EXPLORING REPRESENTATION LEARNING FOR FLEXIBLE FEW-SHOT TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing approaches to few-shot learning deal with tasks that have persistent, rigid notions of classes. Typically, the learner observes data only from a fixed number of classes at training time and is asked to generalize to a new set of classes at test time. Two examples from the same class would always be assigned the same labels in any episode. In this work, we consider a realistic setting where the relationship between examples can change from episode to episode depending on the task context, which is not given to the learner. We define two new benchmark datasets for this flexible few-shot scenario, where the tasks are based on images of faces (Celeb-A) and shoes (Zappos50K). While classification baselines learn representations that work well for standard few-shot learning, they suffer in our flexible tasks since the classification criteria shift from training to testing. On the other hand, unsupervised contrastive representation learning with instance-based invariance objectives preserves such flexibility. A combination of instance and class invariance learning objectives is found to perform best on our new flexible few-shot learning benchmarks, and a novel variant of Prototypical Networks is proposed for selecting useful feature dimensions.

1 INTRODUCTION

Following the success of machine learning applied to fully-supervised settings, there has been a surge of interest in machine learning within more realistic, natural learning scenarios. Among these, meta-learning and few-shot learning (Lake et al., 2011) (FSL) have emerged as exciting alternatives. In the few-shot learning setting, the learner is presented with episodes of new learning tasks, where the learner must identify patterns in a labeled support set and apply them to make predictions for an unlabeled query set. Since its inception, there has been significant progress on FSL benchmarks. However, standard supervised baselines are often shown to perform as well as carefully designed solutions (Chen et al., 2019; Tian et al., 2020). In this work, we argue that this observation is due in part to the rigidity in which FSL episodes are designed.

In a typical few-shot classification setting, each episode consists of a few examples belonging to one of N classes. Across different training episodes, different images are sampled from the classes in the training set but they will always be given the same class label: an elephant is always an elephant. Most current approaches to FSL attempt to remove context. Existing tasks focus on classification judgements, where the query image should be deemed similar to the support image belonging to the same class, factoring out the role of context such as the setting, pose, and presence of other objects. But many judgements are contextual—they depend on the task at hand and frame-of-reference. A rock is similar to a chair when the aim is to sit, but similar to a club if the aim is to hit. Meta-learning is especially appropriate in contextual judgements, as people are able to adapt readily to new contexts and make appropriate judgements. So an important question is how to get context into few-shot classification?

In this work, we define a new flexible few-shot learning (FFSL) paradigm. Instead of building episodes from classes, each episode is a binary classification problem that is constructed with some context that is hidden from the learner. In this way, the same data point may be given different labels across multiple episodes. For example, elephants and tables may belong to the same class if the context is "has legs", but not when the context is "has ears". Importantly, the learner is not given direct access to the context and must infer it from the examples present in the episode.

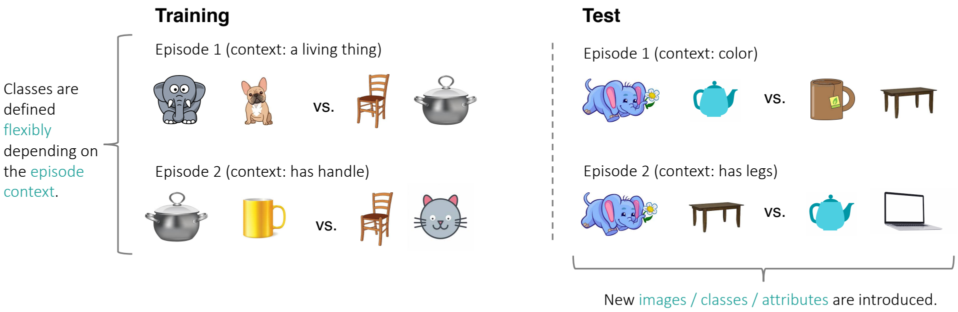

Training
Test
Classes are defined flexibly depending on the episode context.
Episode 1 (context: a living thing)
vs.
Episode 2 (context: has handle)
vs.
Episode 1 (context: color)
vs.
Episode 2 (context: has legs)
vs.
New images / classes / attributes are introduced.

Figure 1: **Illustration of the flexible few-shot learning tasks.** Instead of having a fixed semantic class, each example may belong to different classes flexibly depending on the context of each episode. New classes and attributes are introduced in testing to establish new classification criteria.

Our FFSL problem is significantly more challenging than the standard setup. In each episode, a learner must infer the correct context and adapt their predictions accordingly. In Section 5.1 we study generalization issues that occur under supervised representation learning for the flexible few-shot tasks. We show that these approaches easily overfit to the training attributes, even when given direct access to the attributes that determine the context. We provide additional analysis of a toy problem to illustrate one possible cause of this failure.

In this work, we contribute two new benchmark datasets for this flexible few-shot scenario. The tasks are based on images of faces (Celeb-A) (Liu et al., 2015) and shoes (Zappos50K) (Yu & Grauman, 2014). We provide a thorough empirical evaluation of existing methods on these tasks. We find that successful approaches in the standard FSL setting fall short on the flexible few-shot tasks. Further, while supervised classification baselines can learn good representation in the standard FSL setting, they suffer in FFSL. On the other hand, we found a combination of instance and class invariance objectives is able to provide improved performance on the flexible few-shot tasks. Moreover, we present Mask-ProtoNet which combines prototype classification with feature selection capability, and it performs better compared to standard prototype averaging and linear readout.

2 Background: Standard Few-shot Classification

The vast majority of standard few-shot classification datasets are constructed as follows. First, a standard supervised classification dataset is obtained (e.g. MNIST). Some number of the classes are designated as training classes (e.g. digits 0-4), and the dataset is partitioned so that all images belonging to the training classes are placed into the training set. The remaining classes are used for validation/testing.

At training time, the learner is given episodes ($\mathcal{E}$) to learn from. The episode is divided into a labelled *support set* ($\mathcal{E}_S$) and an unlabelled *query set* ($\mathcal{E}_Q$). An episode is said to be N-way when it contains data points from only N classes. Additionally, the episode is k-shot when there are k labelled data points from each of the N classes in the support set. Given an episode, the learner must successfully predict the class identity of data points in the query set, given the small amount of labelled information in the support set. Throughout, we use $\mathbf{x}$ to denote input data and y the corresponding class labels for this input.

Prototypical networks: A standard prototypical network (Snell et al., 2017) consists of an embedding network, g, and a choice of distance function. In each episode, the labelled support data are used to construct class prototypes, $\mathbf{c}$, by averaging the data points assigned to each class. The likelihood of the query predictions is then given by $p(y = i|\mathbf{x}) = \mathrm{softmax}(-d(g(\mathbf{x}), \mathbf{c}_i))$. Typically d is the squared Euclidean distance or the cosine dissimilarity function.

3 Related Work

Meta-learning and few-shot learning: As one of the earlier studies of FSL, Lake et al. (2011) showed that probabilistic programming can learn about unseen hand-written characters in the Omniglot dataset using few examples. Koch et al. (2015) showed that a deep Siamese network can achieve similar performance. Vinyals et al. (2016) introduced the more challenging *mini*ImageNet dataset. This lead to the development of many meta-learning methods with deep networks including

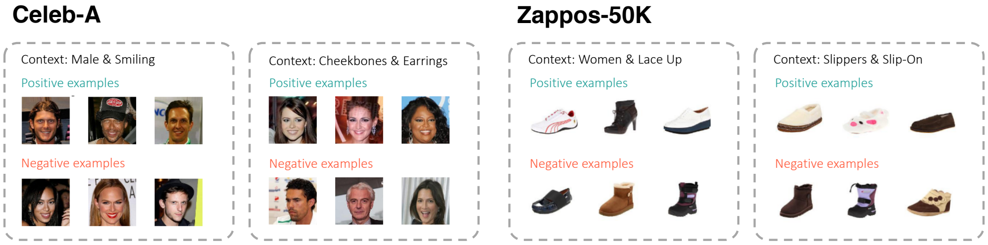


Figure 2: Sample FFSL episodes using Celeb-A (left) and Zappos-50K (right) datasets. Positive and negative examples are sampled according to the context attributes, but the context information is not revealed to the model at test time.

MAML (Finn et al., 2017), Matching Network (Vinyals et al., 2016), and the Prototypical Network (Snell et al., 2017). One hypothesis is that to solve the FSL task, a model needs to be flexible enough to adapt its feature extractor to the unseen test task. Though MAML is very flexible, it is not empirically better than simpler methods such as Prototypical Networks. To strike a balance between flexibility and simplicity, TADAM (Oreshkin et al., 2018) proposed adapting the network using the FiLM layer (Perez et al., 2018), a generalization of conditional normalization.

In our work, we explore some generalization challenges introduced by the FFSL benchmarks. In general, there is limited theoretical support for the success of meta-learning. Most existing work focuses on defining notions of task similarity (Ben-David et al., 2010; Ben-David & Borbely, 2008), building explicit models for meta-learning (Baxter, 2000; Pentina & Lampert, 2014) or on learning good data representations for generalization across tasks (Maurer, 2009; Bullins et al., 2019; Du et al., 2020). Yet another line of work investigates the theoretical limitations of few-shot learning (Hanneke & Kpotufe, 2020; Lucas et al., 2020). Here we study the generalization failure modes of supervised representation learning approaches to the FFSL tasks.

The standard few-shot classification task has been extended in various ways. In few-shot semi-supervised learning, the support set is augmented with unlabelled examples to provide access to extra information (Ren et al., 2018). This has inspired novel algorithms such as a meta-learning version of learning from pseudo-labels (Sun et al., 2019). To capture the possibility that a model needs to deal with varying support set size, and task difficulty, Triantafillou et al. (2019) introduced the Meta-Dataset. They found that a hybrid of Prototypical Networks and MAML performed best. To capture another aspect of learning in the real world, Finn et al. (2018) investigated the possibility of having ambiguous tasks. In the same spirit, we extend the study of few-shot learning by introducing our FFSL benchmarks, and show that this task calls for novel algorithms.

Zero-shot learning: In zero-shot learning (ZSL), a model is asked to recognize classes not present in the training set, supervised only by some auxiliary description or attribute values (see Wang et al. (2019a) for a survey). Lampert et al. (2014) studied the *direct attribute prediction* method. In subsequent sections we also look at pretraining a predictor of attribute values. One motivating factor for ZSL is the situation where no training example is available for the new classes, but only descriptions of them. The motivation behind our FFSL task can be seen as complementary in that sometimes a new concept cannot easily be described, but coming up with a small set of representative examples is easier, e.g. “shoes that I like”. This suggests a comparison to recommendation systems.

Cold start in recommendation systems: Our FFSL tasks share overlap with the cold start problem in recommendation systems (Lam et al., 2008; Gope & Jain, 2017), in which a new user or item is added to the system with little or no information. As data is collected on the new instance, the system must quickly learn to generate good recommendations. The similarity of meta-learning and cold-start recommendation has been explored before (Vartak et al., 2017). However, as new users can be considered as having their own context to classify items, arguably our flexible few-shot tasks share greater similarity with cold-start recommendation than standard FSL settings.

Compositional learning: Compositional features can be used to construct novel concepts. This has been used to improve ZSL where a model not only predicts the class, but also attribute values of unseen objects (Purushwalkam et al., 2019; Wang et al., 2019b; Yang et al., 2020). Another aspect of our FFSL task is the need to reason about the underlying decision criteria. This theme is also important in the Visual IQ test proposed in Barrett et al. (2018). There a model is asked to infer and extrapolate attribute values to solve Raven’s Progressive Matrices.

4 FFSL: Flexible Few-Shot Learning

In this section, we define our FFSL paradigm and introduce our two new benchmark datasets. As in the standard few-shot classification setting (Section 2), our learner is presented with episodes of data. However, the episodes are not constrained to contain data points from only N classes. Instead, each data point is given either a positive or negative label depending on some criteria that is not known to the learner.

Figure 1 shows some examples of different episodes in our FFSL setting. Each episode contains an image of a pot, but the class identity of the pot varies according to the hidden context. In Episode 1, the pot and the chair are given the same labels whereas in Episode 2 they belong to different classes. Moreover, at test time brand new concepts (e.g. tables) or criteria (e.g. color) may be introduced.

Conceptually, each data point $\mathbf{x} \in \mathcal{X}$ represents some combination of hidden attributes $\mathbf{z} \in \mathcal{Z}$. And each context is an injective function, $f : \mathcal{Z} \to \{0, 1\}$, that labels each of the data points depending on their hidden attributes. In this work, we consider contexts that compute conjuctions of binary attributes. The set of training contexts and test contexts need not be the same.

In order to solve the FFSL task, the learner must correctly find a mapping from the data domain $\mathcal{X}$ to the correct labels. One natural way to solve this problem would be to first find a mapping $h : \mathcal{X} \to \mathcal{Z}$, that is persistent across episodes, and then estimate the context in each episode. However, we do not limit our exploration to methods that use this approach, since FFSL allows different partitions of the $\mathcal{Z}$ space for training and testing, and as we will explain in Section 5.1, directly learning to predict $\mathcal{Z}$ can lead to generalization issues.

Next we describe how we generate the FFSL datasets using existing image datasets with attributes, Celeb-A faces (Liu et al., 2015) and Zappos-50K shoes (Yu & Grauman, 2014). Sample episodes from each dataset are shown in Figure 2.

Celeb-A: The Celeb-A dataset contains around 200K images, where we split half to training, and a quarter to validation and testing each. Each image is annotated with 40 binary attributes, detailing hair colour, facial expressions, and other descriptors. We picked 27 salient attributes and split 14 for training and 13 for both val and test. There is no overlap between training or test attributes but they may sometimes belong to a common category, e.g. blonde hair is in training and brown hair is in test. Split details are included in the Appendix B. For each episode, we randomly select one or two attributes and look for positive example belonging to these attributes simultaneously. And we also sample an equal number of negative examples that don't belong to one or both of the selected attributes. This will construct a *support set* of positive and negative samples, and then we repeat the same process for the corresponding *query set* as well.

Zappos-50K: The Zappos-50K dataset contains just under 50K images of shoes annotated with attribute values, out of which we kept a total of 76 that we considered salient. We construct an image-level split that assigns 80% of the images to the training set, 10% to the validation and 10% to the test set. We additionally split the set of attribute values into two disjoint sets that are used to form the training and held-out FFSL tasks, respectively. Sampling an episode from a particular split involves sampling a conjunction of attributes from that split (e.g. 'gender = boy' and 'material = leather'), and then sampling positive and negative examples from the relevant example split. The positive examples obey both clauses of the conjunction and, as a design choice, the negative examples do not obey either clause. The sampled positive and negative examples are then divided into a support and query set for the episode.

5 Exploring Models for Flexible Few-Shot Learning

In this section, we explore different learning models to solve FFSL tasks. Overall, we separate learning into two stages: *representation learning* and *few-shot learning*. In the representation learning stage, a network backbone learns task relevant features over many examples. And in the FSL stage, an episode with a few examples is presented, and the learner utilizes the base backbone network and performs additional learning on top.

For typical *meta-learning* based methods, these two stages are essentially the same—training performs episodic learning just like testing. Aside from meta-learning, simple supervised pretraining can also learn good representation for standard few-shot classification by using a linear classifier readout at test time (Chen et al., 2019; Tian et al., 2020).

5.1 Generalization issues with supervised representation learning

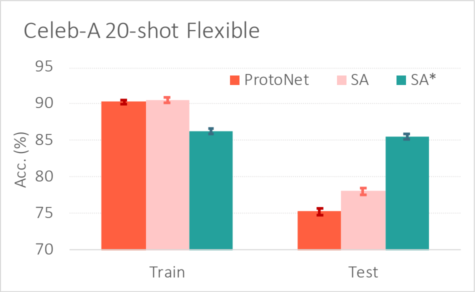


Figure 3: FFSL 20-shot classification. Both supervised attribute classification and standard FSL do not generalize well.

In the FFSL task, any single example can have several positive attributes and the context used to classify them varies across training and test. This suggests that useful representations must be more general than those needed for standard FSL. To investigate this, we first conducted an initial experiment on the Celeb-A benchmark. We adopted a standard prototypical network (**ProtoNet**) with features learned through the episodic query loss as our meta-learning approach. We also explored pretraining-based approaches. We trained a classifier to predict the 14 binary training attributes from the input images to learn a representation. At test time we simply used a linear classifier to solve each episode. This approach is denoted as **SA** (Supervised Attributes), analogous to the setting in Chen et al. (2019). We also trained an oracle classifier (**SA***) on all 40 attributes in the dataset, including both training and testing attributes. Since the tasks are constructed using attribute information, the performance of **SA*** should be considered an upper bound for this problem.

Results are shown in Figure 3. Both ProtoNet and SA perform well on the training tasks since they are exposed to the label information from the training attributes; however, the test performance shows a significant generalization gap. In order to succeed in the training objective, both ProtoNet and SA essentially learn to ignore other features that are potentially useful for testing as classification criteria. By contrast, SA* is able to perform similarly on both training and testing, since the learning does not depend on a particular split of the attributes. Initial experiments therefore suggest that supervised learning alone will likely not be sufficient for our FFSL task.

In Appendix A we study a toy FFSL problem which further illustrates these generalization issues. We explore training a prototypical network on data from a linear generative model, where each episode presents significant ambiguity in resolving the correct context. We show that in this setting, unlike in standarad FSC tasks, the prototypical network is forced to discard information on the test attributes in order to solve the training tasks effectively, and thus fails to generalize.

5.2 Unsupervised constrastive representation learning

Learning good representation for downstream applications has always been a sought-after purpose of deep learning. Hinton & Salakhutdinov (2006) proposed to pretrain subsequent layers of autoencoders for representation learning, and showed good performance for dimensionality reduction, and downstream classification. Following the development of variational autoencoders (VAEs) (Kingma & Welling, 2013), many extensions have been proposed to encourage “disentangled” representation learning by reweighing terms in the evidence lower bound (Higgins et al., 2017; Kim & Mnih, 2018).

In contrast to traditional generative modeling where the objective is grounded on uncovering the data distribution, self-supervised learning recently emerged as a promising approach for representation learning. These include learning to predict rotations (Kolesnikov et al., 2019), maximize mutual information between the input and representation (Belghazi et al., 2018; van den Oord et al., 2018), and contrastive learning approaches (Chen et al., 2020; van den Oord et al., 2018; Tian et al., 2019; He et al., 2019; Xiong et al., 2020). They have shown promise in learning semantic aware representations, almost closing the gap with supervised representation training on the challenging ImageNet benchmark. We follow SIMCLR (Chen et al., 2020) as a representative framework for unsupervised contrastive learning, shown in Figure 4-A. We chose SIMCLR because of its empirical success.

Concretely, it sends a pair of augmented versions of the same image to the input and obtains a hidden representation. The hidden representation is further passed into a decoder, producing unit-norm vectors. The network is trained end-to-end to minimize the InfoNCE loss (van den Oord et al.,

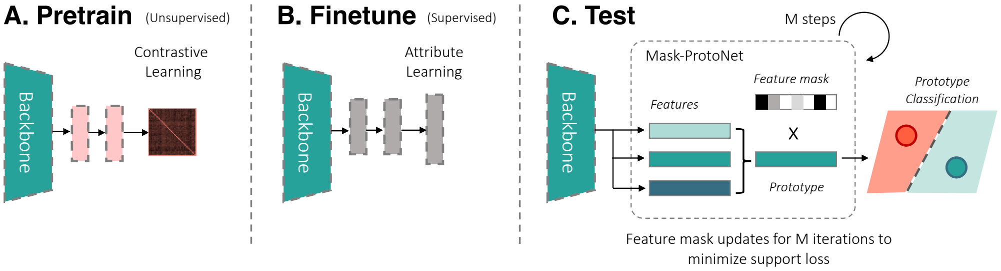


Figure 4: **Our proposed method for FFSL. A:** we first pretrain the network with unsupervised contrastive objective to learn general features. **B:** Then we finetune the network to classify the set of training attributes. Both stages employ a different decoder header so that the representation remains general. **C:** Finally at test time we use Mask-ProtoNet, a variant of ProtoNet that infers feature selection iteratively.

2018), which distinguishes the positive sample from the same pair from the rest by encouraging feature dot product between the positive pair to gain a higher value than negative pairs.

Finetuning with supervised attribute classification We can combine the merits of unsupervised representation learning and supervised attribute classification (SA). To prevent SA from overriding the unsupervised features, we add another classifier decoder MLP before the sigmoid classification layer (see Figure 4-B). Empirically, finetuning on SA is found to be beneficial, but early stopping is needed to prevent optimizing too much towards training attributes, which would cause significant generalization issues (Section 5.1).

During test time, we directly use the representation before both decoders to perform FSL. In the next section, we introduce Mask-ProtoNet, a novel method for FFSL.

5.3 Few-Shot Learning with Mask-ProtoNet

Once the representation is learned, a common approach for FSL is to directly learn a linear classifier on top of the representation, or average the prototypes from the support set. Prototype averaging, however, will consider all feature dimensions, including the ones that are not relevant to the current episode. A linear classifier, on the other hand, learns a weight coefficient for each feature dimension, thus performing some level of feature selection. Still, the weights need to be properly regularized to encourage high-fidelity selection. A popular way is to apply an L1 regularizer on the weights to encourage sparsity. The learning of a classifier is essentially done at the same time as the selection of feature dimensions. In this paper, we propose Mask-ProtoNet as an alternative for few-shot learning that separates the procedure of classifier learning and feature selection: we use prototypes for classification and additionally learn a soft binary mask for feature selection.

Algorithm 1 Mask-ProtoNet

```
Require: Net, ${\mathbf{x}_i^S, y_i^S}_{i=1}^N$, ${\mathbf{x}^Q}_{j=1}^M$
    // An embedding network, $N$ support, $M$ query
Ensure: ${\hat{y}_j^Q}_{j=1}^M$
    // Network representation $\mathbf{h} \in \mathbb{R}^D$
$\mathbf{h}_i^S \leftarrow \text{Net}(\mathbf{x}_i^S) \ \ \forall i; \ \ \mathbf{h}_j^Q \leftarrow \text{Net}(\mathbf{x}_j^Q) \ \ \forall j;$
$\mathbf{w} \leftarrow 0 \in \mathbb{R}^D;$
for $t = 1 ... M + 1$ do
    $\tilde{\mathbf{w}} \leftarrow \sigma(\mathbf{w})$
    $\mathbf{p}[k] \leftarrow \frac{\sum_i (\mathbf{h}_i^S \odot \tilde{\mathbf{w}}) \mathbb{1}[y_i^S = k]}{\sum_i \mathbb{1}[y_i^S = k]}$
    $\hat{y}_{i,k}^S \leftarrow \text{softmax}(-d(\mathbf{h}_i^S \odot \tilde{\mathbf{w}}, \mathbf{p}[k])) \ \forall i;$
    $l \leftarrow -\frac{1}{N} \sum_i CE(\hat{y}_i^S, y_i^S) + \lambda \|\tilde{\mathbf{w}}\|_1$
    $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} l$
end for
$\hat{y}_{j,k}^Q \leftarrow \text{softmax}(-d(\mathbf{h}_j^Q \odot \tilde{\mathbf{w}}, \mathbf{p}[k])) \ \forall j;$
return $\hat{y}_j^Q$
```

Just like a linear classifier, the Mask-ProtoNet learns a weight coefficient for each dimension. This weight is then passed through a sigmoid function to act as a soft binary mask, which is learned for a small number of iterations before termination. Finally classification is performed based on the masked prototypes. Conceptually, the mask will disable unused features and instead focus on dimensions that are activated in the current episode. The mask is updated to minimize the inner loop loss, which is a combination of support set cross entropy and an L1 sparse regularizer. The full algorithm is described in Algorithm 1 and Figure 4-C.

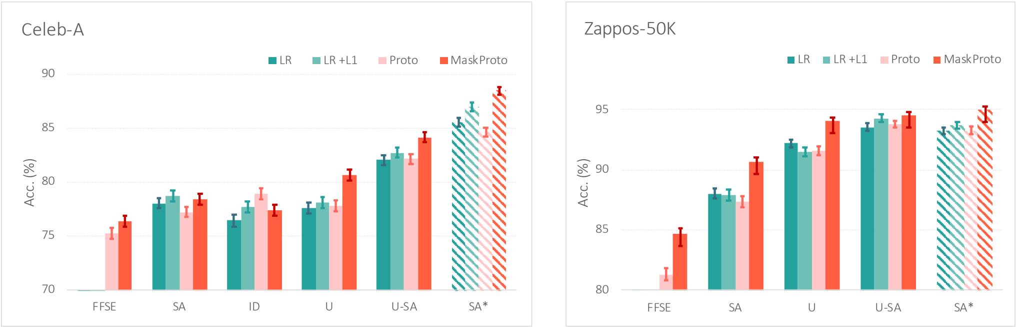


Figure 5: **20-shot FFSL results comparing different representation learning and FSL stage combinations.** **FFSE**: Meta-learning directly using the flexible few-shot episodes. **SA**: Supervised attribute classification. **ID**: Auxiliary representation learning (for Celeb-A this is face ID classification). **U**: Unsupervised contrastive learning. **U-SA**: Our proposed U pretraining followed by SA finetuning. **SA***: Supervised attribute binary classification on **all** attributes, which serves as an oracle (striped bars). A set of few-shot learners are evaluated: 1) logistic regression (**LR**), 2) LR with L1 regularization (**LR +L1**), 3) ProtoNet (**Proto**), and 4) the proposed Mask-ProtoNet (**MaskProto**). U-SA with Mask-ProtoNet achieves the best performance in both benchmarks. Chance is 50%.

6 Experiments

In this section we present our experimental evaluations with various representation learning and few-shot learning methods for our FFSL benchmarks. Representation learning methods include: 1) **FFSE**: Meta-learning through **F**lexible **F**ew-**S**hot **E**pisodes; 2) **SA**: **S**upervised **A**ttribute classification on training attributes only; 3) **ID**: Auxiliary representation learning task, for Celeb-A this is the face **ID**entity classification; 4) **U**: **U**nsupervised representation learning (SIMCLR); 5) **U-SA**: **U**nsupervised representation learning followed by **S**upervised **A**ttribute classification finetuning. This approach is described in Figure 4-A and B; 6) **SA***: **S**upervised **A**ttribute classification on all attributes, which serves as an oracle.

We also compared the following methods for few-shot learning: 1) **LR**: Plain logistic regression on the hidden representation; 2) **LR +L1**: LR with L1 regularization on the weights; 3) **Proto**: Classification with prototypes (Snell et al., 2017); 4) **MaskProto**: Prototypes with additional mask that is learned in an inner loop (as proposed in this paper, described in Algorithm 1).

Implementation details: Images were resized to 84 × 84 × 3. We used ResNet-12 (He et al., 2016; Oreshkin et al., 2018) with 64, 128, 256, 512 channels in each residual module. The decoder network for contrastive learning has two 512-d layers and outputs 128-d vectors. The classifier finetuning decoder network has two 512-d layers and outputs a 512-d vector. We trained SIMCLR using random crop areas of 0.08 – 1.0, color augmentation 0.5, and InfoNCE temperature 0.5, for 1000 epochs using LARS (You et al., 2017) and cosine schedule with batch size 512 and peak learning rate 2.0. SA finetuning lasts for another 2k steps with batch size 128 and learning rate 0.1 for the decoder and 0.01 for the backbone and momentum 0.9. ID, SA and SA* use batch size 256 with a learning rate 0.1 for 30k steps, with 0.1x learning rate decay at 20k and 25k steps, and momentum 0.9. Features are normalized before sending to LR classifiers. We use cosine similarity for ProtoNet and Mask-ProtoNet.

6.1 Results and discussion

Main results: Figure 5 shows our main results on Celeb-A and Zappos-50K with 20-shot FFSL episodes. On both benchmarks, training on flexible few-shot episodes based on training attributes (FFSE) performed worst. This aligns with our observation of the generalization issue explained in Section 5.1. Similarly, supervised attribute (SA) learning faced the same challenge. An auxiliary task of class identification (ID) was not helpful for representation learning either. Interestingly, unsupervised representation learning (U) attained relatively better test performance, suggesting that the training objective in contrastive learning preserves more general features—not just shown for semantic classification tasks in prior literature, but also for the flexible class definitions present here. Surprisingly, finetuning slightly on SIMCLR pretrained networks (U-SA) contributed further gains in performance. We also tried to finetune directly on FFSL episodes using meta-learning approaches but this did not perform well — one possible explanation is given in our toy example (Appendix A). We conclude that meta-learning may not help learn higher-level features about the FFSL task itself. Lastly, we confirmed that U-SA closes the generalization gap between SA and SA*, and obtained

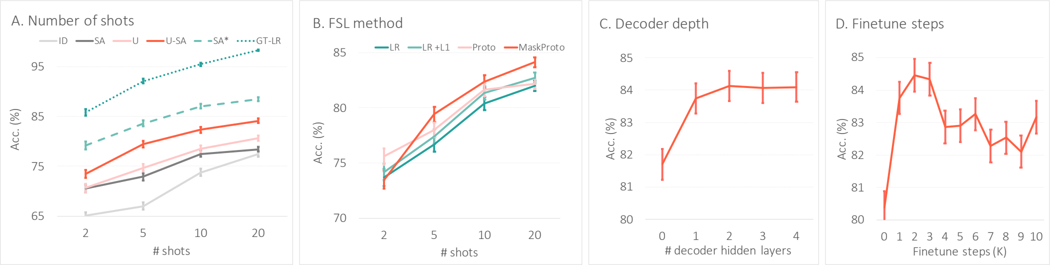


Figure 6: Additional results on the Celeb-A dataset. **A: How many examples are needed for FFSL?** We provide an oracle performance where the feature representation is directly the binary ground-truth attribute vector (**GT-LR**) and we train a logistic regression classifier on top. It suggests that there is natural ambiguity in the task and more examples than standard FSL are needed. **B: Comparison of few-shot learning methods on different number of shots.** Mask-ProtoNet works better with an increasing number of shots. **C: Effect of the number of decoder layers during finetuning.** Adding a decoder keeps the representation general and not overfitting to the training attributes. **D: Effect of the number of finetuning steps.** Small amount of finetuning on the training attribute is beneficial, but eventually the accuracy goes down.

matching performance on Zappos-50K. Lastly, we confirmed that U-SA closes the generalization gap between SA and SA*. These results were consistent across our benchmarks. Therefore, U-SA was the most effective representation learning algorithm we explored for FFSL. Note that this result contrasts with standard FSL literature, where unsupervised representation learning still lags behind supervised pretraining (Medina et al., 2020). Moreover, MaskProto is often the best across different FSL approaches, consistently higher than Proto, which does not reason about feature selection.

Number of shots: Since we have a flexible definition of classes in each episode, it could be the case that the support examples are ambiguous. For example, by presenting both an elephant and a cat in the support set, it is unclear whether the positive set is about animals or mammals. Figure 6-A shows several approaches evaluated using Mask-ProtoNet with varying number of support examples per class in Celeb-A FFSL episodes. In addition to the SA* oracle, we provided another oracle GT-LR, where the representations are the binary attribute values, and readout is done by solving a linear classifier. GT-LR gradually approached 100% accuracy as the number of shots approached 20. This demonstrates that FFSL tasks potentially require more support examples to resolve ambiguity. Again here, U-SA consistently outperformed U, SA, and ID baselines across different number of shots.

Figure 6-B plots the performance of different FSL methods, using a common U-SA representation. Mask-ProtoNet performs better with more support examples, but worse with fewer (e.g. 2), since minimizing the support loss of only two examples can lead to over-confidence.

Effect of decoder depth: Figure 6-C studies the effect of a decoder for attribute classification finetuning. Adding an MLP decoder was found to be beneficial for unsupervised representation learning in prior literature (Chen et al., 2020). Here we found that adding a decoder is also important for SA finetuning, contributing to over 2% improvement.

Effect of SA finetuning: Figure 6-D plots the validation accuracy on FFSL tasks during finetuning for a total of 10k steps. It is found that the accuracy grows from 80% and peaks at 2k steps with over 84%, and then drops. This suggests that a little finetuning on supervised attributes is beneficial, but prolonged finetuning eventually makes the representation less generalizable.

7 Conclusion

The notion of a class often changes depending on the context, yet existing few-shot classification relies on a fixed semantic class definition. In this paper, we propose a flexible few-shot learning paradigm where the classification criteria change based on the episode context. We proposed benchmarks using the Celeb-A and Zappos-50K datasets to create flexible definitions with existing attribute labels. We explored various ways to perform representation learning for this new task. Unlike in standard FSL, we found that supervised representation learning generalizes poorly on the test set, due to the partitioning of training & test attributes. Unsupervised contrastive learning on the other hand preserved more generalizable features, and further finetuning on supervised attribute classification yielded the best results. Finally, a variant of ProtoNet, Mask-ProtoNet is proposed and delivers better readout performance. The development of FFSL benchmarks will hopefully encourage more future research investigating the generalization ability of meta-learning methods.

REFERENCES

David GT Barrett, Felix Hill, Adam Santoro, Ari S Morcos, and Timothy Lillicrap. Measuring abstract reasoning in neural networks. *arXiv preprint arXiv:1807.04225*, 2018.

Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12: 149–198, 2000.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

Shai Ben-David and Reba Schuller Borbely. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine learning*, 73(3):273–287, 2008.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

Brian Bullins, Elad Hazan, Adam Kalai, and Roi Livni. Generalize across tasks: Efficient algorithms for linear representation learning. In Aurélien Garivier and Satyen Kale (eds.), *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, pp. 235–246. PMLR, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *Proceedings of the 7th International Conference on Learning Representations, ICLR*, 2019.

Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, 2017.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

Jyotirmoy Gope and Sanjay Kumar Jain. A survey on solving cold start problem in recommender systems. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 133–138. IEEE, 2017.

Steve Hanneke and Samory Kpotufe. A no-free-lunch theorem for multitask learning. *arXiv preprint arXiv:2006.15785*, 2020.

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, CVPR*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722, 2019.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR*, 2017.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1920–1929, 2019.

Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the 33th Annual Meeting of the Cognitive Science Society, CogSci*, 2011.

Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pp. 208–211, 2008.

Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(3):453–465, 2014.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision, ICCV*, 2015.

James Lucas, Mengye Ren, Irene Kameni, Toniann Pitassi, and Richard S. Zemel. Theoretical bounds on estimation error for meta-learning. *arXiv*, 2020.

Andreas Maurer. Transfer bounds for linear feature learning. *Machine learning*, 75(3):327–350, 2009.

Carlos Medina, Arnout Devos, and Matthias Grossglauser. Self-supervised prototypical transfer learning for few-shot classification. *CoRR*, abs/2006.11325, 2020.

Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: task dependent adaptive metric for improved few-shot learning. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31, NeurIPS*, 2018.

Anastasia Pentina and Christoph H. Lampert. A pac-bayesian bound for lifelong learning. In *Proceedings of the 31th International Conference on Machine Learning, ICML*, 2014.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI*, 2018.

Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-driven modular networks for zero-shot compositional learning. In *IEEE/CVF International Conference on Computer Vision, ICCV*, 2019.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *Proceedings of 6th International Conference on Learning Representations, ICLR*, 2018.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30, NIPS*, 2017.

Qianru Sun, Xinzhe Li, Yaoyao Liu, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. *arXiv preprint arXiv:1906.00562*, 2019.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *CoRR*, abs/1906.05849, 2019.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *CoRR*, abs/2003.11539, 2020.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *8th International Conference on Learning Representations, ICLR*, 2019.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. A meta-learning perspective on cold-start recommendations for items. In *Advances in neural information processing systems*, pp. 6904–6914, 2017.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29, NIPS*, 2016.

Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10 (2):1–37, 2019a.

Xin Wang, Fisher Yu, Trevor Darrell, and Joseph E. Gonzalez. Task-aware feature generation for zero-shot compositional learning. *CoRR*, abs/1906.04854, 2019b.

Yuwen Xiong, Mengye Ren, and Raquel Urtasun. Loco: Local contrastive representation learning. *CoRR*, abs/2008.01342, 2020.

Muli Yang, Cheng Deng, Junchi Yan, Xianglong Liu, and Dacheng Tao. Learning unseen concepts via hierarchical decomposition and composition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2020.

Yang You, Igor Gitman, and Boris Ginsburg. Scaling SGD batch size to 32k for imagenet training. *CoRR*, abs/1708.03888, 2017.

Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2014.

A Flexible Few-Shot Toy Problem

In this section we give details on a toy problem that illustrates the challenges introduced by the flexible few-shot learning setting and the failures of existing approaches on this task. This simple model captures the core elements of our flexible few-shot tasks, including ambiguity, domain shift from training to test time, and the role of learning good representations. The primary limitation of this model is the fact that it is fully linear — in a more realistic FFSL task recovering a good representation from the data is significantly more challenging, and the data points will have a more complex relationship with the attributes as in our benchmark datasets.

Problem setup We define a flexible few-shot learning problem where the data points $\mathbf{x} \in \mathbb{R}^m$ are generated from binary attribute strings, $\mathbf{z} \in \{0, 1\}^d$, with $\mathbf{x} = A\mathbf{z} + \boldsymbol{\zeta}$ for some matrix $A \in \mathbb{R}^{m \times d}$ with full column rank and noise source $\boldsymbol{\zeta}$. Thus, each data point $\mathbf{x}$ is a sum of columns of A with some additive noise.

We consider contexts that classify the examples as positive when two attributes are 1-valued, and negative otherwise (an AND attribute context). For the training episodes, the contexts depend only on the first $d_1 < d$ attributes. At test time, the episode contexts depend on the remaining $d - d_1$ attributes. The episodes are generated by sampling a context uniformly, and then uniformly sampling k data points with positive labels and k with negative labels.

Linear prototypical network Now, consider training a prototypical network on this data with a linear embedding network, $g(\mathbf{x}) = W\mathbf{x}$. Within each episode, the prototypical network computes the prototypes for the positive and negative classes,

$$\mathbf{c}_j = \frac{1}{k} \sum_{\mathbf{x}_i \in S_j} g(\mathbf{x}_i), = \frac{1}{k} \sum_{\mathbf{x}_i \in S_j} \sum_{l=1}^{d} z_{il} W \mathbf{a}_l, \text{ for } j \in \{0, 1\},$$

where S_j is the set of data points in the episode with label j, and $\mathbf{a}_l$ is the l^{th} column of the matrix A. Further, the prototypical network likelihood is given by,

$$p(y = 0|\mathbf{x}) = \frac{\exp\left\{-\|W\mathbf{x} - \mathbf{c}_0\|_2^2\right\}}{\exp\left\{-\|W\mathbf{x} - \mathbf{c}_0\|_2^2\right\} + \exp\left\{-\|W\mathbf{x} - \mathbf{c}_1\|_2^2\right\}}.$$

The goal of the prototypical network is thus to learn weights W that lead to small distances between data points in the same class and large distances otherwise. In the flexible few-shot learning tasks, there is an additional challenge in that class boundaries shift between episodes. The context defining the boundary is unknown and must be inferred from the episode. However, with few shots (small k) there is ambiguity in the correct context — with a high probability that several possible contexts provide valid explanations for the labels.

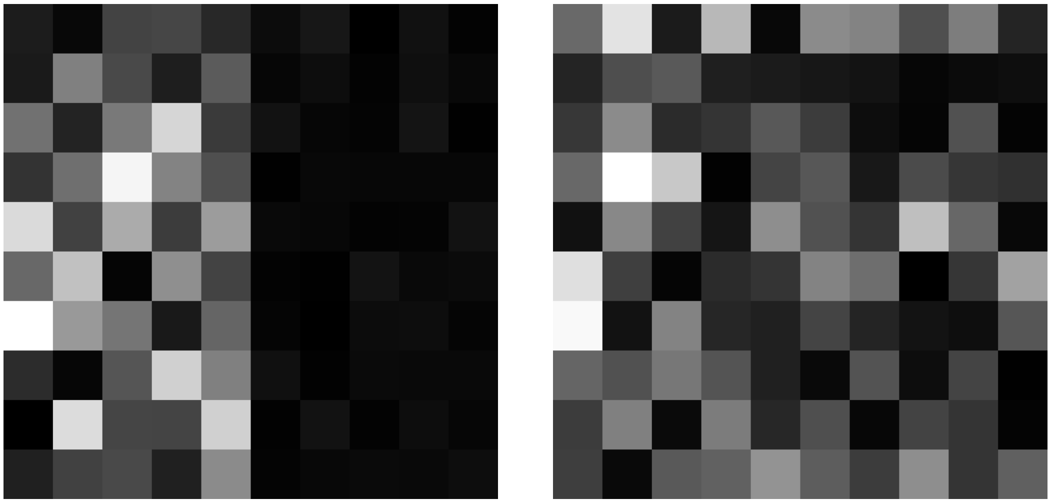
Fitting the prototypical network Notice that under our generative model, with $\mathbf{x} = W\mathbf{z} + \boldsymbol{\zeta}$ we have,

$$W\mathbf{x} - \mathbf{c}_j = WA(\mathbf{z} - \frac{1}{k} \sum_{\mathbf{z}_i \in S_j} \mathbf{z}_i) + \frac{1}{k} \sum_i W\boldsymbol{\zeta}_i + W\boldsymbol{\zeta}, \text{ for } j \in \{0, 1\}.$$

Notice that if $A(\mathbf{z} - \frac{1}{k} \sum_{\mathbf{z}_i \in S_j} \mathbf{z}_i) \in \mathrm{Ker}(W)$ then the entire first term is zero. If $\mathbf{z} \in S_j$ then there is no contribution from the positive attribute features in this term. Otherwise, this term is guaranteed to have some contribution from the positive attribute features.

Therefore, if W projects to the linear space spanned by the positive attribute features then the model will be able to solve the episode without ambiguity. This suggests that the optimal weights are those that project to the set of features used in the training set — destroying all information about the test attributes which would otherwise introduce ambiguity.

We observed this effect empirically in Figure 7, where we have plotted the matrix $\mathrm{abs}(WA)$. Each column of these plots represents a column of A mapped to the prototypical network's embedding space. The first 5 columns correspond to attributes used at training time, and the remaining 5 to those used at test time.



(A) Flexible task learned features

(B) Standard task learned features

Figure 7: Projecting data features into prototypical network embedding space (WA) for the linear toy problem. On the flexible task, the model destroys information from the test attributes to remove ambiguity at training time.

Table 1: **Attribute split for Celeb-A**

Train	5_o_Clock_Shadow	Black_Hair	Blond_Hair	Chubby
	Double_Chin	Eyeglasses	Goatee	Gray_Hair
	Male	No_Beard	Pale_Skin	Receding_Hairline
	Rosy_Cheeks	Smiling		
Val/Test	Bald	Bangs	Brown_Hair	Heavy_Makeup
	High_Cheekbones	Mouth_Slightly_Open	Mustache	Narrow_Eyes
	Sideburns	Wearing_Earrings	Wearing_Hat	Wearing_Lipstick
	Wearing_Necktie			

In the flexible task described above, the learned prototypical feature weights project out the features used at test time (the last 5 columns). As a result, the model achieves 100% training accuracy but only 51% test accuracy (chance is 50%). We also compared against a similar problem set up that resembles the standard few-shot learning setting. In this setting, the binary attribute strings may have only a single non-zero entry and each episode is a binary classification problem where the learner must distinguish between two classes. In this standard few-shot setting, the model is not forced to throw away test-time information and achieves 100% training accuracy and 99% test accuracy.

Settings for Figure 7 We use 10 attributes, 5 of which are used for training and 5 for testing. We use a uniformly random sampled $A \in \mathbb{R}^{30\times10}$ and the prototypical network learns $W \in \mathbb{R}^{10\times30}$. We use additive Gaussian noise when sampling data points with a standard deviation of 0.1. The models are trained with the Adam optimizer using default settings over a total of 30000 random episodes, and evaluated on an additional 1000 test episodes. We used $k = 20$ to produce these plots, but found that the result was consistent over different shot counts.

B Dataset split

We include the attribute split for Celeb-A in Table 1.