Proceedings of Machine Learning Research 140:77–89, 2021 AAAI Workshop on Meta-Learning and MetaDL Challenge

Fuzzy Simplicial Networks: A Topology-Inspired Model to Improve Task Generalization in Few-shot Learning

Henry Kvinge[1] HENRY.KVINGE@PNNL.GOV
Zachary New[1] ZACHARY.NEW@PNNL.GOV
Nico Courts[1,2] NICOLAS.COURTS@PNNL.GOV
Jung H. Lee[1] JUNG.LEE@PNNL.GOV
Lauren A. Phillips[3] LAUREN.PHILLIPS@PNNL.GOV
Courtney D. Corley[3] COURT@PNNL.GOV
Aaron Tuor[1] AARON.TUOR@PNNL.GOV
Andrew Avila[3] ANDREW.AVILA@PNNL.GOV
Nathan O. Hodas[3] NATHAN.HODAS@PNNL.GOV
[1] *Pacific Northwest National Laboratory, Seattle, Washington, USA*
[2] *Department of Mathematics, University of Washington, Seattle, Washington, USA,*
[3] *Pacific Northwest National Laboratory, Richland, Washington, USA*

Editors: Isabelle Guyon, Jan N. van Rijn, Sébastien Treguer, Joaquin Vanschoren

Abstract

Deep learning has shown great success in settings with massive amounts of data but has struggled when data is limited. Few-shot learning algorithms, which seek to address this limitation, are designed to generalize well to new tasks with limited data. Typically, models are evaluated on unseen classes and datasets that are defined by the same fundamental task as they are trained for (e.g. category membership). One can also ask how well a model can generalize to fundamentally different tasks within a fixed dataset (for example: moving from category membership to tasks that involve detecting object orientation or quantity). To formalize this kind of shift we define a notion of "independence of tasks" and identify three new sets of labels for established computer vision datasets that test a model's ability to generalize to tasks which draw on orthogonal attributes in the data. We use these datasets to investigate the failure modes of metric-based few-shot models. Based on our findings, we introduce a new few-shot model called Fuzzy Simplicial Networks (FSN) which leverages a construction from topology to more flexibly represent each class from limited data. In particular, FSN models can not only form multiple representations for a given class but can also begin to capture the low-dimensional structure which characterizes class manifolds in the encoded space of deep networks. We show that FSN outperforms state-of-the-art models on the challenging tasks we introduce in this paper while remaining competitive on standard few-shot benchmarks.

1. Introduction

Traditionally deep learning requires large amounts of labelled data to build models that do not overfit to their training set (LeCun et al., 2015). However, preparing sufficient amounts of data can be costly, and in many applications impractical, limiting deep learning's utility. To address this challenge, the area of few-shot learning aims to develop methods that leverage the strengths of deep learning to solve problems where one may only have a handful of examples from each class.

Many of the most effective models in few-shot learning fall into the family of metric-based methods. Notable examples of such models include Prototypical Networks (Snell et al., 2017) and Matching Networks (Vinyals et al., 2016). These models rely on an encoder function (usually a deep network) that learns to extract rich features from data while being trained on a related task. At inference time the encoder function maps instances of new classes into the learned feature space and builds a class representation from them. Predictions are then made by comparing the image of unlabeled instances in the encoded space with each of the class representations. Prototypes can then be hard-coded as simple geometric structures such as a centroid (Snell et al., 2017).

While these models have proven to be remarkably successful in many contexts, investigations into their effectiveness have mostly focused on cases where the tasks that the models are evaluated on are broadly similar to those that they trained on (for instance, evaluating the performance of a model on Caltech-UCSD Birds 200 (Welinder et al., 2010) when it was trained on ImageNet (Deng et al., 2009)). In this paper we are interested in understanding how metric-based models handle more challenging tasks with the ultimate goal of understanding how to make them even more responsive and flexible to new examples given at test time. To this end we introduce three new label sets for well-known computer vision datasets. These labels are easy for a human to understand and predict and they draw on many of the same types of features that are useful in class membership tasks (such as edges, texture, and shape). Importantly though, our labels are "independent" of the original labels, a notion that we describe in Section 3. Unsurprisingly, we find that the metric-based models that we evaluated perform very poorly on these tasks that are independent of the biases formed during training on the ImageNet classification task.

While these results might suggest that our only hope is to re-train the encoder at inference time, we find that even when faced with these challenging label sets the encoder often extracts features that can discriminate between classes. In fact, our analysis suggests that often the class representations themselves fail to capture the relevant features, and that instead unrelated features overwhelm them. This suggests revisiting the kind of representations that we use in our models. Drawing inspiration from a geometric structure known as a simplicial complex, we propose a new model which we call *Fuzzy Simplicial Networks (FSN)*. Simplicial complexes can approximate almost all geometric structures arising in nature while at the same time being built from a simple building block: the simplex. Given that they can approximate spaces much more flexibly than centroids or subspaces for example, they are an ideal candidate for class representations.

We show that FSN significantly outperforms other metric-based models (with different representations but the same base encoder architecture) on the challenging label sets we introduce below, after being trained on ImageNet. Following insights into few-shot model evaluation found by Triantafillou et al. (2019), we also show that under the same conditions FSN displays strong generalization performance across a diverse range of other datasets.

In summary, our contributions in this paper include the following.

- A description of three new label sets for existing computer vision datasets. These new labels allow a few-shot model to be tested on tasks that are independent of the type it was trained for.

- We analyze why these datasets are challenging, using Prototypical networks as a case study.
- We introduce a new metric-based few-shot model called Fuzzy Simplicial Networks which models classes as a novel structure called a fuzzy simplicial complex that we define in this paper.

2. Background and Related Work

2.1. Few-shot Learning

There are a number of different approaches to few-shot learning. Fine-tuning methods (Chen et al., 2019) train a model on a surrogate dataset and then fine-tune on a small number of examples. Data augmentation methods (Hariharan and Girshick, 2017) produce additional examples of a class through augmentation and other methods. Gradient-based meta-learning (Finn et al., 2017; Nichol et al., 2018) is a class of sophisticated methods that optimize specifically for model parameters that are easily updated during fine-tuning for each few-shot episode. Metric-based models learn an encoding of the data into a space where the task can be solved using notions of distance or similarity between labeled and unlabeled examples. In this paper we choose to focus on metric-based methods since we are interested in understanding and improving how classes are represented in this encoded space.

We work within the standard few-shot framework (Vinyals et al., 2016) where a classification task, or *episode*, consists of a *support set* of labeled examples

$$S = \{(x_1, y_1), \ldots, (x_r, y_r)\}$$

where x_i is the datapoint and y_i is the label belonging to classes $C = \{c_1, \ldots, c_k\}$ and a *query set* Q of unlabeled examples also belonging to classes from C.

We write S_i for the subset of S containing only examples with label c_i. In this paper we will always assume that $|S_i| = n$ is fixed for all $1 \leq i \leq k$. The number n is known as the *shots* of the episode, while k (the number of different classes) is known as the *ways*. This is often written as *n-shot k-way*. By *few-shot training* and *few-shot evaluation* we mean the process of iterative training/testing by episode.

2.2. Metric-based Models

Metric-based few-shot learning algorithms can be very roughly decomposed into three fundamental components: (i) an encoder function $f_\theta : X \to \mathbb{R}^m$ that takes data from a space X and maps it into a feature (or encoded) space $\mathbb{R}^m$, (ii) a method of representing encoded points from a support set class, $f_\theta(S_i)$, as a single coherent representation γ_i, and (iii) a distance function $d : \mathbb{R}^m \times \Gamma \to \mathbb{R}_{\geq 0}$, where Γ is the set of all possible representations of the given type associated with the model. Given a query point $q \in Q$, the model predicts q to belong to class c_t if $t = \arg\min_{1 \leq i \leq k} d(f_\theta(q), \gamma_i)$.

Probably the most notable example of this type of model is Prototypical Networks (Snell et al., 2017) (or ProtoNets). In this case γ_i is the centroid of the encoded points $f_\theta(S_i)$, and d is the standard Euclidean distance in the encoded space. Prototypical Networks have proven to be a simple but robust model that has been important in the development of few-shot learning more broadly. In Deep Subspace Networks (Simon et al., 2020) and Regression Networks (Devos and Grossglauser, 2019), γ_i is a low-dimensional affine subspace that approximates $f_\theta(S_i)$ and d is the usual distance between a point and an affine subspace induced by the Euclidean metric.

The models above form a single representation for all elements in a support class. As we will show below, there are times where one would like multiple representations for multiple clusters within $f_\theta(S_i)$. A simple approach to this is to set γ_i to be the full support set and d to be some flavor of nearest neighbor distance (Wang et al., 2019). Another more recent approach generalizes Prototypical Networks to allow for multiple centroids to represent a single class (Allen et al., 2019).

Finally, in the work of Zhang et al. (2018) γ_i is the k-simplex whose vertices are formed by the elements of $f_\theta(S_i)$ (see Section 4 for a refresher on simplices) and d is a distance function obtained by calculating the quotient of the volume of the $(k+1)$-simplex with vertices $f_\theta(S_i)$ and q over the volume of the simplex whose vertices are the points $f_\theta(S_i)$ without q. This approach assumes that classes can be represented by a single, connected, convex structure. However, a single simplex does not provide the flexibility for multiple representations within a support class, a feature that was shown to be important by Allen et al. (2019).

The present work attempts to leverage many of the advantages of the above models while avoiding their limitations.

2.3. Topological Data Analysis

One domain in which simplicial complexes play a leading role is in topological data analysis (TDA). The idea of persistence homology pioneered by Edelsbrunner et al. (2000); Zomorodian and Carlsson (2005) proceeds by computing the homology of a series of Vietoris-Rips complexes (which can be roughly interpreted as simplicial complexes assigned to a point cloud) in order to understand the topology of a dataset. To ensure our models can be trained efficiently, the flavor of simplicial complex we use in this paper differs substantially from those in the TDA literature.

3. Independence of Labels

In order to better motivate the datasets that we will introduce in this section, we first describe a notion of independence between sets of labels attached to a dataset. Note that this idea is simply the usual property of independence from probability theory applied to two possibly distinct sets of labels attached to a single dataset.

Definition 1 *Suppose that $\ell : D \to C = \{c_1, \dots, c_\ell\}$ and $\tilde{\ell} : D \to \widetilde{C} = \{\tilde{c}_1, \dots \tilde{c}_t\}$ are two labeling functions on a dataset D. We say that ℓ and $\tilde{\ell}$ are* independent labelings *on D if for any randomly chosen $x \in D$, and $c \in C$ and $\tilde{c} \in \widetilde{C}$,*

$$p\big(\ell(x) = c, \tilde{\ell}(x) = \tilde{c}\big) = \\ p\big(\ell(x) = c\big)p\big(\tilde{\ell}(x) = \tilde{c}\big).$$

We say that two tasks T and $\widetilde{T}$ on X are independent *if their corresponding label sets ℓ and $\tilde{\ell}$ are independent.*

We now describe three new sets of labels for existing computer vision datasets where each new label set is designed to be (approximately) independent from the original label set. We measure this independence via the metric of mutual information between the original and new label set. A pair of variables is independent if and only if the mutual information between the pair is zero.

3.0.1. Stem/No-stem (SNS) Dataset

The *Stem/No-stem dataset* is a re-labeling of the *Fruits 360 dataset* (Mureşan and Oltean, 2018). The original dataset contains images of different types of fruit with the only variation between images being fruit type and fruit orientation. While the original labels classified images by fruit type, our SNS labels instead focus on orientation and, in particular, whether or not the stem or blossom node are facing the camera (see Figure 1). The mutual information between the labels on SNS and Fruit 360 is 0.031. For reference, the mutual information between the Fruits 360 labels and random binary labels is .015, while the mutual information between the Fruits 360 labels and labels based on the first letter of the fruit type (which are highly correlated) is 2.318. Note that random labels should by construction have close to zero mutual information. Deviation from 0 only arises as an artifact of sampling. We would expect similar values for the mutual information of random labels applied to the subsequent datasets.

3.0.2. Back/No-back (BNB) Dataset

The *Caltech-UCSD Birds 200 dataset* (Welinder et al., 2010) is a common benchmark dataset for few-shot learning featuring images of birds. The original labels correspond to the species of bird. In order to assign new "Back" and "No-back" labels, based on whether the back of the bird is visible, we use the visibility attribute within the "parts" metadata associated with this dataset. The mutual information between the labels on BNB and Birds 200 is 0.043.

3.0.3. One/Many (OM) Dataset

The *Stanford Dogs Dataset* (Khosla et al., 2011) is another commonly used dataset in computer vision which involves predicting the breed of a dog. The authors of the dataset tag each dog in the image separately, so we were able to extract the number of dogs in each image. We used this information to construct labels 'One' and 'Many' for a subset of breeds based on the number of dogs in the image. The mutual information between the labels on OM and Stanford dogs is 0.001.

3.1. What Makes These Datasets Difficult?

In this section we ask why a ProtoNet model trained on ImageNet struggles with these datasets by focusing on stem/no-stem. A ProtoNet model with few-shot training on ImageNet achieves 71.2% accuracy for the binary SNS task when given 5-shots. On the other hand, the same model achieves 96.4% accuracy on the 5-shot, 5-way Fruits 360 task. Visualizations of encoded SNS/Fruits 360 images suggest that an encoder pretrained on ImageNet has a strong bias toward grouping the images by type of fruit. Below we explore an array of additional explanations for the difficulty of this task:

1. ***Hypothesis:*** *A ProtoNets model trained on ImageNet does not extract the features required to solve the stem-no-stem problem.*

   The model can easily differentiate between SNS within a particular fruit cluster. For example, when restricted to images from the *Apple red 2* class or *Green pepper* class the model achieves 98.7% and 99.3% accuracy on the SNS task respectively. This shows that at least locally (that is, within a cluster), the model extracts high quality features that can be used for discriminating between stem/no-stem images.

2. ***Hypothesis:*** *A ProtoNets model trained on ImageNet is able to extract discriminative features within clusters, but these are not sufficient to differentiate between the classes globally.*

   The encoding of fruit images obtained from our ProtoNet model is linearly separable with respect to SNS labels. Indeed, using a linear support vector machine model we were able to find a hyperplane in the feature space which separated stem and not-stem points

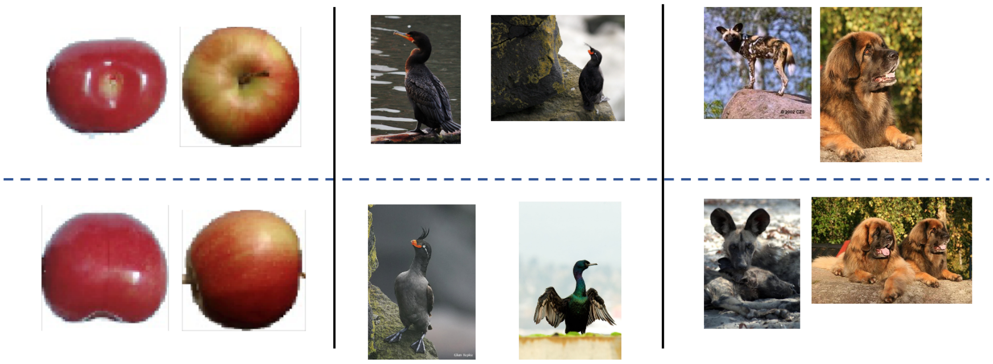

Figure 1: Examples of the stem/no-stem (left), back/no-back (center), and one/many (right).

.

with 100% accuracy. Furthermore, this was not simply due to the high dimension (2048) of the encoded space. While a random binary labeling of a random sampling of Gaussian points in $\mathbb{R}^{2048}$ with covariance similar to encoded SNS is also linearly separable, the margin is much less significant (a .858 margin for real SNS points and a .004 margin for random labels on random points).

3. ***Hypothesis:*** *The encoder in our ProtoNets model has a strong bias toward extracting features that separate the fruit images by type. This separation tends to overwhelm the features salient to the stem-no-stem task.*

   To test this we altered our already trained ProtoNets model so that it mean centers all points corresponding to a given type of fruit in the encoded space, removing the separation between clusters. We found that doing this improved the accuracy by nearly 5%, indicating that the bias toward separating fruit by type interferes with other tasks.

4. ***Hypothesis:*** *Centroids fail to capture the lower dimensional structure of a class in encoded space.*

   Using centroids to represent a class makes sense if points from the class actually follow either a Gaussian or some other distribution that has the same intrinsic dimension as the ambient space. The singular values of points from the SNS dataset in the encoded space (which decay rapidly) suggest that the dataset is actually better approximated by a lower dimensional structure.

The observations above suggest using a more flexible and adaptive framework for building representations which is able to account for multiple representations of a class and also able to model the lower-dimensional structure of the data manifolds on which encoded classes sit.

4. Simplices, Simplicial Complexes, and Fuzzy Simplicial Complexes

Simplicial complexes have a long history in mathematics due to the fact that they can effectively approximate a broad range of geometric structures even though they are built from extremely simple constituent parts: simplices. For $k \leq m$ a *k-dimensional simplex* or *k-simplex* in $\mathbb{R}^m$, Σ^k, is the convex hull of $k+1$ (affinely independent) points $x_0, \ldots, x_k \in \mathbb{R}^m$.

Simplices of dimensions 0, 1, 2, and 3 will already be familiar to the reader as points, line segments, triangles, and tetrahedrons. One of the key properties of a k-simplex Σ^k on vertices $x_0, \ldots, x_k$ is that the convex hull of any subset of $\ell+1 \leq k+1$ of these vertices, $x_{i_0}, \ldots, x_{i_\ell}$, is itself an ℓ-simplex known as a *face of* Σ^k. Thus Σ^k has, as subsets, $2^{k+1}-1$ nonempty simplices/faces (of dimensions 0 through k) corresponding bijectively to all non-empty subsets of $\{x_0, \ldots, x_k\}$. Abusing notation, we write $\Sigma^k = \{x_0, \ldots, x_k\}$. The volume of Σ^k can be calculated as the square root of the determinant of $A^T A$ (where A is the matrix whose columns are $x_1 - x_0, \ldots, x_k - x_0$), normalized by $\frac{1}{k!}$.

Let $\Sigma^k = \{x_0, \ldots, x_k\}$ be a k-simplex and q any point in its ambient space. We define the *subspace distance* $d_{\text{sub}}(\Sigma^k, q)$ to be the Euclidean distance between q and its projection onto the affine subspace based at x_0 and spanned by the vectors $x_1 - x_0, \ldots, x_k - x_0$. Note that this definition is invariant under a relabelling of the vertices of Σ^k.

A *simplicial complex* C is a collection of simplices of varying dimensions, where individual simplices may be glued together along shared faces. A simplex Σ in C is called a *facet* if it is not a face of a higher dimensional simplex in C.

To adapt simplicial complexes to model real data which can be noisy, we define the notion of a *fuzzy simplicial complex* (which was inspired by the use of fuzzy simplicial sets of McInnes et al. (2018)). This in turn was inspired by fuzzy sets, a generalization of sets, where the extent to which an element x belongs to fuzzy set U is measured by a membership function $m : U \to [0,1]$, with $m(x) = 0$ denoting that $x \notin U$ and $m(x) = 1$ corresponding to $x \in U$.

Definition 2 *Given a set of points $U = \{x_1, \ldots, x_t\}$, a* fuzzy simplicial complex *on U denoted by $C = (G(U), m)$ consists of the set $G(U)$ of all simplices that can be constructed from points in U as well as a membership function $m : G(U) \to [0,1]$ that determines the extent to which each simplex in $G(U)$ belongs to C.*

Given that for even a small set U, $G(U)$ is very large, in practice we will work with fuzzy simplicial complexes where we assume that facets have fixed dimension k, and only work with these simplices in our calculations. We let $G_k(U)$ then denote the set of all k-dimensional simplices that can be formed from points in U. We then calculate the distance between fuzzy simplicial complex $C = (G_k(U), m)$ and a query point q as:

$$d_{\text{fuzz}}(C, q) := \sum_{\Sigma \in G_k(U)} m(\Sigma) d_{\text{sub}}(\Sigma, q). \quad (1)$$

5. Fuzzy Simplicial Networks

In this section we introduce a class of models we call *Fuzzy Simplicial Networks* (FSNs). These are metric-based few-shot models which use fuzzy simplicial complexes as representations of support classes. A FSN consists of three components: an encoder function f_θ, a method for building a fuzzy simplicial complex C_i for each encoded support set class $f_\theta(S_i)$, and a method for measuring the distance between an unlabeled query point $f_\theta(q)$ and each C_i. We chose the second and third of these so that they are differentiable and the entire model can be trained episodically in an end-to-end manner using backpropogation.

To improve training and inference speed and avoid memory issues, we make free use of the approximations introduced at the end of Section 4. Specifically, a top dimension k is fixed for simplices in all C_i and then all our calculations only include these k-dimensional facets in each $G_k(f_\theta(S_i))$. To choose k, we ran a hyperparameter sweep in which we found that, generally, restricting to simplices of around dimension $|S_i|/2$ gave the best learning and evaluation performance on ImageNet. This value of k allows for the maximal amount of possible simplices and thus presumably greater expressivity. This expressivity, however, comes at the cost of significantly higher memory usage and our final choice of hyperparameter (k=8) reflects this.

In order to obtain a fuzzy structure on $G_k(f_\theta(S))$ we need a membership function. We define V_k to be the function from all k-simplices in $\mathbb{R}^m$ to $\mathbb{R}_{\geq 0}$ such that for a k-simplex Σ^k, $V_k(\Sigma^k) := 1/\mathrm{vol}(\Sigma^k)$. We use V_k as the basis for our membership function m under the logic that a simplex with large volume has at least one point that is distant from the others indicating that we should have less certainty that this simplex actually captures the structure of the class. Thus for $\Sigma^k \in G_k(f_\theta(S_i))$ we set

$$m(\Sigma^k) := \frac{V_k(\Sigma^k)}{\sum_{\Sigma^{k\prime} \in G_k(f_\theta(S_i))} V_k(\Sigma^{k\prime})}. \tag{2}$$

Once the hyperparameter k (the dimension of simplices to be used) has been fixed, the FSN model proceeds with inference as follows. The encoder function f_θ maps all support set classes $S_1, \ldots, S_r$ and query q into the encoded space. For each S_i, all k-simplices $G_k(f_\theta(S_i))$ are extracted and the a membership function $m_i : G_k(f_\theta(S_i)) \to [0,1]$ as defined above is calculated. The distance function d_{fuzz} is used to calculate the fuzzy simplicial complex C_t that $f_\theta(q)$ is "closest to". Query q is then predicted to belong to class t.

We note that one could use statistics other than volume to define the membership function m. In fact, we also tested models that learned to compute uncertainties from a small fully-connected network that took as input the Gram matrix associated to all simplex vertices shifted to the origin. We denote this type of model as *FSN Learned*. In general we found that the models where the uncertainty calculation was hard-coded performed better than when uncertainty calculation was learned.

While our model makes significant gains compared to other few-shot models it also has a few limitations. FSN models take up a larger memory footprint when compared to ProtoNets. FSNs exhibit polynomial memory growth when the number of shots or the number of ways is increased. Additionally, compared to ProtoNets, there is increased complexity in the class representations making it harder to interpret why the model would make a particular choice. FSNs also lose the ability to compare distances between class representations via the same metric used to compute distances between query points.

6. Experiments

We are primarily interested in how different class representations can leverage the features extracted from a strong encoder, even when the task they are evaluated on is very different from the one that they were trained for. Thus we trained and validated all the models in our experiments on a few-shot version of ImageNet and then tested on a diverse range of datasets (without further training). Training was performed in an episodic manner. We trained and tested ProtoNets (Snell et al., 2017), nearest neighbor based models, Simplex (Zhang et al., 2018), Deep Subspace

	Stem/	Back/	One/
	No-stem	**No-Back**	**Many**
ProtoNet	73.2±0.2	57.1±0.3	54.8±0.4
Nearest Neighbor	74.0±0.5	56.7±0.2	55.5±0.3
Simplex	75.4±0.2	57.7±0.3	54.5±0.2
Subspace	72.7±0.6	57.1±0.2	53.7±0.2
FSN (Ours)	**77.9±0.3**	**59.2±0.3**	**58.0±0.2**
FSN Learned (Ours)	75.7±0.7	**58.8±0.2**	56.6±0.5

Table 1: Accuracy comparisons across the three challenging label sets introduced above in the 10-shot, 2-way regime.

Networks (Simon et al., 2020), and two versions of our FSN (one that uses the hard-coded volume based weighting of simplices and one that learns a weighting as described in Section 5).

In our experiments all models used a ResNet50 encoder (He et al., 2016) with the final layer removed as the base encoder and were initialized (prior to training) with the pre-trained weights available through the TorchVision library (Marcel and Rodriguez, 2010). Thus the only part of each model that differed was the class representation and distance used in the encoded space. Note that the use of the larger ResNet50 encoder differs from most few-shot learning experiments which leverage smaller encoders (Snell et al., 2017; Finn et al., 2017; Nichol et al., 2018). We elected to run experiments with a larger encoder to ensure our feature vectors captured as much relevant information from the training task as possible and were not limited by encoder size.

7. Results

All models were evaluated a total of 20 times on each dataset. Results reported are means and 95% confidence intervals computed under the assumption that the data was distributed normally around the true value. The results of tests on our novel label sets are found in Table 1. Our FSN model using simplex volume to measure membership outperforms all other models we tested, with a maximum margin of 2% between the confidence intervals for our model and the next best model on One/Many. The fixed FSN also outperforms the variant using a learned membership function, although this model still performs strongly when compared to the non-FSN models.

Although the FSN model performs well on tasks such as SNS, one might wonder whether FSN still performs well on more traditional few-shot learning tasks. To evaluate this, we take our models trained on ImageNet and evaluate them on 13 datasets. In this way we are able to assess whether the FSN representation also supports generalization to other datasets as in Triantafillou et al. (2019). Table 2 contains those datasets where FSN did better than all other models. Table 3 gives the results for those datasets where our models did not outperform others. We note that in all cases our model was within 1% of the accuracy of the of top per-

	Omniglot	Adience Faces	Aircraft	Describable Textures	Buildings	Fruits 360	Plant Seedlings
ProtoNet	93.0±0.2	65.7±0.3	55.7±0.6	84.2±0.2	96.2±0.2	99.2±0.0	78.1±0.7
NearestNeighbor	89.5±0.2	62.6±0.3	49.1±0.3	76.9±0.3	93.2±0.3	99.4±0.1	74.2±0.5
Simplex	91.4±0.1	66.3±0.3	52.7±0.2	82.3±0.2	96.7±0.2	99.6±0.0	80.0±0.5
Subspace	91.7±0.2	65.7±0.4	55.0±0.3	83.0±0.1	95.9±0.2	99.6±0.0	78.1±0.4
FSN (Ours)	**94.8±0.3**	**71.7±0.3**	**59.1±0.2**	**85.2±0.2**	**97.9±0.1**	**99.7±0.0**	**88.7±0.5**
FSN Learned (Ours)	**94.6±0.2**	70.7±0.2	**58.9±0.4**	**84.8±0.2**	**98.1±0.1**	**99.7±0.0**	**87.4±0.4**

Table 2: Datasets where our models outperform all models evaluated. Accuracies were measured in the 10-shot, 5-way regime. Our benchmark datasets include Omniglot (Lake et al., 2015), Adience Faces (Eidinger et al., 2014), FGVC Aircraft (Maji et al., 2013), Describable Textures (Cimpoi et al., 2014), Urban Buildings for Image Retrieval (Niafas, 2016), Fruits 360, Plant Seedlings (Giselsson et al., 2017).

	ImageNet	CIFAR100	CIFAR100 Superclass	Cars	Birds	Dogs
ProtoNet	98.2±0.0	84.2±0.5	82.0±0.3	**78.7±0.2**	**92.8±0.1**	97.2±0.1
NearestNeighbor	97.5±0.0	83.7±0.3	80.3±0.4	72.3±0.3	90.2±0.1	96.9±0.1
Simplex	97.3±0.0	**86.4±0.2**	83.2±0.2	70.5±0.3	88.0±0.1	95.8±0.1
Subspace	**98.4±0.0**	**86.5±0.2**	**84.4±0.3**	77.5±0.4	92.5±0.1	**97.5±0.1**
FSN (Ours)	98.2±0.1	86.1±0.1	**84.4±0.3**	**78.7±0.4**	**92.6±0.2**	96.9±0.1
FSN Learned (Ours)	98.1±0.1	85.9±0.2	**84.0±0.2**	77.9±0.4	**92.5±0.2**	97.0±0.0

Table 3: Datasets where FSN and FSN Learned performed either as well as or less well than other models evaluated. Accuracies were measured in the 10-shot, 5-way regime. Our benchmark datasets include ImageNet, CIFAR100 and CIFAR100 Superclass (Krizhevsky et al., 2009), Stanford Cars (Krause et al., 2013), Caltech-UCSD Birds-200-2011, and Stanford Dogs.

forming model. While we find that performance on ImageNet itself (and other datasets with large overlap with ImageNet such as Birds, Dogs, and Cars) does not improve, FSN shows a strong advantage on datasets that are very distinct from ImageNet including Adience Faces (5.4% better), FGVC Aircraft (3.4% better), and Plant Seedlings (7.5% better). This points to FSN being better at generalizing to new datasets, while still remaining competitive on fine-grained tasks similar to those encountered during training.

Together our results strongly suggest that fuzzy simplicial complexes are a more flexible and adaptive representation that captures the structure of different possible support classes. Our results also suggest that building a membership function based on simplex volume is more robust than trying to learn a weighting from the structure of individual simplices themselves.

8. Conclusion

In this paper we studied the performance of metric-based few-shot models when they are evaluated on tasks that are significantly different from those that they were trained to solve. Our goal was to find representations that can better model support classes. We showed that even when the encoder function extracts the features needed to distinguish between classes in a support set, the class representatives can fail to capture these. We introduced three new label sets for existing computer vision datasets which are approximately independent from their original labels which we used to evaluate how well models could capture novel class structure. Our analysis of the failures of existing models on these datasets motivated the introduction of our model, FSN. We showed that FSN not only achieves significantly higher average accuracy on the new label sets when compared to a selection of other metric-based few-shot models, but also outperforms or is competitive with these models on common few-shot benchmark datasets.

Acknowledgments

This work was funded by the U.S. Government.

References

Kelsey Allen, Evan Shelhamer, Hanul Shin, and Joshua Tenenbaum. Infinite mixture prototypes for few-shot learning. In *International Conference on Machine Learning*, pages 232–241, 2019.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv:1904.04232*, 2019.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

Arnout Devos and Matthias Grossglauser. Subspace networks for few-shot classification. *arXiv:1905.13613*, 2019.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000.

Eran Eidinger, Roee Enbar, and Tal Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12): 2170–2179, 2014.

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-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Thomas Mosgaard Giselsson, Rasmus Nyholm Jørgensen, Peter Kryger Jensen, Mads Dyrmann, and Henrik Skov Midtiby. A public image database for benchmark of plant seedling classification algorithms. *arXiv:1711.05458*, 2017.

Bharath Hariharan and Ross Girshick. Lowshot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266): 1332–1338, 2015. ISSN 0036-8075.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, page 436–444, 2015.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, Johns Hopkins University, 2013.

Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1485–1488, 2010.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29): 861, 2018.

Horea Mureşan and Mihai Oltean. Fruit recognition from images using deep learning. *Acta Universitatis Sapientiae, Informatica*, 10(1):26–42, 2018.

Stavros Niafas. Image retrieval platform for building recognition in urban environments. Master's thesis, Informatique, Synthese D'Images et Conception Graphique, 10 2016. URL https://www.kaggle.com/sniafas/vyronas-database.

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

Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4136–4145, 2020.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv:1903.03096*, 2019.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V.

Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc., 2016.

Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv:1911.04623*, 2019.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Bowen Zhang, Xifan Zhang, Fan Cheng, and Deli Zhao. Few shot learning with simplex, 2018.

Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.