI Giryes, ERC-StG RAPID PI Bronstein, and the European Unions Horizon 2020 research and innovation programme under grant agreement 688930. Rogerio Feris is partly supported by IARPA via DOI/IBC contract number D17PC00341. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government).

## References


[1] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-Shot Object Detection. arXiv:1804.04340, 2018. 1, 8

[2] Abhijit Bendale and Terrance Boult. Towards Open World Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1893–1902, 2015. 4

[3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS —Improving Object Detection With One Line of Code. IEEE International Conference on Computer Vision (ICCV), pages 5562–5570, 2017. 5

[4] Dong Chen, Xudong Cao, Liwei Wang, Fang Wen, and Jian Sun. Bayesian face revisited: A joint formulation. In European Conference on Computer Vision (ECCV), pages 566–579, 2012. 3

[5] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. LSTD: A Low-Shot Transfer Detector for Object Detection. AAAI, 2018. 1, 3, 6

[6] Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. Semantic Feature Augmentation in Few-shot Learning. arXiv:1804.05298v2, 2018. 1, 3

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable Convolutional Networks. The IEEE International Conference on Computer Vision (ICCV), pages 764–773, 2017. 1, 2, 5, 6, 7, 8

[8] Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. Few-shot Object Detection. Arxiv:1706.08249, pages 1–11, 2017. 3

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. arXiv:1703.03400, 2017. 3

[10] Bharath Hariharan and Ross Girshick. Low-shot Visual Recognition by Shrinking and Hallucinating Features. IEEE International Conference on Computer Vision (ICCV), 2017. 1, 3

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. arXiv:1703.06870, 2017. 1, 2

[12] Gao Huang, Zhuang Liu, L v. d. Maaten, and Kilian Q Weinberger. Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, 2017. 1

[13] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel Dataset for Fine-Grained Image Categorization. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop, 2011. 3, 5

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. Advances In Neural Information Processing Systems, pages 1–9, 2012. 1

[15] Brian Kulis. Metric Learning: A Survey. Foundations and Trends in Machine Learning, 5(4):287–364, 2013. 3

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 3, 5

[17] 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 Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 8693 LNCS, pages 740–755, 2014. 5

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9905 LNCS:21–37, 2016. 3

[19] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. Meta-Learning with Temporal Convolutions. arXiv:1707.03141, 2017. 3

[20] Maria Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. Proceedings - 6th Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP, pages 722–729, 2008. 3, 5

[21] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C V Jawahar. Cats and dogs. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3498–3505, 2012. 3, 5

[22] Qi Qian, Rong Jin, Shenghuo Zhu, and Yuanqing Lin. Fine-grained visual categorization via multi-stage metric learning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 07-12-June:3716–3724, 2015. 2, 3, 5, 6

[23] Shafin Rahman, Salman Khan, and Fatih Porikli. Zero-Shot Object Detection: Learning to Simultaneously Recognize and Localize Novel Concepts. arXiv:1803.06049, 2018. 1

[24] Sachin Ravi and Hugo Larochelle. Optimization As a Model for Few-Shot Learning. International Conference on Learning Representations (ICLR), pages 1–11, 2017. 3

[25] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric Learning with Adaptive Density Discrimination. arXiv:1511.05939, pages 1–15, 2015. 2, 3, 5, 6, 7

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. IJCV, 9 2015. 1

[27] Olga Russakovsky and Li Fei-Fei. Attribute learning in large-scale datasets. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6553 LNCS(PART 1):1–14, 2010. 3, 5

[28] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 815–823, 2015. 1

[29] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex M Bronstein. -Encoder: an Effective

Sample Synthesis Method for Few-Shot Object Recognition. NIPS, 2018. 3

[30] Ross Girshick Jian Sun Shaoqing Ren, Kaiming He. Weakly Supervised One-Shot Detection with Attention Siamese Networks. NIPS, pages 1–9, 2015. 3

[31] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training Region-based Object Detectors with Online Hard Example Mining. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 3, 5

[32] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR arXiv:1409.1556, abs/1409.1:1–14, 2014. 1

[33] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical Networks for Few-shot Learning. Advances In Neural Information Processing Systems (NIPS), 2017. 1, 3, 6

[34] Kihyuk Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. Neural Information Processing Systems (NIPS), pages 1–9, 2016. 3

[35] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep Metric Learning via Lifted Structured Feature Embedding. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4004–4012, 2016. 3

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. arXiv:1512.00567, 2015. 3, 5

[37] L J P Van Der Maaten and G E Hinton. Visualizing high-dimensional data using t-sne. Journal of Machine Learning Research, 9:2579–2605, 2008. 5

[38] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. Advances In Neural Information Processing Systems (NIPS), 2016. 1, 3

[39] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep Metric Learning with Angular Loss. In Proceedings of the IEEE International Conference on Computer Vision, volume 2017-Octob, pages 2612–2620, 2017. 3

[40] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-Shot Learning from Imaginary Data. arXiv:1801.05401, 2018. 1, 3

[41] Kilian Q Weinberger and Lawrence K Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. The Journal of Machine Learning Research, 10:207–244, 2009. 3, 6

[42] Yabin Zhang, Hui Tang, and Kui Jia. Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 11212 LNCS, pages 241–256, 7 2018. 5

[43] Xuefei Zhe, Shifeng Chen, and Hong Yan. Directional Statistics-based Deep Metric Learning for Image Classification and Retrieval. arXiv:1802.09662, pages 1–12, 2018. 2, 3, 5, 6

[44] Fengwei Zhou, Bin Wu, and Zhenguo Li. Deep Meta-Learning: Learning to Learn in the Concept Space. arXiv:1802.03596, 2 2018. 1, 3

[45] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning Transferable Architectures for Scalable Image Recognition. arXiv:1707.07012, 2017. 1