Few-Shot Classification with Feature Map Reconstruction Networks

Davis Wertheimer* Luming Tang* Bharath Hariharan
Cornell University
{dww78,lt453,bh497}@cornell.edu

Abstract

In this paper we reformulate few-shot classification as a reconstruction problem in latent space. The ability of the network to reconstruct a query feature map from support features of a given class predicts membership of the query in that class. We introduce a novel mechanism for few-shot classification by regressing directly from support features to query features in closed form, without introducing any new modules or large-scale learnable parameters. The resulting Feature Map Reconstruction Networks are both more performant and computationally efficient than previous approaches. We demonstrate consistent and substantial accuracy gains on four fine-grained benchmarks with varying neural architectures. Our model is also competitive on the non-fine-grained mini-ImageNet and tiered-ImageNet benchmarks with minimal bells and whistles.[1]

1. Introduction

Convolutional neural classifiers have achieved excellent performance in a wide range of settings and benchmarks, but this performance is achieved through large quantities of labeled images from the relevant classes. In practice, such a large quantity of human-annotated images may not always be available for the categories of interest. Instances of relevant classes may be rare in the wild, and identifying them may require expensive expert annotators, limiting the availability of training points and labels respectively. These problems are compounded in settings such as robotics, where a model may need to learn and adapt quickly in deployment, without waiting for offline data collection. Producing a performant classifier in these settings requires a neural network that can rapidly fit novel, possibly unseen classes from a small number of reference images.

A promising approach to this problem of *few-shot classification* is the family of *metric learning* techniques, where the standard parametric linear classifier head is replaced with a class-agnostic distance function. Class membership is determined by distance in latent space from a point or points known to belong to each class. Simple distance functions such as cosine [13, 8] and Euclidean distance [30] lead to surprisingly powerful classifiers, though more complex [29], non-Euclidean [17], and even learned parametric options [32] are possible, and yield sizable gains.

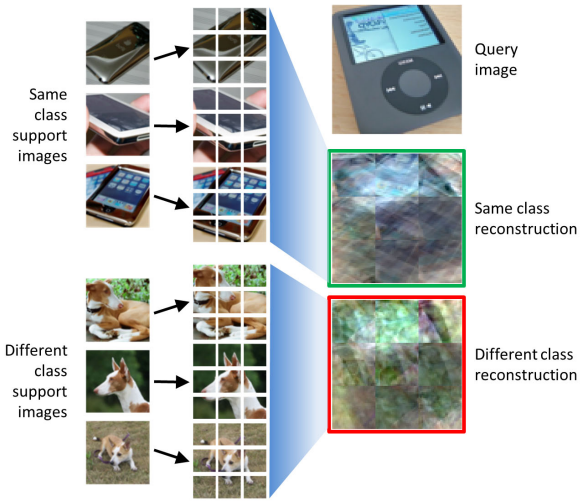

Figure 1. Visual intuition for FRN: we reconstruct each query image as a weighted sum of components from the support images. Reconstructions from the same class are better than reconstructions from different classes, enabling classification. FRN performs the reconstruction in latent space, as opposed to image space, here.

One overarching problem common to all these techniques is the fact that the convolutional feature extractors used to learn the metric spaces produce *feature maps* characterizing appearance at a *grid of spatial locations*, whereas the chosen distance functions require a *single vectorial representation for the entire image*. The researcher must decide how to convert the feature map into a vector representation. Optimally, this conversion would preserve the spatial granularity and detail of the feature map without overfitting to pose, but existing, widely-employed approaches do not accomplish this. Global average-pooling, the standard solution for parametric softmax classifiers, averages information from disparate parts of the image, completely

*Equal contribution
[1]Code is available at https://github.com/Tsingularity/FRN

discarding spatial details that might be necessary for fine distinctions. Flattening the feature map into a single long vector preserves the individual features [30, 32], but also encodes the explicit *location* of each feature. This sensitivity to feature location and arrangement (i.e., object pose), regardless of underlying semantic content, is highly undesirable. Larger and more responsive receptive fields will reduce this sensitivity, but instead overfit to specious cues [9]. We aim to avoid these tradeoffs entirely, preserving spatial detail while disentangling it from location.

We introduce Feature Map Reconstruction Networks (FRN), which accomplish this by framing class membership as a problem of *reconstructing feature maps*. Given a set of images all belonging to a single class, we produce the associated feature maps and collect the component feature vectors *across locations and images* into a single pool of support features. For each query image, we then attempt to reconstruct *every location* in the feature map as a weighted sum of support features, and the negative average squared reconstruction error is used as the class score. Images from the same class should be easier to reconstruct, since their feature maps contain similar embeddings, while images from different classes will be more difficult and produce larger reconstruction errors. By evaluating the reconstruction of the full feature map, FRN preserves the spatial details of appearance. But by allowing this reconstruction to use feature vectors from *any location* in the support images, FRN explicitly discards nuisance location information.

While prior methods based on feature map reconstruction exist, these methods either rely on constrained iterative procedures [43] or large learned attention modules [9, 16]. Instead, we frame feature map reconstruction as a ridge regression problem, allowing us to rapidly calculate a solution in closed form with only a single learned, soft constraint.

The resulting reconstructions are discriminative and semantically rich, making FRN both simpler and more powerful than prior reconstruction-based approaches. We validate these claims by demonstrating across-the-board superiority on four fine-grained few-shot classification datasets (CUB [38], Aircraft [21], meta-iNat and tiered meta-iNat [41]) and two general few-shot recognition benchmarks (mini-ImageNet [37] and tiered-ImageNet [27]). These results hold for both shallow and deep network architectures (Conv-4 [30, 18] and ResNet-12 [14, 18]).

2. Background and Related Work

The few-shot learning setup: Typical few-shot training and evaluation involves sampling task *episodes* from an overarching task distribution – typically, by repeatedly selecting small subsets from a larger set of classes. Images from each class in the episode are partitioned into a small *support set* and a larger *query set*. The number of classes per episode is referred to as the *way*, while the number of support images per class is the *shot*, so that episodes with five classes and one labeled image per class form a "5-way, 1-shot" classification problem. Few-shot classifiers are trained on a large, disjoint set of classes with many labeled images, typically using this same episodic scheme for each batched iteration of SGD. Optimizing the few-shot classifier over the task distribution teaches it to generalize to new tasks from a similar distribution. The classifier learns to learn new tasks, thus episodic few-shot training falls under the umbrella of "meta-learning" or "meta-training".

Prior work in few-shot learning: Existing approaches to few-shot learning can be loosely organized into the following two main-stream families. Optimization-based methods [12, 28, 23] aim to learn a good parameter initialization for the classifier. These learned weights can then be quickly adapted to novel classes using gradient-based optimization on only a few labeled samples. Metric-based methods, on the other hand, aim to learn a task-independent embedding that can generalize to novel categories under a chosen distance metric, such as Euclidean distance [30], cosine distance [13], hyperbolic distance [17], or a distance parameterized by a neural network [32].

As an alternative to the standard meta-learning framework, many recent papers [7, 34, 40] study the performance of standard end-to-end pre-trained classifiers on few-shot tasks. Given minimal modification, these classifiers are actually competitive with or even outperform episodic meta-training methods. Therefore some recent works [43, 42, 8] take advantage of both, and utilize meta-learning after pre-training, further boosting performance.

Few-shot classification through reconstruction: Feature reconstruction is a classic approach [3] to object tracking and alignment [10, 6, 31, 39], but has only recently been utilized for few-shot classification. DeepEMD [43] formulates reconstruction as an optimal transport problem. This formulation is sophisticated and powerful, but training and inference come with significant computational cost, due to the reliance on iterative constrained convex optimization solvers and test-time SGD. CrossTransformer [9] and CrossAttention [16] add attention modules that project query features into the space of support features (or vice versa), and compare the class-conditioned projections to the target to predict class membership. These attention-based approaches introduce many additional learned parameters over and above the network backbone, and place substantial constraints on the projection matrix (weights are non-negative and rows must sum to 1). In contrast, FRN efficiently calculates minimally constrained, least-squares-optimal reconstructions in closed form.

Closed-form solvers in few-shot learning: The use of closed-form solvers for few-shot classification is also not entirely new, though to our knowledge they have not been applied in the explicit context of feature reconstruction. [4]

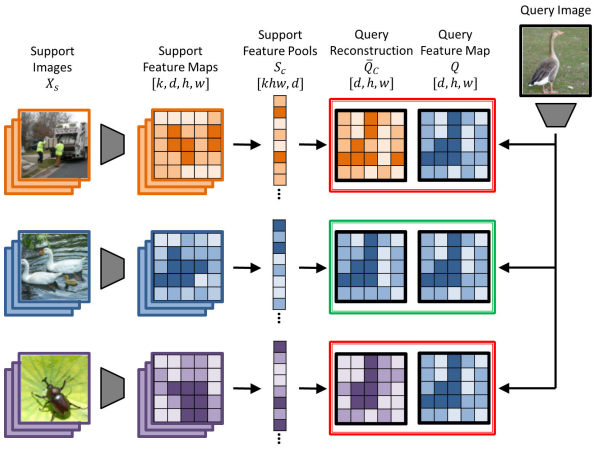

Figure 2. Overview of FRN classification for a k-shot problem. Support images are converted into feature maps (left), which are aggregated into class-conditional pools (middle). The best-fit reconstruction of the query feature map is calculated for each category, and the closest candidate yields the predicted class (right). h, w is feature map resolution and d is the number of channels.

uses ridge regression to map features directly to classification labels, while [32] accomplishes the same mapping with differentiable SVMs. Deep Subspace Networks [29] use the closed-formed projection distance from query embeddings to subspaces spanned by support points as the similarity measure. In contrast, FRN uses closed-form ridge regression to reconstruct entire feature maps, rather than performing direct comparisons between single points in latent space, or regressing directly to class label targets.

3. Method

Feature Map Reconstruction Networks use the quality of query feature map reconstructions from support features as a proxy for class membership. The pool of features associated with each class in the episode is used to calculate a candidate reconstruction, with a better reconstruction indicating higher confidence for the associated class. In this section we describe the reconstruction mechanism of FRN in detail, and derive the closed-form solution used to calculate the reconstruction error and resulting class score. An overview is provided in Fig. 2. We discuss memory-efficient implementations and an optional pre-training scheme, and draw comparisons to prior reconstruction-based approaches.

3.1. Feature Map Ridge Regression

Let X_s denote the set of support images with corresponding class labels in an n-way, k-shot episode. We wish to predict a class label y_q for a single input query image x_q.

The output of the convolutional feature extractor for x_q is a feature map $Q \in \mathbb{R}^{r \times d}$, with r the spatial resolution (height times width) of the feature map, and d the number of channels. For each class $c \in C$, we pool all features from the k support images into a single matrix of support features $S_c \in \mathbb{R}^{kr \times d}$. We then attempt to reconstruct Q as a weighted sum of rows in S_c by finding the matrix $W \in \mathbb{R}^{r \times kr}$ such that $WS_c \approx Q$. Finding the optimal $\bar{W}$ amounts to solving the linear least-squares problem:

$$\bar{W} = \arg\min_{W} \; ||Q - WS_c||^2 + \lambda ||W||^2 \tag{1}$$

where $||\cdot||$ is the Frobenius norm and λ weights the ridge regression penalty term used to ensure tractability when the linear system is over- or under-constrained ($kr \neq d$).

The foremost benefit of the ridge regression formulation is that it admits a widely-known closed-form solution for $\bar{W}$ and the optimal reconstruction $\bar{Q}_c$ as follows:

$$\bar{W} = QS_c^T(S_cS_c^T + \lambda I)^{-1} \tag{2}$$

$$\bar{Q}_c = \bar{W}S_c \tag{3}$$

For a given class c, the negative mean squared Euclidean distance between Q and $\bar{Q}_c$ over all feature map locations yields the scalar probability logit. We also incorporate a learnable temperature factor γ, following [8, 13, 42]. The final predicted probability is thus given by:

$$\langle Q, \bar{Q}_c \rangle = \frac{1}{r}||Q - \bar{Q}_c||^2 \tag{4}$$

$$P(y_q = c|x_q) = \frac{e^{(-\gamma\langle Q, \bar{Q}_c\rangle)}}{\sum_{c' \in C} e^{(-\gamma\langle Q, \bar{Q}_{c'}\rangle)}} \tag{5}$$

We optimize our network by sending the predicted class probabilities for the query images in each episode through a cross-entropy loss, as in standard episodic meta-training. An overview of this process can be found in Fig. 2.

3.2. Learning the Degree of Regularization

The difficulty of the Eq. 1 reconstruction problem varies widely. If $kr > d$, reconstruction may become trivial, as the support features can span the feature space. Conversely, reconstruction is difficult when $d > kr$. To ensure a balanced objective and stable training, we therefore rescale the regularizer λ by $\frac{kr}{d}$. This has the added benefit of making our model somewhat robust to shot, in that concatenating a support pool to itself now yields unchanged reconstructions.

Even with rescaling, though, it is not immediately clear how one should set the regularizer λ. Instead of choosing heuristically, we have the network *learn* λ through meta-learning. This is significant, as it allows the network to pick a degree of regularization such that reconstruction is *discriminative*, rather than strictly least-squares optimal.

Changing λ can have multiple effects. Large λ discourages overreliance on particular weights in W, but also reduces the norm of the reconstruction, increasing reconstruction error and limiting discriminative power. We therefore

disentangle the degree of regularization from the magnitude of $\bar{Q}_c$ by introducing a *learned recalibration term* ρ:

$$\bar{Q}_c = \rho \bar{W} S_c \tag{6}$$

By increasing ρ alongside λ, the network gains the ability to penalize large weights without sending all reconstructions to the origin at the same time. λ and ρ are parameterized as e^{α} and e^{β} to ensure non-negativity, with α and β initialized to zero. Thus, all together, our final prediction is given by:

$$\lambda = \frac{kr}{d} e^{\alpha} \qquad \rho = e^{\beta} \tag{7}$$

$$\bar{Q}_c = \rho \bar{W} S_c = \rho Q S_c^T (S_c S_c^T + \lambda I)^{-1} S_c \tag{8}$$

$$P(y_q = c|x_q) = \frac{e^{(-\gamma \langle Q, \bar{Q}_c \rangle)}}{\sum_{c' \in C} e^{(-\gamma \langle Q, \bar{Q}_{c'} \rangle)}} \tag{9}$$

The model is meta-trained in a similar manner to prior work: sample episodes from a labeled base class dataset and minimize cross entropy on the predicted query labels [30].

Our approach introduces only three learned parameters: α, β and γ. The temperature γ appears in prior work [13, 8, 42]. Ablations on α and β can be found in Sec. 5.1.

3.3. Parallelization

While we have described our approach as finding reconstructions for a single query image, it is relatively straightforward to find the reconstructions for an entire batch of query images. We have already calculated the optimal reconstruction for each of the r feature vectors in Q independently; all we need to do for a batch of b images is to pool the features into a larger matrix $Q' \in \mathbb{R}^{br \times d}$ and run the algorithm as written. Thus for an n-way episode we will only ever need to run the algorithm n times, once for each support matrix S_c, regardless of the quantity or arrangement of queries. These n runs can also be parallelized, given parallel implementations of matrix multiplication and inversion.

3.4. Alternative Formulation

The formula for $\bar{Q}$ in Eq. 8 is efficient to compute when $d > kr$, as the most expensive step is inverting a $kr \times kr$ matrix that does not grow with d. Computing the matrix product from left to right also avoids storing a potentially large $d \times d$ matrix in memory. However, if feature maps are large or the shot number is particularly high ($kr > d$), Eq. 8 may quickly become infeasible to compute. In this case an alternative formulation for $\bar{Q}$ exists, which swaps d for kr in terms of computational requirements. This formulation is owed to the Woodbury Identity [24] as applied in [4]:

$$\bar{Q}_c = \rho \bar{W} S_c = \rho Q (S_c^T S_c + \lambda I)^{-1} S_c^T S_c \tag{10}$$

Here, the most expensive step is a $d \times d$ matrix inversion, and computing the product from right to left avoids storing any large $kr \times kr$ or $br \times kr$ matrices in memory. As r and d are determined by the network architecture, the researcher may employ either formulation depending on k. The network can also decide on the fly at test time. In terms of classifier performance the two formulations are algebraically equivalent, and pseudo-code for both is provided in Supplementary Materials (SM) Sec. 7. For consistency, we employ Eq. 10 in our implementations.

3.5. Auxiliary Loss

In addition to the classification loss, we employ an auxiliary loss that encourages support features from different classes to span the latent space [29]:

$$L_{\text{aux}} = \sum_{i \in C} \sum_{j \in C, j \neq i} ||\hat{S}_i \hat{S}_j^T||^2 \tag{11}$$

where $\hat{S}$ is row-normalized, with features projected to the unit sphere. This loss encourages orthogonality between features from different classes. Similar to [29], we downscale this loss by a factor of 0.03. We use L_{aux} as the auxiliary loss in our subspace network implementation [29], and it replaces the SimCLR episodes in our CrossTransformer implementation [9]. We include it in our own model for consistency, and include an ablation study in Sec. 5.1.

3.6. Pre-Training

Prior work [8, 42] has demonstrated that few-shot classifiers can benefit greatly from non-episodic pre-training. For traditional metric learning based approaches, the feature extractor is initially trained as a linear classifier with global average-pooling on the full set of training classes. The linear layer is subsequently discarded, and the feature extractor is fine-tuned episodically.

This pre-training does not work out-of-the-box for FRN due to its novel classification mechanism. Because the linear classifier uses average-pooling, the feature extractor does not learn spatially distinct feature maps in the way FRN requires (see Sec. 5.1 for analysis).

We therefore devise a new pre-training scheme for FRN. To keep the classifier consistent with FRN meta-training, we continue to use feature reconstruction error as the predicted class logit. Similar to [43], the classification head is parametrized as a set of class-specific *dummy feature maps*, where we introduce a learnable matrix $M_c \in \mathbb{R}^{r \times d}$ for each category c, acting as a proxy for S_c. Following Eq. 10, the prediction for a sample x_q with feature map $Q \in \mathbb{R}^{r \times d}$ is:

$$\bar{Q}_c = \rho Q (M_c^T M_c + \lambda I)^{-1} M_c^T M_c \tag{12}$$

$$P(y_q = c|x_q) = \frac{e^{(-\gamma \langle Q, \bar{Q}_c \rangle)}}{\sum_{c' \in C} e^{(-\gamma \langle Q, \bar{Q}_{c'} \rangle)}} \tag{13}$$

It should be noted that C in this setting is no longer the sampled subset of episode categories, but rather the entire set of

Model	Solver	1-shot Latency	5-shot Latency
DeepEMD [43]	qpth [2]	23,275	>800,000
DeepEMD [43]	OpenCV [5]	178	18,292
FRN (ours)	Eq. 10	**73**	**88**
FRN (ours)	Eq. 8	**63**	**79**

Table 1. Latency (ms) for 5-way mini-ImageNet evaluation with ResNet-12. Detailed discussion and comparison in SM Sec. 10.

training classes (e.g., $|C| = 64$ for mini-ImageNet). We then use this output probability distribution to calculate the standard cross-entropy classification loss. During the pre-training stage, we fix $\alpha = \beta = 0$ but keep γ a learnable parameter. After pre-training is finished, all learned matrices $\{M_c | c \in C\}$ are discarded (similar to the pre-trained MLP classifier in [34, 42, 40, 7, 8]). The pre-trained model size is thus the same as when trained from scratch.

While pre-training is broadly applicable and generally boosts performance, for the sake of fairness we do not pre-train any of our fine-grained experiments, as baseline methods do not consistently pre-train in these settings.

3.7. Relation to Prior Reconstructive Classifiers

DeepEMD [43]: FRN uniquely combines feature map comparison with an unconstrained, closed-form reconstruction objective; prior approaches include one or the other, but not both. Like FRN, DeepEMD solves for a $r \times kr$ reconstruction matrix $\bar{W}$ and uses reconstruction quality (measured as transport cost) as a proxy for class membership. This technique is more sophisticated than ridge regression, but also highly constrained. As a transport matrix, $\bar{W}$ must hold nonnegative values, with rows and columns that sum to 1. More importantly, $\bar{W}$ cannot be calculated in closed form, requiring an iterative solver that can be slow in practice and scales poorly to k greater than one. We found computing the FRN reconstruction orders of magnitude faster than the EMD equivalent (see Table 1). DeepEMD also requires finetuning via back-propagation at test time, whereas our approach scales out of the box to a range of k, r, d.

CrossTransformer (CTX) [9]: CTX and related approaches [16] are more similar to FRN, in that they explicitly produce class-wise linear reconstructions $\bar{Q}_c = \bar{W} S_c$. However, rather than solving for $\bar{W}$, these methods approximate it using attention and extra learned projection layers. CTX reprojects the feature pools S_c and Q into two different "key" and "value" subspaces, yielding S_1, Q_1 and S_2, Q_2. The reconstruction of Q_2 is given by:

$$\bar{Q}_2 = \sigma(\frac{1}{\sqrt{d}} Q_1 S_1^T) S_2 \tag{14}$$

where $\sigma(\cdot)$ denotes a row-wise softmax. While Eq. 14 is loosely analogous to Eq. 8, with the $\sqrt{d}$-scaled softmax replacing the inverted matrix term, we find that performance differs in practice. The CTX layer is also somewhat unwieldy: the two reprojection layers introduce extra parameters into the network, and during training it is necessary to store the $br \times kr$ matrix of attention weights $\sigma(\frac{1}{\sqrt{d}} Q_1 S_1^T)$ for back-propagation. This can lead to a noticeable memory footprint as these values increase – while we did not observe a difference in our experimental settings, simply increasing r from 5×5 to 10×10 in our implementation was sufficient to introduce a 2-3GB overhead (see SM Sec. 10).

Model	Feature map	Regression objective
Proto [30]	×	×
DSN [29]	×	✓
CTX [9]	✓	×
FRN (ours)	✓	✓

Table 2. Relationships between our implemented models.

Deep Subspace Networks (DSN) [29]: DSN predicts class membership by calculating the distance between the query point and its projections onto the latent subspaces formed by the support images for each class. This is analogous to our approach with $r = 1$, with average-pooling performing the spatial reduction. The crucial difference is that DSN assumes (accurately) that $d > k$, whereas in our setting it is not always the case that $d > kr$. In fact, for many of our models S spans the latent space, so the projection interpretation falls apart and we instead rely on the ridge regression regularizer to keep the problem well-posed.

Of the methods that produce explicit reconstructions, CTX compares feature maps while DSN utilizes a closed-form regression objective. FRN captures both concepts, leading to the organization shown in Table 2. We thus re-implement CTX and DSN as direct comparison baselines (details in SM Sec. 9.3). As shown in the following section, FRN leverages a unique synergy between these concepts to improve even when CTX or DSN on their own do not.

4. Experiments

Feature Map Reconstruction Networks focus on spatial details without overfitting to pose, making them particularly powerful in the fine-grained few-shot recognition setting, where details are important and pose is not discriminative. We demonstrate clear superiority on four such benchmarks. For general few-shot learning, FRN with pre-training achieves highly competitive results without extra bells or whistles.

Implementation details: We conduct experiments on two widely used backbones: 4-layer ConvNet (Conv-4) and ResNet-12. Same as [42, 18], Conv-4 consists of 4 consecutive 64-channel convolution blocks that each downsample by a factor of 2. The shape of the output feature maps for input images of size 84×84 is thus 64×5×5. For ResNet-12, we use the same implementation as [42, 34, 18]. The input image size is the same as Conv-4 and the output feature map shape is 640×5×5. During training, we use the

Model	Conv-4		ResNet-12	
	1-shot	5-shot	1-shot	5-shot
MatchNet$^\flat$ [37, 42, 43]	67.73	79.00	71.87	85.08
ProtoNet$^\flat$ [30, 42, 43]	63.73	81.50	66.09	82.50
Hyperbolic [17]	64.02	82.53	-	-
FEAT$^\flat$ [42]	68.87	82.90	-	-
DeepEMD$^\flat$ [43]	-	-	75.65	88.69
ProtoNet† [30]	63.21	83.88	79.09	90.59
DSN† [29]	66.01	85.41	80.80	91.19
CTX† [9]	69.64	87.31	78.47	90.90
FRN (ours)	**73.48**	**88.43**	**83.16**	**92.59**

Table 3. Performance on CUB using bounding-box cropped images as input. $\flat$: use of non-episodic pre-training. Confidence intervals for our implemented models are all below 0.24.

Model	Backbone	1-shot	5-shot
Baseline$^\flat$ [7]	ResNet-18	65.51±0.87	82.85±0.55
Baseline++$^\flat$ [7]	ResNet-18	67.02±0.90	83.58±0.54
MatchNet [7, 37]	ResNet-18	73.49±0.89	84.45±0.58
ProtoNet [7, 30]	ResNet-18	72.99±0.88	86.64±0.51
MAML [7, 12]	ResNet-18	68.42±1.07	83.47±0.62
RelationNet [7, 32]	ResNet-18	68.58±0.94	84.05±0.56
S2M2$^\flat$ [22]	ResNet-18	71.43±0.28	85.55±0.52
Neg-Cosine$^\flat$ [19]	ResNet-18	72.66±0.85	89.40±0.43
Afrasiyabi *et al.*$^\flat$ [1]	ResNet-18	74.22±1.09	88.65±0.55
ProtoNet† [30]	ResNet-12	78.60±0.22	89.73±0.12
DSN† [29]	ResNet-12	79.96±0.21	91.41±0.34
CTX† [9]	ResNet-12	79.34±0.21	91.42±0.11
FRN (ours)	ResNet-12	**83.55±0.19**	**92.92±0.10**

Table 4. Performance on CUB using raw images as input. $\flat$: use of non-episodic pre-training.

standard data augmentation as in [42, 43, 40, 7], which includes random crop, right-left flip and color jitter. Further training details can be found in SM Sec. 9.

Evaluation is performed on standard 5-way, 1-shot and 5-shot settings. Accuracy scores and 95% confidence intervals are obtained over 10,000 trials, as in [42, 8, 40].

4.1. Fine-Grained Few-Shot Classification

For our fine-grained experiments, we re-implement three baselines: Prototypical Networks (ProtoNet†) [30], CTX† [9], and DSN† [29], where † denotes our implementation. For fair comparison, we do not use pre-training for any of our implemented models here, or tune FRN hyperparameters separately from baseline models.

CUB [38] consists of 11,788 images from 200 bird classes. Following [7], we randomly split categories into 100 classes for training, 50 for validation and 50 for evaluation. Our split is identical to [33] (discussion of class splits can be found in SM Sec. 11). Prior work on this benchmark pre-processes the data in different ways: [7] uses raw images as input, while [42, 43] crop each image to a human-annotated bounding box. We experiment on both settings for fair comparison.

Aircraft [21] contains 10,000 images spanning 100 airplane models. Following the same ratio as CUB, we randomly split classes into 50 train, 25 validation and 25 test. Images are pre-cropped to the provided bounding box.

Model	Conv-4		ResNet-12	
	1-shot	5-shot	1-shot	5-shot
ProtoNet† [30]	47.72	69.42	66.57	82.37
DSN† [29]	48.14	66.36	68.16	81.85
CTX† [9]	50.20	67.25	65.60	80.20
FRN (ours)	**53.20**	**71.17**	**70.17**	**83.81**

Table 5. Performance on Aircraft. All 95% confidence intervals are below 0.25.

Model	meta-iNat		tiered meta-iNat	
	1-shot	5-shot	1-shot	5-shot
ProtoNet† [30]	55.34	76.43	34.34	57.13
Covar. pool† [41]	57.15	77.20	36.06	57.48
DSN† [29]	58.08	77.38	36.82	60.11
CTX† [9]	60.03	78.80	36.83	60.84
FRN (ours)	**62.42**	**80.45**	**43.91**	**63.36**

Table 6. Performance on meta-iNat and tiered meta-iNat using Conv-4 backbones. All 95% confidence intervals are below 0.24.

meta-iNat [41, 15] is a benchmark of animal species in the wild. This benchmark is particularly difficult, as classes are unbalanced, distinctions are fine-grained, and images are not cropped or centered, and may contain multiple animal instances. We follow the class split proposed by [41]: of 1135 classes with between 50 and 1000 images, one fifth (227) are assigned to evaluation and the rest to training. While [41] propose a full 227-way, k-shot evaluation scheme with $10 \leq k \leq 200$, we instead perform standard 5-way, 1-shot and 5-shot evaluation, and leave extension to higher shots and unbalanced classes for future work.

tiered meta-iNat [41] represents a more difficult version of meta-iNat where a large domain gap is introduced between train and test classes. The 354 test classes are populated by insects and arachnids, while the remaining 781 classes (mammals, birds, reptiles, etc.) form the training set. Training and evaluation are otherwise the same.

Results on fine-grained benchmarks can be found in Tables 3, 4, 5, and 6, corresponding to cropped CUB, uncropped CUB, Aircraft, and combined meta-iNat and tiered meta-iNat, respectively. FRN is superior across the board, with a notable **2-7 point jump in accuracy** (mean 3.5) from the nearest baseline in all 1-shot settings.

Note that our re-implemented baselines in Tables 3 and 4 are competitive with (and in some cases beat outright) prior published numbers. This shows that in the experiments without prior numbers, our baselines still provide fair competition. We do not give FRN an unfair edge – if anything, our baselines are more competitive, not less.

Based on the above observations, we conclude that FRN is broadly effective at fine-grained few-shot classification.

Model	Backbone	mini-ImageNet 1-shot	mini-ImageNet 5-shot	tiered-ImageNet 1-shot	tiered-ImageNet 5-shot
MatchNet$^{\flat\heartsuit}$ [37, 42, 43]	ResNet-12	65.64±0.20	78.72±0.15	68.50±0.92	80.60±0.71
ProtoNet$^{\flat}$ [30, 42]	ResNet-12	62.39±0.21	80.53±0.14	68.23±0.23	84.03±0.16
MetaOptNet$^{\sharp}$ [18]	ResNet-12	62.64±0.61	78.63±0.46	65.99±0.72	81.56±0.53
Robust 20-distill$^{\flat\sharp\natural}$ [11]	ResNet-18	63.06±0.61	80.63±0.42	65.43±0.21	70.44±0.32
SimpleShot$^{\flat}$ [40]	ResNet-18	62.85±0.20	80.02±0.14	69.09±0.22	84.58±0.16
CAN$^{\flat\heartsuit}$ [16]	ResNet-12	63.85±0.48	79.44±0.34	69.89±0.51	84.23±0.37
S2M2$^{\flat\diamondsuit}$ [22]	ResNet-18	64.06±0.18	80.58±0.12	-	-
Meta-Baseline$^{\flat}$ [8]	ResNet-12	63.17±0.23	79.26±0.17	68.62±0.27	83.29±0.18
GNN+FT$^{\flat\heartsuit\natural}$ [36]	ResNet-10	**66.32±0.80**	81.98±0.55	-	-
DSN‡ [29]	ResNet-12	62.64±0.66	78.83±0.45	66.22±0.75	82.79±0.48
FEAT$^{\flat\heartsuit}$ [42]	ResNet-12	**66.78±0.20**	82.05±0.14	70.80±0.23	84.79±0.16
DeepEMD$^{\flat\diamondsuit\clubsuit}$ [43]	ResNet-12	65.91±0.82	**82.41±0.56**	71.16±0.87	86.03±0.58
Neg-Cosine$^{\flat\diamondsuit\natural}$ [19]	ResNet-12	63.85±0.81	81.57±0.56	-	-
Afrasiyabi *et al.*$^{\flat\heartsuit\diamondsuit\natural}$ [1]	ResNet-18	59.88±0.67	80.35±0.73	69.29±0.56	85.97±0.49
E^{3}BM$^{\flat\heartsuit\diamondsuit}$ [20]	ResNet-12	64.09±0.37	80.29±0.25	71.34±0.41	85.82±0.29
RFS-simple$^{\flat\ddagger}$ [34]	ResNet-12	62.02±0.63	79.64±0.44	69.74±0.72	84.41±0.55
RFS-distill$^{\flat\ddagger\sharp}$ [34]	ResNet-12	64.82±0.60	82.14±0.43	**71.52±0.69**	86.03±0.49
FRN (ours)$^{\flat}$	ResNet-12	66.45±0.19	**82.83±0.13**	71.16±0.22	86.01±0.15
FRN (ours)$^{\flat\clubsuit}$	ResNet-12	-	-	**72.06±0.22**	**86.89±0.14**

Table 7. Performance of selected competitive few-shot models on mini-ImageNet and tiered-ImageNet, ordered chronologically. ‡: use of data augmentation during evaluation; ♯: label smoothing, model ensemble or knowledge distillation; ♡: modules with many additional learnable parameters; ♢: use of SGD during evaluation; ♭: non-episodic pre-training or classifier losses; ♮: network's input resolution is larger than 84. ♣: use of tiered-ImageNet data from DeepEMD's implementation.

4.2. General Few-Shot Classification

We evaluate performance on two standard benchmarks. Compared to direct episodic meta-training from scratch, recent works [8, 34] gain a large advantage from pre-training on all training data and labels, followed by episodic fine-tuning. We follow the framework of [42, 43] and pre-train our model on the entire training set as described in Sec. 3.6.

mini-ImageNet [37] is a subset of ImageNet containing 100 classes in total, with 600 examples per class. Following [25], we split categories into 64 classes for training, 16 for validation and 20 for test.

tiered-ImageNet [27] is a larger subset of ImageNet with 351-97-160 categories for training-validation-testing, respectively. Like tiered meta-iNat, tiered-ImageNet ensures larger domain differences between training and evaluation compared to mini-ImageNet. Most works [40, 34, 8] use images from [27][2] or [18][3], which have 84×84 resolution. DeepEMD [43]'s implementation[4] has 224×224 instead. For fair comparison, we experiment on both settings.

As shown in Table 7, FRN is highly competitive with recent state-of-the-art results. FRN leverages pre-training, but no other extra techniques or tricks. FRN also requires no gradient-based finetuning at inference time, which makes it more efficient than many existing baselines in practice.

[2] https://github.com/renmengye/few-shot-ssl-public

[3] https://github.com/kjunelee/MetaOptNet

[4] https://github.com/icoz69/DeepEMD

Model	Backbone	1-shot	5-shot
MAML$^{\sharp\diamondsuit}$ [12, 7]	ResNet-18	-	51.34±0.72
ProtoNet$^{\sharp}$ [30, 7]	ResNet-18	-	62.02±0.70
Baseline$^{\flat\natural\diamondsuit}$ [7]	ResNet-18	-	65.57±0.70
Baseline++$^{\flat\natural\diamondsuit}$ [7]	ResNet-18	-	62.04±0.76
MetaOptNet$^{\sharp}$ [18, 22]	ResNet-12	44.79±0.75	64.98±0.68
Diverse 20-full$^{\flat\sharp\natural}$ [11]	ResNet-18	-	66.17±0.55
SimpleShot$^{\flat}$ [40, 44]	ResNet-18	48.56	65.63
MatchNet+FT$^{\flat\heartsuit\natural}$ [36]	ResNet-10	36.61±0.53	55.23±0.83
RelationNet+FT$^{\flat\heartsuit\natural}$ [36]	ResNet-10	44.07±0.77	59.46±0.71
GNN+FT$^{\flat\heartsuit\natural}$ [36]	ResNet-10	47.47±0.75	66.98±0.68
Neg-Softmax$^{\flat\diamondsuit\natural}$ [19]	ResNet-18	-	69.30±0.73
Afrasiyabi *et al.*$^{\flat\heartsuit\diamondsuit\natural}$ [1]	ResNet-18	46.85±0.75	70.37±1.02
FRN (ours)$^{\flat}$ on classes from [7]	ResNet-12	**54.11±0.19**	**77.09±0.15**
FRN (ours)$^{\flat}$ on classes from [33]	ResNet-12	**51.60±0.21**	**72.97±0.18**
FRN (ours)$^{\flat}$ on all CUB classes	ResNet-12	**53.39±0.21**	**75.16±0.17**

Table 8. Performance comparison in the cross-domain setting: mini-ImageNet→CUB. Symbols and organization match Table 7.

4.3. Cross-Domain Few-Shot Classification

Finally, we evaluate on the challenging cross-domain setting proposed by [7], where models trained on mini-ImageNet base classes are evaluated on test classes from CUB. We evaluate on three sets of CUB test classes: the split from [7] (common in prior work), the split from [33] (used in Sec. 4.1), and the full set of 200 CUB classes. As shown in Table 8, our FRN model from Sec. 4.2 outperforms previous methods by a wide margin.

training setting	1-shot	5-shot
episodic train from scratch	63.03±0.20	78.01±0.15
after avg-pool pre-train	59.43±0.20	70.88±0.16
episodic finetune	62.13±0.20	76.28±0.15
after FRN pre-train	60.97±0.21	75.11±0.18
episodic finetune	**66.45±0.19**	**82.83±0.13**

Table 9. Impact of pre-training for FRN on mini-ImageNet. Both pre-training and episodic finetuning are important. Classical pre-training with global average-pooling works poorly.

5. Analysis

5.1. Ablation Study

Training shot: Surprisingly, we found that FRN models trained on 1-shot episodes consistently underperform the same models trained with 5-shot episodes, even on 1-shot evaluation. We therefore report the superior numbers from 5-shot models in Sec. 4 and include the 1-shot performance as an ablation in Table 11 of SM. Though clearly worse than the 5-shot counterpart, 1-shot FRN is broadly competitive with the best-performing baselines.

Pre-training: FRN pre-training is crucial for competitive general few-shot performance, especially when compared to pre-trained baselines. However, pre-training alone does not produce a competitive few-shot learner. An FRN trained from scratch outperforms a pre-trained FRN evaluated naively (Table 9, bottom rows). The two-round process of pre-training followed by episodic fine-tuning appears to be crucial. This finding is in line with prior work [42, 8].

Classical pre-training with average-pooling, however, does not produce a viable classifier (Table 9, middle rows). These features are not spatially distinct enough for FRN fine-tuning to recover a meaningful feature space. The resulting classifier is *worse* than one trained from scratch.

Auxiliary loss and λ, ρ regularizers: We ablate these components of both Conv-4 and ResNet-12 models on cropped CUB, with results in Table 12 in SM. The auxiliary loss has little to no impact on FRN performance – we include this loss in our experimental models only for consistent comparisons. Fixing α, β to 0 (and thus λ, ρ to constants) yields mixed results. The 4-layer network clearly benefits from learning both values, but the ResNet-12 architecture does not, likely because the high-dimensional feature space is rich enough to overcome any regularization problems on its own.

5.2. Reconstruction Visualization

While our results suggest that FRN produces more semantically faithful reconstructions from same-class support images than from different classes, we would like to confirm this visually. We therefore train image re-generators for the 5-shot ResNet-12 FRN on CUB and mini-ImageNet, which use an inverted ResNet-12 to map FRN features back to the original image. Training details can be found in SM Sec. 9.4. Results are reported on validation images.

Input	CUB	mini-IN
ground-truth feature map	.208	.177
same-class reconstruction	.343	.307
diff-class reconstruction	.385	.337

Table 10. L2 pixel error between original images and regenerated images from different latent inputs. Results are averaged over 1,000 trials and 95% confidence intervals are below 1e-3.

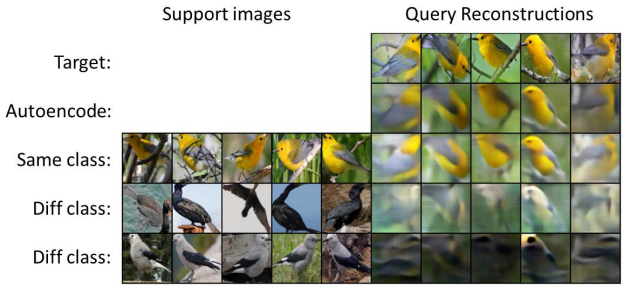

Figure 3. CUB images are regenerated from ground-truth feature maps (row 2), and reconstructions from same-class (row 3) and different-class support images (rows 4, 5). Same-class reconstructions are more faithful to the original. Best viewed digitally.

If same-class feature map reconstructions are more semantically faithful than different-class ones, we should observe a corresponding difference in regenerated image quality. Fig. 3 and Table 10 verify this. Reconstructions from ground-truth features are not particularly good, as classifiers discard class-irrelevant details. However, the increase in ground-truth pixel error relative to these target feature maps is clearly smaller for same-class reconstructions. Additional visualizations are provided in SM Sec. 12. We conclude that FRN reconstructions are semantically faithful for same-class support images and less faithful otherwise.

6. Conclusion

We introduce Feature Map Reconstruction Networks, a novel approach to few-shot classification based on reconstructing query features in latent space. Solving the reconstruction problem in closed form produces a classifier that is both straightforward and powerful, incorporating fine spatial details without overfitting to position or pose. We demonstrate state-of-the-art performance on four fine-grained few-shot classification benchmarks, and highly competitive performance in the general setting.

Acknowledgements : This work was funded by the DARPA Learning with Less Labels program (HR001118S0044). The authors would like to thank Cheng Perng Phoo for valuable suggestions and assistance.

References

[1] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 6, 7

[2] Brandon Amos and J. Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 136–145, 2017. 5, 15

[3] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56:221–255, 2004. 2

[4] Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 2, 4

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 5, 15

[6] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 2887–2894, 2012. 2

[7] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 2, 5, 6, 7, 16

[8] Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[9] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2, 4, 5, 6, 14

[10] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 1078–1085, 2010. 2

[11] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Diversity with cooperation: Ensemble methods for few-shot classification. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 3722–3730, 2019. 7

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, 2017. 2, 6, 7

[13] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4367–4375, 2018. 1, 2, 3, 4, 13, 14

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. 2

[15] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist species classification and detection dataset. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8769–8778, 2018. 6

[16] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4005–4016, 2019. 2, 5, 7

[17] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan V. Oseledets, and Victor S. Lempitsky. Hyperbolic image embeddings. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6417–6427, 2020. 1, 2, 6

[18] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10657–10665, 2019. 2, 5, 7, 12, 13, 14

[19] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 6, 7, 16

[20] Yaoyao Liu, Bernt Schiele, and Qianru Sun. An ensemble of epoch-wise empirical bayes for few-shot learning. In *European Conference on Computer Vision (ECCV)*, 2020. 7

[21] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 2, 6

[22] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2218–2227, 2020. 6, 7, 16

[23] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 2

[24] K. B. Petersen and M. S. Pedersen. The matrix cookbook. Version 20081110. 4

[25] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. 7, 16

[26] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 49–58, 2016. 16

[27] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. 2, 7, 12

[28] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 2

[29] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 4135–4144, 2020. 1, 3, 4, 5, 6, 7, 12, 14

[30] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4077–4087, 2017. 1, 2, 4, 5, 6, 7, 14, 16

[31] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 824–832, 2015. 2

[32] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1199–1208, 2018. 1, 2, 3, 6

[33] Luming Tang, Davis Wertheimer, and Bharath Hariharan. Revisiting pose-normalization for fine-grained few-shot recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14340–14349, 2020. 6, 7, 16

[34] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 5, 7, 12

[35] Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 2252–2262, 2017. 16

[36] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020. 7

[37] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3630–3638, 2016. 2, 6, 7

[38] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 6

[39] Chaoyang Wang, Hamed Kiani Galoogahi, Chen-Hsuan Lin, and Simon Lucey. Deep-lk for efficient adaptive object tracking. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 627–634, 2018. 2

[40] Yan Wang, Wei-Lun Chao, Kilian Q. Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019. 2, 5, 6, 7

[41] Davis Wertheimer and Bharath Hariharan. Few-shot learning with localization in realistic settings. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6558–6567, 2019. 2, 6, 13

[42] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8805–8814, 2020. 2, 3, 4, 5, 6, 7, 8, 12, 14, 16

[43] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 12200–12210, 2020. 2, 4, 5, 6, 7, 12, 13, 15

[44] Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. Laplacian regularized few-shot learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11660–11670. PMLR, 2020. 7, 16