Few-Shot Learning for Dermatological Disease Diagnosis

Viraj Prabhu[1*] VIRAJP@GATECH.EDU
Anitha Kannan[3] ANITHA@CURAI.COM
Murali Ravuri[3] MURAI@CURAI.COM
Manish Chablani[3] MANISH@CURAI.COM
David Sontag[2†] DSONTAG@MIT.EDU
Xavier Amatriain[3] XAVIER@CURAI.COM

[1] *Georgia Tech* [2] *MIT* [3] *Curai*

Abstract

We consider the problem of clinical image classification for the purpose of aiding doctors in dermatological disease diagnosis. Diagnosis of dermatological conditions from images poses two major challenges for standard off-the-shelf techniques: First, the distribution of real-world dermatological datasets is typically long-tailed. Second, intra-class variability is large. To address the first issue, we formulate the problem as low-shot learning, where once deployed, a base classifier must rapidly generalize to diagnose novel conditions given very few labeled examples. To model intra-class variability effectively, we propose Prototypical Clustering Networks (PCN), an extension to Prototypical Networks (Snell et al., 2017) that learns a mixture of "prototypes" for each class. Prototypes are initialized for each class via clustering and refined via an online update scheme. Classification is performed by measuring similarity to a weighted combination of prototypes within a class, where the weights are the inferred cluster responsibilities. We demonstrate the strengths of our approach in effective diagnosis on a realistic dataset of dermatological conditions.

1. Introduction

Globally, skin disease is one of the most common human illnesses that affects 30% to 70% of individuals, with even higher rates in at-risk subpopulations where access to care is scarce (NHANES, 1978; Bickers et al., 2006; J.K.Scholfield et al., 2009; Hay and Fuller, 2012; Basra and Shahrukh, 2009). Untreated or mistreated skin conditions often lead to detrimental effects including physical disability and death (Basra and Shahrukh, 2009). A large fraction of skin conditions are diagnosed and treated at the first point of contact, *i.e.* by primary care practitioners (PCPs), either in a clinical setting or in a telemedicine scenario. While this makes access to care faster, recent studies indicate that general physicians, especially those with limited experience, may not be well-trained for diagnosing many skin conditions (Federman et al., 1999; Goldman, 2007). Effective solutions to *aid* scale doctors in accurate diagnosis motivates this work.

What makes diagnosis of skin conditions challenging? One important factor is the sheer number of dermatological conditions. The International Classification for human Diseases (ICD) enumerates more than 1000 skin or skin-related illnesses. However, most PCPs are trained on a few tens of

[*] Work done as research intern at Curai.
[†] Work done as advisor to Curai.

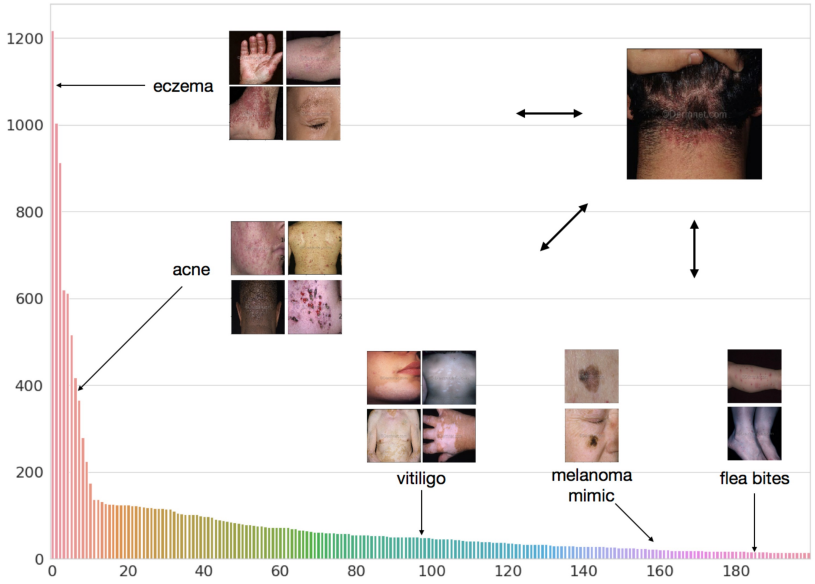

Figure 1: Long-tailed class distribution of Dermnet (shown here for the top-200 classes). Also shown are nearest neighbors to four of the many prototypes learned for select classes using the proposed Prototypical Clustering Network approach. This is illustrative of the huge intra-class variability in the data. For a novel test image, shown at the upper right corner, the model predicts the correct class by measuring weighted similarity to per-class clusters in the embedding space learned through a deep convolutional neural network.

common skin ailments (Wilmer et al.). To make an accurate diagnosis, the knowledge of all possible diseases is important, especially to workup and eliminate potentially life-threatening conditions. The difficulty of diagnosis is further compounded by large intra-class variability (acne may occur on the face, hand, scalp, etc.). To motivate the scale of this problem, see Figure 1, where we show the class distribution of the Dermnet Skin Disease Atlas [1], a publicly available large-scale dataset of dermatological conditions.

These issues create an opportunity for incorporating machine learning systems into the doctor's workflow, aiding them in sieving through possible skin conditions. AI systems have shown promising results in the healthcare domain, with early applications on automated detection of skin lesions from images (*c.f.* Esteva et al. (2017)), and diagnosis based on radiology data (*c.f.* Li et al. (2018)). **Clinical Relevance:** Inspired by these successes, this paper tackles the problem of fine-grained skin disease classification. We conjecture that a high-fidelity AI system can serve as a diagnostic decision support system to general physicians. By suggesting candidate diagnoses, it can greatly reduce effort and compensate for the possible lack of experience or time at the point of care. Further, such a system can aid in triaging the right doctor resource in a timely manner, especially when acute conditions need immediate care (Goldman, 2007). Our primary goal is to design a dermatological diagnosis model amenable to deployment in a primary care setting, that can aid doctors by providing a list of candidate diagnoses. Additionally, we seek to expand the scope of diagnosis to data-poor

1. http://www.dermnet.com/

diseases, as well as diseases exhibiting significant variability in manifestation, in an attempt to improve the coverage, applicability, and cost-effectiveness of such services.

Technical Signficance: Motivated by the long-tailed and diverse characteristics of dermatological data, we formulate dermatological diagnosis as a *few-shot learning* problem and propose models that can both effectively learn from such data and easily be extended to diagnose new conditions given very few labeled examples. Specifically, our task is to learn a base diagnosis model that, once deployed, can be easily extended to new classes from a few labeled examples (potentially labeled by a physician). To our knowledge, ours is the first work to study the applicability of few-shot learning in assistive diagnosis. Our approach pursues the following objectives:

- **Modeling intra-class variability:** Several conditions contain significant intra-class variability, *e.g.* a condition like acne may occur on the face, back, scalp, etc.
- **Modeling the long-tail.** The data distribution is invariably long tailed. Some conditions are rare and may not have many recorded examples (Fig. 1, "melanoma mimic"), while others may be common but easy to diagnose and so not frequently recorded (Fig. 1, "flea bites").
- **Learning without forgetting:** The ability to diagnose novel conditions must not compromise the performance on base classes.
- **Privacy preservation:** As access to most dermatological data (usually part of Electronic Health Records) is strictly controlled, the model should not require access to the original (potentially, proprietary) training data when being extended.

Our proposed model, that we call Prototypical Clustering Networks (PCN), extends prior few-shot learning work on Prototypical Networks (Snell et al., 2017), which have been shown to be a simple and highly effective metric learning formulation to learn from limited data. Specifically, to model the diversity of dermatological data, we represent each class as a mixture of learned representative "prototypes", instead of a single prototype. Training this classifier involves learning an embedding space while simultaneously learning to represent each class by a set of prototypes. Prototypes are initialized for each class via clustering and refined via an online update scheme. Classification is performed by measuring similarity to a weighted combination of prototypes within a class, where the weights are the inferred cluster responsibilities. The examples shown in Figure 1 are, in fact, nearest neighbors to prototypes of the classes learned using the proposed approach. Quantitative results demonstrate the strengths of our approach for dermatological disease diagnosis against strong baselines, and we further provide detailed analyses of the learned representations.

2. Related Work

Dermatological Classification. A few prior works address the problem of dermatological classification. In Esteva et al. (2017), authors focus specifically on diagnosing skin cancer, and establish a benchmark on a large closed-source dataset of skin lesions by finetuning a pretrained deep convolutional neural network (CNN). In Liao (2016), authors study the problem of skin disease diagnosis on the Dermnet dataset but focus on coarse 23-way classification using its top-level hierarchy. In Sun et al. (2016), the authors propose a benchmark dataset for skin disease diagnosis containing 6584 clinical images. In follow up work, Yang et al. (2018) propose an approach to learn representations inspired by diagnostic criteria employed by dermatologists on this dataset. In this work, we study fine-grained recognition of skin conditions on the Dermnet dataset which is a signficantly larger dermatological resource containing over 23000 images. Further, we formulate this as a few-

shot learning setup, and propose a method to model diverse classes and generalize effectively to previously unseen novel classes with very little labeled data.

Class-imbalanced datasets. Real-world visual datasets frequently possess long tails (Van Horn and Perona, 2017; Wang et al., 2017; Zhu et al., 2014), and learning robust representations from such data is a topic of active research. Conventional training methods typically lead to poor generalization on tail classes as class-prior statistics are skewed towards the head of the distribution. Simple techniques such as random oversampling (or undersampling) by repeating (or removing) tail instances are found to help mitigate this issue to a degree (Buda et al., 2018). Alternative approaches perform meta-learning to transfer knowledge from data-rich head classes to the tail (Wang et al., 2017). In this work, we propose a few-shot learning approach on a real-world imbalanced dataset of dermatological conditions, and demonstrate strong few-shot generalization capabilities.

Few-shot learning. Few-shot learning aims to learn good class representations given very few training examples (Koch et al., 2015; Santoro et al., 2016; Vinyals et al., 2016; Snell et al., 2017). Main paradigms of approaches include simulating data starved environments at training time, and including non-parametric structures in the model as regularizers. Matching networks (Vinyals et al., 2016) learn an attention mechanism over support set labels to predict query set labels for novel classes. Prototypical networks (Snell et al., 2017) jointly learn an embedding and centroid representations (as class *prototypes*), that are used to classify novel examples based on Euclidean distance. In both Vinyals et al. (2016) and Snell et al. (2017), embeddings are learned end-to-end and training employs episodic sampling. Some recent approaches learn to directly predict weights for new layers from embedding layer activations of support examples (Qi et al., 2018; Qiao et al., 2017). In Gidaris and Komodakis (2018), the motivation is to perform few shot learning 'without forgetting', i.e. extending to novel classes without catastrophic forgetting (also studied as generalized few-shot learning). In Hariharan and Girshick (2017), authors study few-shot learning by creating an imbalanced few-shot benchmark from ImageNet (Russakovsky et al., 2015), and propose a method to "hallucinate" additional samples for such data-starved classes. In this work, we focus on a similar setup on the real-world long-tailed Dermnet dataset.

Prototypical Networks. Prior extensions to Protoypical Networks exist in the literature, and here we distinguish our contributions (Triantafillou et al., 2018; Fort, 2017). In Triantafillou et al. (2018), authors propose extending Prototypical Networks to a semi-supervised setting by using unlabeled examples while producing prototypes. In Fort (2017), authors propose additionally predicting a covariance estimate for each embedding and using a direction and class dependent distance metric instead of euclidean distance. In this work, we extend prototypical networks to model multimodal classes in an automated diagnostic setting by learning multiple prototypes per class, that are initialized via clustering and refined via an online update scheme.

3. Approach

We formulate dermatological image classification as a low-shot learning problem. During training time, we have access to a labeled dataset of images $S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ where each x_i is an observation and $y_i \in \{1, ..., K_{base}\}$ is the label mapping to one of the *base* classes known at training time. At test time, we are also provided with a small labeled dataset corresponding to K_{novel} novel classes, and must learn to perform $K_{base+novel}$ way classification.

Algorithm 1: Training episode loss computation for Prototypical Clustering Networks. N is the number of examples in the training set, K_{base} is the number of base classes for training, M_k is the number of clusters for class k, $N_C \leq K_{base}$ is the number of classes per episode, N_S is the number of support examples per class, N_Q is the number of query examples per class. RANDOMSAMPLE(S, N) denotes a set of N elements chosen uniformly at random from set S, without replacement. *Differences from Algorithm 1 in Snell et al. (2017) in blue.*

Input:Training set $\mathcal{D} = \{(x_1, y_1), \cdots, (x_N, y_N)\}$, where each $y_i \in \{1, \cdots, K\}$. $\mathcal{D}_k$ denotes the subset of $\mathcal{D}$ containing all class prototypes, i.e. elements $(x_i, y_i) = \{\mu_{z,k}\}_{z=1}^{M_k} \forall k \in \{1, \cdots, K\}$

Output: The loss L_ϕ for a randomly generated training episode

V $\leftarrow$ RANDOMSAMPLE($\{1, \cdots, K\}$, N_C) // Select class indices for episode

for $k \in \{1, \cdots, N_C\}$ **do**

  $S_k \leftarrow$ RANDOMSAMPLE($\mathcal{D}_{v_k}$, N_S) // Select support examples

  $Q_k \leftarrow$ RANDOMSAMPLE($\mathcal{D}_{v_k \setminus S_k}$, N_Q) // Select query examples

  // Compute probabilistic assignment of x to y's clusters

  for $z \in \{1, \cdots, M_k\}$ **do**

    for $(x, y) \in S_k$ **do**

      $q(z|k, x) = \frac{\exp(-d(f_\phi(x), \mu_{z,k})/\tau)}{\sum_{z'} \exp(-d(f_\phi(x), \mu_{z',k})/\tau)}$

    end for

    $\mu_{z,k}^{new} \leftarrow \alpha \mu_{z,k}^{old} + (1 - \alpha) \frac{\sum_{(x,y) \in S_k} q(z|k,x) f_\phi(x)}{\sum_{(x,y) \in S_k} q(z|k,x)}$

  end for

end for

$L_\phi \leftarrow 0$

for $k \in \{1, \cdots, N_C\}$ **do**

  for $(x, y) \in Q_k$ **do**

    $r_{x,y} \leftarrow \sum_z q(z|k, x) d(f_\phi(x), \mu_{z,k}) + \log \sum_{k'} \exp(-\sum_{z'} q(z'|k', x) d(f_\phi(x), \mu_{z',k'}))$

    $L_\phi \leftarrow L_\phi + \frac{r_{x,y}}{N_C N_Q}$

  end for

end for

3.1. Model

Prototypical Clustering Networks (PCN) build upon recent work in Prototypical Networks (Snell et al., 2017). PCN represents each class using a set of prototypical representations learned from the data. Let $\{\mu_{z,k}\}_{z=1}^{M_k}$ be the collection of M_k prototypes for class k. Then, at test time, we measure similarity to these representations to derive its corresponding class label. In particular,

$$p(y = k|\mathbf{x}) = \frac{\exp(-\sum_z q(z|k,x) d(f_\phi(x), \mu_{z,k}))}{\sum_{k'} \exp(-\sum_z q(z|k,x) d(f_\phi(x), \mu_{z,k'}))} \quad (1)$$

where $f_\phi(x)$ is the embedding function with learnable parameters ϕ that maps input x to a learned representation space, d is a distance function and $q(z|k,x)$ (eq. 2) is a soft assignment of examples to clusters from the class. When $M_k = 1$ for all classes, we revert to prototypical networks.

3.1.1. MODEL TRAINING

Our goal is to learn a model with parameters ϕ so as to maximize the probability of the correct class such that $\phi^* = \arg\max_\phi \sum_{(\mathbf{x},y)} \log p(y|\mathbf{x};\phi) = \arg\min_\phi L_\phi$, where L_ϕ is the corresponding loss function. We use episodic training (Snell et al., 2017; Vinyals et al., 2016; Ravi and Larochelle, 2016) to learn the embedding function by optimizing the loss and updating the cluster prototypes for each class. In particular, a training epoch consists of E episodes. Algo. 1 provides the details of computing the loss for one episode that is used in learning the function.

Class-specific cluster responsibilities: The assignment of an example within a class is given by:

$$q(z|k,x) = \frac{\exp(-d(f_\phi(x), \mu_{z,k})/\tau)}{\sum_{z'} \exp(-d(f_\phi(x), \mu_{z',k})/\tau)}, \tag{2}$$

where τ is temperature parameter that controls the variance of the distribution. As we decrease the temperature, the distribution becomes more peaky, and becomes flatter as we increase it. The importance of τ can be understood by studying the loss function L_ϕ in line 15 of Algo. 1. During training, if clusters are well-separated, $q(z|k,x)$ will be peaky so that each example effectively contributes to the update of a single cluster in a class, whereas if clusters overlap, $q(z|k,x)$ will be diffuse and the corresponding example will contribute to multiple prototypes.

Class-specific cluster prototypes: In episodic training, an epoch corresponds to a fixed number of episodes and within each episode, classes are sampled uniformly. In our setting with huge class imbalance, this translates to oversampling examples from the tail and undersampling examples from the head, which can adversely affect model training. To mitigate this, at the start of an epoch, we initialize cluster prototypes for each class using k-means [2] on the learned embedding representation of examples from the entire training set of that class. We rerun this clustering step at the start of each epoch to prevent collapse to using only a single cluster per class.

Subsequently, in each episode, we use an *online* update scheme that balances between the local estimate of the prototype computed from embeddings of the current support set (to account for the evolving embedding space), and the prototypes learned so far:

$$\mu_{z,k}^{new} \leftarrow \alpha \mu_{z,k}^{old} + (1-\alpha) \frac{\sum_{(x,y)\in S_k} q(z|k,x) f_\phi(x)}{\sum_{(x,y)\in S_k} q(z|k,x)}, \tag{3}$$

where α trades off memory from previous episodes and its current estimate.

3.2. Understanding the role of multiple clusters

We can derive insights about the role of multiple clusters by interpreting PCN as a non-linear generalization of PN (Snell et al., 2017). Using squared Euclidean distance in Eqn. 1, we expand the term in the exponent so that:

2. We empirically found using a fixed number of clusters per class to work best. Choosing an optimal number of clusters per class is challenging, largely due to differences in the amount of intra-class variability across classes. In an attempt to bypass this challenge, we experimented with using affinity propagation (Frey and Dueck, 2007), but found it to be highly unstable as a) the similarity function (needed for clustering) is based on the embedding representation that is optimized concurrently, and b) there is no straightforward online update rule (equivalent to eq. 3) to subsequently update within an episode.

$$-\sum_z q(z|k,x)||f_\phi(x)-\mu_{z,k}||^2 = \text{const. for k} + 2\sum_z q(z|k,x) f_\phi(x)^T \mu_{z,k} \quad (4)$$

$$-\sum_z q(z|k,x)\mu_{z,k}^T \mu_{z,k}$$

$$= \text{const. for k} + 2 w_{k,x}^T f_\phi(x) - b_{k,x} \quad (5)$$

where

$$w_{k,x} = \sum_z q(z|k,x)\mu_{z,k} \quad (6)$$

$$b_{k,x} = \sum_z q(z|k,x)\mu_{z,k}^T \mu_{z,k} \quad (7)$$

The last two terms in Eqn. 5 are non-linear functions of the data, where the non-linearity is captured through both the embedding and the mixing variables. The functional forms of the factors, namely $w_{k,x}$ and $b_{k,x}$, also sheds light on the advantage of using multiple clusters per class. In particular, unlike in prototypical networks, $w_{k,x}$ is an *example-specific* "prototypical" representation for class k, obtained by using a convex combination of prototypes for the class, weighted by posterior probability over within-class cluster assignments. When $q(z|k,x)$ is confident with a peaky posterior, the model behaves like a regular prototypical network. In contrast, when the posterior has uncertainty, PCN interpolates between the prototypes by modulating $q(z|k,x)$.

4. Results

4.1. Experimental setup

Dataset: We construct our dataset from the Dermnet Skin Disease Atlas, one of the largest public photo dermatology sources containing over 23,000 images of dermatological conditions. These images are clinical images collected through various sources, including mobile phones, digital cameras, etc. and so vary in pose, lighting, and resolution. Images are annotated at a two level hierarchy – a coarse top-level containing parent 23 categories, and a fine-grained bottom-level containing more than 600 skin conditions. We focus on the more challenging bottom-level hierarchy for our experiments. First, we remove duplicates from the dataset based on name, and also based on collisions found using perceptual image hashing (Zauner, 2010).

Figure 1 presents a histogram of the resulting class distribution, filtered to the top-200 classes. We can see that the dataset has a long tail with only the 100 largest classes having more than 50 images; beyond 200 classes, the number of images per class reduces to double digits, and with 300 classes to single digits. Unless otherwise stated, for experimental comparisons, we focus on the top-200 classes so that $K_{base+novel} = 200$, which contains 15507 images. Similar to Hariharan and Girshick (2017), we treat the largest 150 classes as base classes ($K_{base} = 150$) and the remaining 50 classes as novel ($K_{novel} = 50$). This helps in ensuring reasonably sized splits for training, validation, and testing. In particular, we sample $max(5, 20\%)$ without replacement for each base class to get validation and test splits (3163 images each). The remaining is used for training (9181 images). For the low-shot learning phase, following the procedure used in Hariharan and Girshick (2017), we sample 5 examples each for training and testing, respectively. We report mean and standard deviation of metrics over 10 cross validation runs.

Metrics: As our dataset is imbalanced, we report mean of per-class accuracy (mca) as our metric, treating each class as equally important. For a dataset consisting of C classes, with T_c examples in each class, mean accuracy is the average of per-class accuracies: mca $= \frac{1}{C}\sum_c \frac{\sum_{t=1}^{T_c} I[\hat{y}^{(t)}[0]=y^{(t)}]}{T_k}$, where, for t^{th} example, $\hat{y}^{(t)}[j]$ is the j^{th} top class predicted from a model and $y^{(t)}$ is its corresponding ground truth label, where I denotes the indicator function.

We use $\text{mca}_{\text{base+novel}}$ to report combined mca performance of examples from all classes. mca_{base} corresponds to similar $K_{base+novel}$-way evaluation but restricted to test examples from base classes, and $\text{mca}_{\text{novel}}$ corresponds to evaluation on test examples belonging to novel classes alone.

We also report recall@k (k $\in \{5, 10\}$). This metric (also called *sensitivity*) is valuable in deployment contexts that involve aiding doctors in diagnosis, as it ensures that the relevant condition is considered within a small range of false positives. However, since our test set is imbalanced, recall@k metrics unfairly reward strong performance on the head classes, and so to provide a fairer comparison, we report *balanced* (or macro) recall@k metrics, wherein we compute recall@k for each class and average, treating each as equally important [3].

Model: We initialize a 50-layer ResNet-v2 (He et al., 2016), a state-of-the-art convolutional neural network architecture for image classification, with ImageNet pretraining [4], and train a Prototypical Clustering Network as described in Sec. 3 on K_{base} classes. We use 10 and 4 clusters per class for base and novel classes respectively, and a temperature of 1.0 (all picked via grid search).

Baselines

- Prototypical Network (PN): We train an ImageNet-pretrained ResNet-V2 CNN as a Prototypical Network (Snell et al., 2017) on K_{base} classes.
- Finetuned Resnet with nearest neighbor (FT_K-*NN): Here, we finetune an ImageNet-pretrained ResNet-v2 convolutional neural network with 50 layers (He et al., 2016) on training data from K classes. We report numbers for K $\in \{K_{base}, K_{base+novel}\}$. The model is trained as a softmax classifier with a standard cross entropy objective. Then, we obtain embeddings for the entire training set consisting of $K_{base+novel}$ classes, followed by $*$-nearest neighbor classification on the test set belonging to $K_{base+novel}$ classes.
- Finetuned ResNet (FT_K-CE): We use the same ResNet model as above, with $K = K_{base+novel}$, i.e. trained for $K_{base+novel}$ way classification using training data from both base and novel classes, and validated using the corresponding validation set on the base classes alone (due to lack of data in novel classes). We train the model with class balancing. This is a strong baseline as we use all $K_{base+novel}$ during training, and also due to class balancing, which has been shown to be an effective strategy for training CNN models in the presence of class imbalance (Buda et al., 2018).

Hyperparameters: For PN and PCN, we use episodic batching with 10-way 10-shot classification (at train), and 200 episodes per epoch. At test, we compute per-class prototypes using the training set for all $K_{base+novel}$ classes, and perform $K_{base+novel}$ way classification. The embedding function for PCN and PN produces 256-dimensional embeddings, and uses the same architecture as in FT_K-CE (with one less fully connected layer). Models are trained with early stopping using Adam (Kingma and Ba, 2014), a learning rate of 10^{-4}, and L2 weight decay of 10^{-5}. For a detailed analysis of impact of episodic memory and softmax temperature, see supplementary.

3. For completeness, we also report micro-averaged recall@k in supplementary
4. We also experimented with training from scratch, and found it to perform significantly worse

Approach	$n = 5$			$n = 10$		
	$\text{mca}_{\text{base+novel}}$	mca_{base}	$\text{mca}_{\text{novel}}$	$\text{mca}_{\text{base+novel}}$	mca_{base}	$\text{mca}_{\text{novel}}$
FT_{150}-1NN	46.2 ± 0.8	55.3 ± 0.3	18.8 ± 3.3	49.5 ± 0.3	54.9 ± 0.5	33.4 ± 1.4
FT_{150}-3NN	44.3 ± 0.3	54.8 ± 0.5	12.8 ± 1.5	47.0 ± 0.6	54.1 ± 0.4	25.6 ± 1.5
FT_{200}-1NN	46.5 ± 0.4	54.2 ± 0.3	22.5 ± 0.8	49.9 ± 0.5	53.8 ± 0.4	38.3 ± 1.3
FT_{200}-3NN	44.7 ± 0.4	52.6 ± 0.2	20.9 ± 2.0	48.0 ± 0.1	52.5 ± 0.1	34.3 ± 0.2
FT_{200}-CE	$\mathbf{47.8 \pm 0.5}$	$\mathbf{55.8 \pm 0.7}$	24.0 ± 3.2	$\mathbf{51.5 \pm 0.4}$	$\mathbf{55.2 \pm 0.3}$	40.4 ± 2.4
PN	43.9 ± 0.4	48.7 ± 0.4	29.6 ± 2.4	44.9 ± 0.8	47.6 ± 0.4	37.1 ± 3.4
PCN (ours)	$\mathbf{47.8 \pm 0.7}$	53.7 ± 0.2	$\mathbf{30.0 \pm 2.8}$	$\mathbf{50.9 \pm 0.6}$	51.4 ± 0.3	$\mathbf{49.6 \pm 2.8}$

Table 1: Mean per-class accuracy (MCA) on top 200 classes. We focus on the low-shot setting, using all training data for the base classes (the largest 150) and $n = 5$ or 10 examples for the remaining 50 classes (denoted as "novel"). Note that FT_{200}-CE and FT_{200}-*NN use training data for all 200 classes, whereas others use only the base classes for representation learning, using support sets from novel classes after training to directly derive prototypes.

4.2. Main results

Table 1 highlights our main MCA results. The table shows test set MCA over the 200 classes available during test time for two different low shot settings: train shots of 5 and 10 with test shot of 5. In both low shot settings, we observe the following trends:

- FT_K-CE and PCN share similar performance on combined MCA. However, their performance on base and novel classes is quite distinct. Much of the performance gains for FT_K-CE come from the base classes that have a lot more training examples than novel classes. In contrast, PCN, through episodic training aims at learning discriminative feature representations that are generalizable to novel classes with highly constrained numbers of examples; this is evident by its significantly better performance (9% absolute gains) in generalizing to novel classes. At the same time, PCN ensures that performance on novel classes does not come at the cost of lower accuracy on base classes. Also note that the FT_K-CE model requires re-training for adding novel classes while PCN only requires a single forward pass to learn prototypes for novel classes.

- FT_K-*NN models learn robust representations for base classes, but are unable to generalize to novel classes, outperforming a regular PN model on top-200 MCA but underperforming against PCN. Interestingly, we find that increasing the number of nearest neighbors leads to poor performance, especially on novel classes. This could be due to sparsity of training data.

- PCN outperforms PN on combined base and novel classes by a large margin. This demonstrates that representing classes with multiple prototypes leads to better generalization on both base and novel classes. In Figure 2, we show the nearest neighbor to class prototype for PN and to four of the PCN prototypes, for select classes. We can see that PCN has learned to model intra-class variability much more effectively. As an example, for eczema and acne

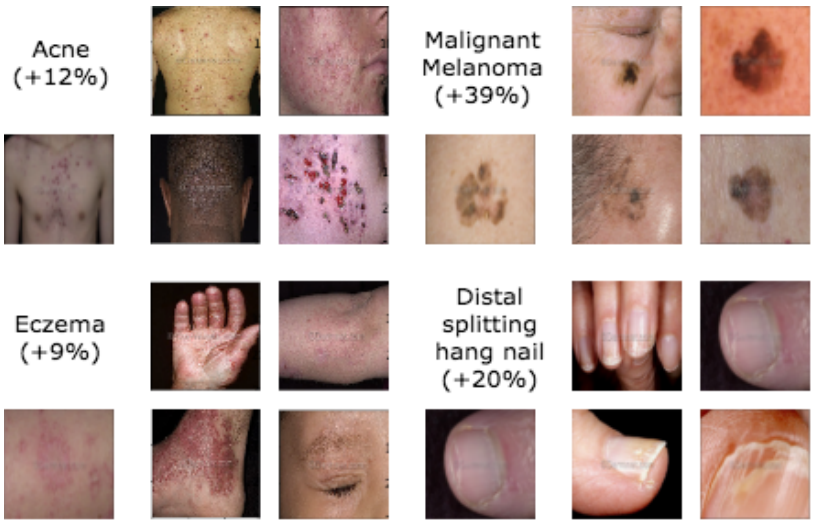

Figure 2: Learned prototypes shown using their nearest neighbors in the training set. Each condition is displayed in a 2×3 grid; The image below the name of the condition corresponds to PN, while the 2×2 grid to its right corresponds to nearest neighbors of four (randomly selected) cluster prototypes. +X% below the name denotes $\text{mca}_{base+novel}$ improvement of PCN over PN for that class. Note that novel classes such as 'Distal splitting hang nail' can also be diverse, as shown by clusters identified with PCN.

classes we can see that PCN learns clusters corresponding to these skin conditions in different anatomical regions. We provide a more in-depth comparison in the next section.

In Table 2 we report balanced recall@k (br@k) performance. Here we clearly find PCN to outperform all baselines owing to strong performance across the board on base and novel classes. Further, in supplementary we provide a per-class performance comparison of our approach against the FT_{200}-CE baseline that highlights its strengths in diagnosis of data-poor novel conditions.

Finally, as a sanity check of the general applicability of our approach, as well as to compare to the state of the art, we conduct experiments on the *mini*ImageNet dataset (Vinyals et al., 2016), an established benchmark in the few shot learning literature. We find that our approach performs competitively with the state of the art. For details, please look at supplementary material.

4.3. Comparison between PCN and PN

PCN or PN with post-hoc clustering? To understand the effectiveness of PCN, we compare it to a PN model with which we perform "post-hoc" clustering: (a) cluster novel class representations using the PN model's learned embeddings (with cluster size of 4) (b) cluster both base and novel class representations (with cluster size of 10 and 4, as in PCN). Rows 1-3 in Table 3 compare the performance between different post-hoc clustering variants of PN against PCN. We see that PCN leads in all metrics across the board; thus such post-hoc clustering does not lead to improved performance. A reason for this is that the PN model is optimized to learn representations assuming a projection to a single cluster for each class, and hence clustering on such learned representation does not improve performance. This further validates the importance of training with multiple clusters.

	Approach	br@$5_{\text{base+novel}}$	br@5_{base}	br@5_{novel}	br@$10_{\text{base+novel}}$	br@10_{base}	br@10_{novel}
n=5	FT_{200}-CE	65.4 ± 0.7	74.6 ± 0.2	38.0 ± 2.1	73.1 ± 0.5	82.3 ± 0.2	45.3 ± 1.5
	PN	66.5 ± 0.6	67.6 ± 0.2	$\mathbf{63.2 \pm 2.3}$	75.3 ± 0.5	74.9 ± 0.1	$\mathbf{76.3 \pm 2.1}$
	PCN (ours)	$\mathbf{70.7 \pm 0.6}$	$\mathbf{74.5 \pm 0.2}$	$\mathbf{59.2 \pm 2.6}$	$\mathbf{79.1 \pm 1.0}$	$\mathbf{81.4 \pm 0.3}$	$\mathbf{72.2 \pm 3.6}$
n=10	FT_{200}-CE	69.9 ± 0.5	$\mathbf{73.8 \pm 0.6}$	58.0 ± 3.2	77.9 ± 0.6	81.9 ± 0.2	65.9 ± 2.6
	PN	67.5 ± 0.4	66.0 ± 0.3	$\mathbf{72.2 \pm 1.7}$	75.9 ± 0.6	72.9 ± 0.3	$\mathbf{84.7 \pm 2.2}$
	PCN (ours)	$\mathbf{71.4 \pm 0.7}$	70.4 ± 0.2	$\mathbf{74.5 \pm 2.6}$	$\mathbf{79.9 \pm 0.5}$	$\mathbf{78.2 \pm 0.2}$	$\mathbf{85.0 \pm 2.0}$

Table 2: Balanced Recall@k on top 200 classes.

Model	Eval CPC (base / novel)	$\text{mca}_{\text{base+novel}}$	mca_{base}	$\text{mca}_{\text{novel}}$	recall@5	recall@10
PN	1 / 1	43.9 ± 0.4	48.7 ± 0.4	29.6 ± 2.4	70.9 ± 0.4	80.2 ± 0.3
PN	1 / 4	44.4 ± 0.5	50.4 ± 0.4	26.4 ± 2.3	74.2 ± 0.2	83.5 ± 0.3
PN	10 / 4	43.8 ± 0.8	50.3 ± 0.2	24.2 ± 3.0	75.6 ± 0.2	84.0 ± 0.2
PCN (ours)	10 / 4	$\mathbf{47.8 \pm 0.7}$	$\mathbf{53.7 \pm 0.2}$	$\mathbf{30.0 \pm 2.8}$	$\mathbf{77.8 \pm 0.2}$	$\mathbf{86.0 \pm 0.4}$

Table 3: Does post-hoc clustering on PN help?

Varying test shot: Table 4 highlights the effect of number of support examples (shot). As we increase the shot, the performance improves on both methods, but that improvement is larger for PCN than for PN. Because of this, the performance gap between the two methods drastically increases. PCN is better at utilizing the availability of more data by partitioning the space with clusters.

Extending the tail: We now vary the number of novel classes at test time from 50 to 150, bringing the total number of classes up from 200 to 300. Table 4 reports results. We use a train and test shot of 2 and 5 respectively, as tail classes are extremely sparse. Results are reported with 10-fold cross validation. While there is a drop in performance for both models due to small shot sizes, we can see that the performance gap between PCN and PN continues to hold.

		PN	PCN
Shot	2	42.26	45.36
	5	44.35	47.79
	10	44.93	50.92
K	200	42.26	45.58
	250	37.08	40.37
	300	34.70	37.67

Table 4: Extending Shot, Tail.

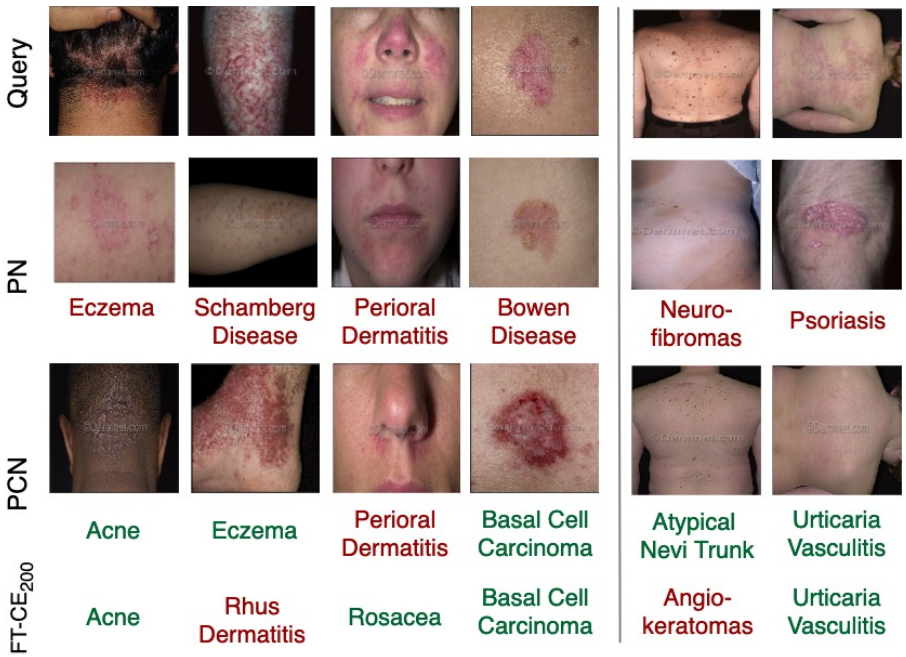

Figure 3: We compare PCN with PN and FT_{200}-CE on query images from the test set. For each image, we color code correct predictions in green and incorrect predictions in red. For PN, we show the nearest neighbor to the prototype of the predicted class. Similarly, for PCN, we show the nearest neighbor of the *top cluster* according to $q(z|c, x)$ of the predicted class. The last two columns correspond to examples from novel classes.

4.4. Qualitative Results

Figure 3 provides qualitative examples comparing the three methods. Consider column 1. Acne is one of the largest classes in the base classes with large intra-class variability. Both FT_{200}-CE and PCN can diagnose this example correctly. However, PN due to its limited capacity to represent the huge variability in the class is confused with another large class, eczema. PCN, due to having access to multiple clusters can learn a better representation and correctly diagnose acne. The last two columns correspond to novel classes.

Effectiveness of multiple clusters. In Sec 3, we show how PCN can interpolate between the learned prototypes by modulating $q(z|c, x)$. In Figure 4 we show some qualitative examples to illustrate this. We show query images from the test set for various classes, with a mix of correct and incorrectly classified examples. Below the class label, we show the nearest neighbor image from the training set to each of the learned prototypes for the class *predicted* by PCN, and below each query image, cluster responsibilities placed by the model on each of these prototypes. As an example, consider Figure 4(a). This example (acne) is correctly classified by PCN. Interestingly, 4(a), the model appears to interpolate between two prototypes that are similar to the query image in pose and skin texture respectively to make its prediction. We can see that the $q(z|c, x)$ distribution varies quite a bit across examples, being a lot more diffuse in some cases than others. It can also be seen that the model learns to accurate place probability mass on similar prototypes. Figure 4(c), (d)(bottom) shows incorrectly predicted examples; Even for these examples, the model seems to interpolate,

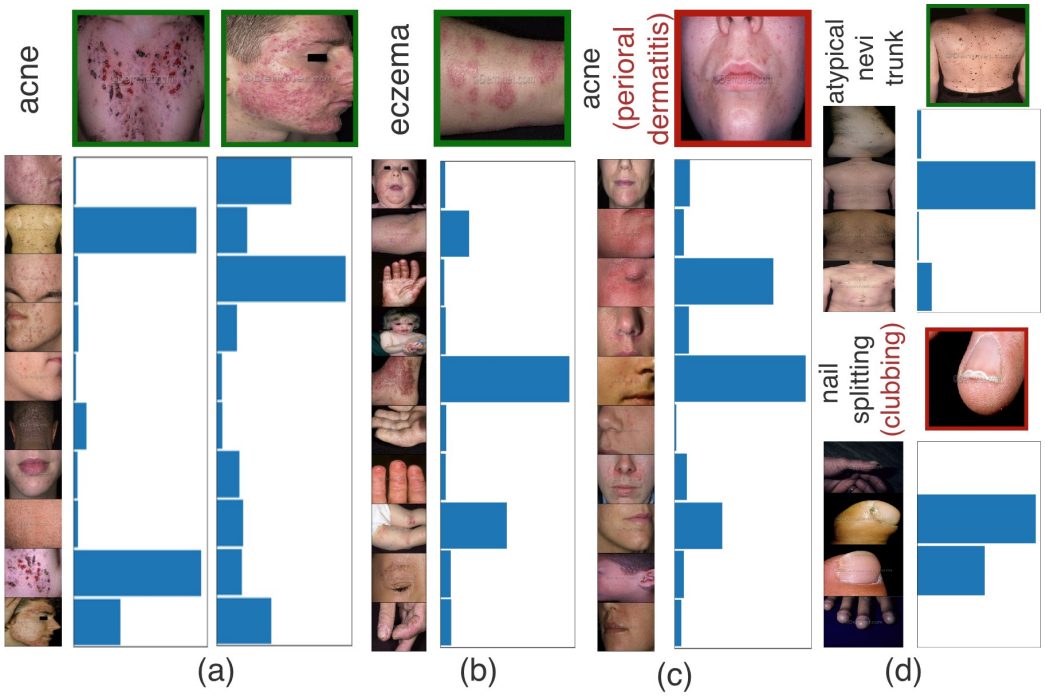

Figure 4: Effectiveness of using multiple clusters. Shown for base and novel classes. For each class, we show the nearest neighbor to the learned prototypes of the class *predicted* by PCN, as well as inferred cluster responsibilities $q(z|c, x)$ conditioned on the prediction. (a)-(b): Correctly classified test examples. (c): "Acne" test example misclassified as "perioral dermatitis" (d): Test examples corresponding to novel classes.

albeit incorrectly, to make predictions. Note that Figure 4(d) corresponds to examples from novel classes. Such visualizations lend a degree of interpretability to the model's decision process.

5. Conclusion

We propose Prototypical Clustering Networks, a few shot learning approach to dermatological image classification. This method is scalable to novel classes, and can effectively capture intra-class variability. We observe that our approach outperforms strong baselines on this task, especially on the long tail of the data distribution. Such a machine learning system can be a valuable aid to medical providers, and improve access, equity, quality, and cost-effectiveness of healthcare.

There exist a number of future directions worth pursuing. The true utility of our system is in aiding the physician, and this requires studies that include such a deployment. Other extensions include determining the absence of any condition (adding *i.e.* a 'normal' class), and controlling for demographic variables when appropriate. Another interesting direction would be to incorporate additional modalities of data for more robust diagnosis. Further, while our approach learns features end to end, a natural alternative would be to fuse such learned representations with features designed using domain expertise, such as diagnostic markers and patient symptoms that are typically tracked by dermatologists. Incorporating such medical symptoms into the classification task will be another interesting direction of future work.

References

World health organization, international statistical classification of diseases and related health problems: Chapter xii: Diseases of the skin and subcutaneous tissue. URL http://apps.who.int/classifications/icd10/browse/2016/en.

M.K.A. Basra and M. Shahrukh. Burden of skin diseases. *Expert Review Pharmacoeconomics Outcomes Research*, 2009.

D.R. Bickers, H.W. Lim, and D. Margolis. The burden of skin diseases: 2004 a joint project of the american academy of dermatology association and the society for investigative dermatology. *Journal of American Academy Dermatology*, 2006.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639): 115, 2017.

DG. Federman, J.Concato, and RS. Kirsner. Comparison of dermatologic diagnoses by primary care practitioners and dermatologists. a review of the literature. *Archives of Family Medicine*, 8(2), 1999.

Stanislav Fort. Gaussian prototypical networks for few-shot learning on omniglot. *arXiv preprint arXiv:1708.02735*, 2017.

Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315:2007, 2007.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018.

Erik. E. Goldman. Skin diseases get misdiagnosed in primary care. *Family Practice News*, 2007. URL https://www.mdedge.com/sites/default/files/issues/articles/71073_main_1.pdf.

Bharath Hariharan and Ross B Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, pages 3037–3046, 2017.

R.J. Hay and L.C. Fuller. The assessment of dermatological needs in resource-poor regions. *International Journal of Dermatology*, 2012.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

J.K.Scholfield, D. Grindlay, and H.C. Williams. Skin conditions in the uk: A health needs assessment. *University of Nottingham, Centre of Evidence Based Dermatology UK; Nottingham, UK*, 2009.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Zhe Li, Chong Wang, Mei Han, Yuan Xue, Wei Wei, Li-Jia Li, and Fei-Fei Li. Thoracic disease identification and localization with limited supervision. *IEEE Computer Vision and Pattern Recognition*, 2018.

Haofu Liao. A deep learning approach to universal skin disease classification. *University of Rochester Department of Computer Science, CSC*, 2016.

NHANES. Skin conditions and related need for medical care among persons 174 years. *NHANES: United State*, 1978.

Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5822–5830, 2018.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan Yuille. Few-shot image recognition by predicting parameters from activations. *arXiv preprint arXiv:1706.03466*, 2, 2017.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*, 2016.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

Xiaoxiao Sun, Jufeng Yang, Ming Sun, and Kai Wang. A benchmark for automatic visual classification of clinical skin disease images. In *European Conference on Computer Vision*, pages 206–222. Springer, 2016.

Eleni Triantafillou, Hugo Larochelle, Jake Snell, Josh Tenenbaum, Kevin Jordan Swersky, Mengye Ren, Richard Zemel, and Sachin Ravi. Meta-learning for semi-supervised few-shot classification. 2018.

Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pages 7029–7039, 2017.

EN. Wilmer, CJ. Gustafson, CS. Ahn, SA. Davis, SR. Feldman, and WW. Huang. Most common dermatologic conditions encountered by dermatologists and nondermatologists. *Cutis*, 94(6).

Jufeng Yang, Xiaoxiao Sun, Jie Liang, and Paul L Rosin. Clinical skin lesion diagnosis using representations inspired by dermatologist criteria. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1258–1266, 2018.

Christoph Zauner. Implementation and benchmarking of perceptual image hash functions. 2010.

Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. Capturing long-tail distributions of object subcategories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, 2014.

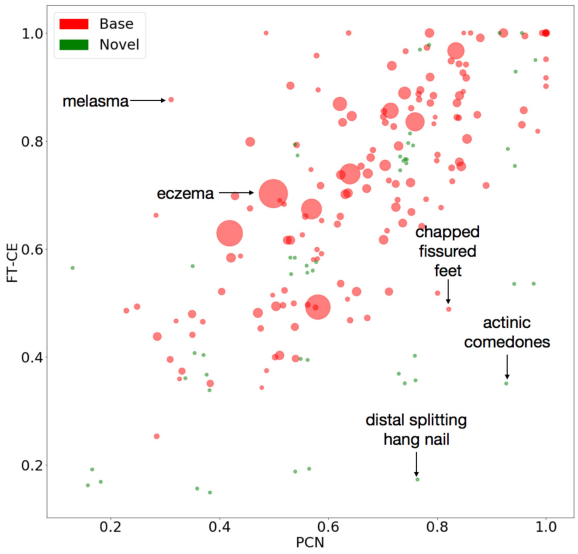

Figure 1: Comparison between FT_{200}-CE and PCN: Per-class accuracy. Each class is denoted by a dot and the area of dot is proportional to the number of training examples for the class.

Supplementary Material: Few-Shot Learning for Dermatological Disease Diagnosis

In this supplementary material we first provide a per-class performance comparison of our proposed approach against the FT_{200}-CE baseline. We then analyze the role of our training hyperparameters, and report additional recall@k metrics. We then provide additional qualitative examples of the interpolation performed by our model. Finally, as a sanity check, we study the generality of our approach by reporting performance on the standard *mini*ImageNet benchmark.

1. Per-class Accuracy

In Figure 1 we provide a class-wise performance comparison on the Dermnet dataset between the PCN and FT_{200}-CE models as a scatter plot (shown here for the best performing PCN model evaluated with a train shot of 10 with $\text{mca}_{\text{base+novel}} = 50.92$).
We make the following observations: Overall metrics indicate that FT_{200}-CE demonstrates slightly stronger average performance on base classes. For a large fraction of the base classes, both methods have similar performance. For the ones in which PCN performance is lower, there is usually a reasonable lower bound on the classification accuracy. In contrast, for novel classes, PCN performs better on average. Importantly, when FT_{200}-CE performance is lower than PCN, it is usually signif-

Approach	α	$\text{mca}_{\text{base+novel}}$
PCN	0	45.62 ± 0.89
PCN	0.5	47.49 ± 0.71
PN	0	44.35 ± 0.53
PN*	0.5	45.84 ± 0.46

Table 1: Importance of episodic memory

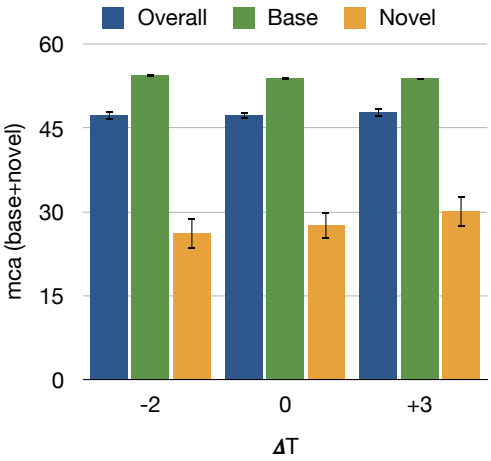

Figure 2: Effect of temperature on PCN.

	Approach	$\text{r@5}_{\text{base+novel}}$	r@5_{base}	$\text{r@5}_{\text{novel}}$	$\text{r@10}_{\text{base+novel}}$	$\text{r@10}_{\text{base}}$	$\text{r@10}_{\text{novel}}$
n=5	FT_{200}-CE	**77.7** $\pm$ **0.8**	**80.8** $\pm$ **0.8**	38.0 ± 2.1	84.9 ± 0.5	**88.1** $\pm$ **0.4**	45.3 ± 1.5
	PN	70.9 ± 0.4	71.5 ± 0.3	**63.2** $\pm$ **2.3**	80.2 ± 0.3	80.5 ± 0.2	**76.3** $\pm$ **2.1**
	PCN (ours)	**77.7** $\pm$ **0.2**	**79.2** $\pm$ **0.2**	**59.2** $\pm$ **2.6**	**86.0** $\pm$ **0.4**	87.1 ± 0.2	**72.2** $\pm$ **3.6**
n=10	FT_{200}-CE	**78.6** $\pm$ **0.1**	**80.2** $\pm$ **0.3**	58.0 ± 3.2	**86.4** $\pm$ **0.3**	**88.1** $\pm$ **0.1**	65.9 ± 2.6
	PN	69.4 ± 0.3	69.1 ± 0.3	**72.2** $\pm$ **1.7**	78.6 ± 0.3	78.1 ± 0.3	**84.7** $\pm$ **2.2**
	PCN (ours)	76.3 ± 0.2	76.4 ± 0.2	**74.5** $\pm$ **2.6**	85.0 ± 0.2	85.0 ± 0.3	**85.0** $\pm$ **2.0**

Table 2: Recall@k on top 200 classes.

icantly lower. As an example is the novel class 'distal splitting hang nail' for which PCN performs significantly better.

2. Role of Hyperparameters

Importance of Temperature: Fig. 2 reports performance by varying $\Delta_\tau = \tau_{test} - \tau_{train}$, the difference in temperature used in test versus train time. We can see that while performance is agnostic for base classes, higher interpolation through an increased temperature leading to $\Delta_\tau > 0$ leads to improved performance on novel classes. Conversely, when $\Delta_\tau < 0$, performance drops. This suggests that at inference time, the model requires interpolating between the cluster prototypes to effectively predict a class label.

Does episodic memory help? Table 1 shows that we can get improvements even with a simple online update rule that blends prototypes computed using the support set in the current episode with the past, using $\alpha = 0.5$. This trend is also seen for prototypical networks (denoted by PN*). We leave as future work the task of modeling adaptive α.

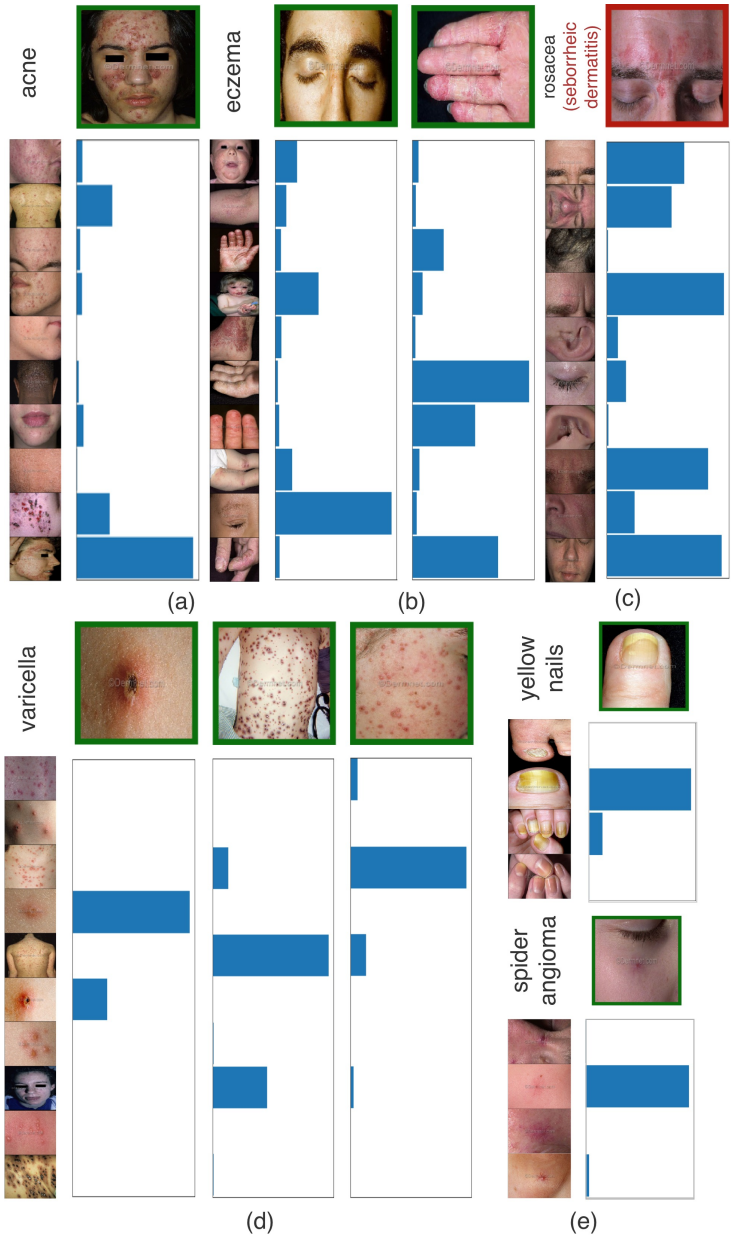

Figure 3: Effectiveness of using multiple clusters. Shown for base and novel classes. (a), (b), (d), (e): Examples from test set that are correctly classified by PCN. For each class, we show the nearest neighbor to the learned prototypes. We also present examples (columns) whose labels are correctly predicted and the inferred cluster responsibilities $q(z|c, x)$ conditioned on the correct class. (c): Test set example that is misclassified by PCN. Correct label is shown in black, while the incorrect prediction is shown in red. We show the nearest neighbors to the learned cluster prototypes of the *predicted* (incorrect) class, and the corresponding cluster responsibilities. Note that green outlines around query images denote correct classification while red denotes incorrect classification. (e) corresponds to novel classes.

3. Additional Metrics

In Table 2 we provide recall@5 and recall@10 metrics for PN, PCN, and FT_{200}-CE approaches, for train shots $n = 5$ and $n = 10$. PCN performs on par with the FT_{200}-CE baseline and outperforms PN on these metrics. Additionally, PCN and PN models dominate in recall@k on novel classes.

4. Qualitative Examples

Sec 3, we provide additional qualitative examples of how PCN can interpolate between the learned prototypes by modulating $q(z|c, x)$. As before, we show query images from the test set for various classes, with a mix of correct and incorrectly classified examples. Below the class label, we show the nearest neighbor image from the training set to each of the learned prototypes for the class *predicted* by PCN, and below each query image, cluster responsibilities placed by the model on each of these prototypes. We can see that the $q(z|c, x)$ distribution varies quite a bit across examples, being a lot more diffuse in some cases than others. For instance, while classes in the first row see relatively diffuse responsibility distributions, the distribution is far more peaked for the varicella class in (d). Overall, it can be seen that the model learns to accurately place probability mass on similar prototypes. Figure 3(c) shows an incorrectly predicted example. In Figure 3(e), we show similar examples for novel classes.

5. Applicability to *mini*ImageNet

As a *sanity check*, we evaluate our approach on the *mini*ImageNet dataset (Vinyals et al., 2016). We use the splits proposed in Ravi and Larochelle (2016), and match our implementation and training to prior work (Snell et al., 2017; Vinyals et al., 2016). Our embedding network is a CNN architecture comprised of four convolutional blocks, that leads to 1600-dimensional embeddings.

	n=1		n=5	
	5-way	20-way	5-way	20-way
PN	56.58	44.43	70.83	48.29
PCN	56.70	44.97	74.93	53.49

Table 3: *mini*ImageNet accuracy on $K_{base_2} \cup K_{novel-test}$

Recall that our primary objective is learning *without forgetting*. To evaluate this, we modify the existing *mini*ImageNet benchmark as follows: Similar to Hariharan and Girshick (2017), we train a base classifier and validate it on heldout data from base classes K_{base}(representation learning phase). Next, in the low-shot learning phase, we split K_{base} into disjoint subsets $K_{base_1} \cup K_{base_2}$. We cross-validate hyperparameters on $K_{base_1} \cup K_{novel-val}$ classes, and report averaged accuracy over test 600 episodes on $K_{base_2} \cup K_{novel-test}$ classes. Note that as before, both PN and PCN get access to the appropriate subset of the entire training set to generate prototypes for base classes.
Table 3 presents results of various classification settings, with a test-query size of 15 and averaged over 600 test episodes. As seen, PCN strongly outperforms PN in the 5-shot case and has similar performance in the 1-shot case, where interpolation for novel classes is not possible. Our best performing PCN model is trained with 5 clusters per class, and uses 1 cluster for novel classes and a temperature value of 4.0 (all values picked via grid search).

	5-shot 5-way
Baseline-NN (Ravi and Larochelle, 2016)	$51.0 \pm 0.7\%$
MN (Vinyals et al., 2016)	$55.3 \pm 0.7\%$
MAML (Finn et al., 2017)	$63.1 \pm 0.9\%$
PFA-Simple (Qiao et al., 2017)	$\mathbf{67.9 \pm 0.2\%}$
PN (Snell et al., 2017)	$65.3 \pm 0.7\%$
PCN (ours)	$\mathbf{66.9 \pm 0.7\%}$

Table 4: Accuracy on *mini*ImageNet. We report mean accuracies with 95% confidence intervals. Note that PFA-Simple corresponds to the 'Ours-Simple' model reported in Qiao et al. (2017).

Further, we report numbers on the *standard mini*ImageNet benchmark in Table 4, and compare against recently proposed work. We note that while several other approaches have also been proposed that report performance using deeper residual features, for fairness we only compare against approaches that use identical CNN architectures. We observe that our model outperforms several prior approaches and is competitive with the current state of the art. As with prior work, we train our model by monitoring generalization performance on unseen novel classes from the validation set, and report performance on 5-shot-5-way performance averaged over 600 episodes from the test split. We train with a way of 20 at train time. We use the identical CNN architecture as in Snell et al. (2017), with 4 convolutional blocks, leading to 1600-dimensional embeddings for the 84x84 images in the *mini*ImageNet dataset. Each block is structured as follows: a 64-filter 3x3 convolution layer, batch normalization layer, ReLU nonlinearity, and 2x2 max-pooling layer. We train our models via SGD with Adam (Kingma and Ba, 2014), starting with a learning rate of 1e-3 that we anneal by half every 2000 episodes. Note that we our PCN model is built on top of our PN reimplementation, and for fair comparison we report performance with our implementation of Prototypical Networks in row 3, which achieves slightly lower performance than that reported in Snell et al. (2017).

References

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Bharath Hariharan and Ross B Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, pages 3037–3046, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan Yuille. Few-shot image recognition by predicting parameters from activations. *arXiv preprint arXiv:1706.03466*, 2, 2017.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.