# Generating Representative Samples for Few-Shot Classification

Jingyi Xu
Stony Brook University
jingyixu@cs.stonybrook.edu

Hieu Le*
Amazon Robotics
ahieu@amazon.com

## Abstract

*Few-shot learning (FSL) aims to learn new categories with a few visual samples per class. Few-shot class representations are often biased due to data scarcity. To mitigate this issue, we propose to generate visual samples based on semantic embeddings using a conditional variational autoencoder (CVAE) model. We train this CVAE model on base classes and use it to generate features for novel classes. More importantly, we guide this VAE to strictly generate representative samples by removing non-representative samples from the base training set when training the CVAE model. We show that this training scheme enhances the representativeness of the generated samples and therefore, improves the few-shot classification results. Experimental results show that our method improves three FSL baseline methods by substantial margins, achieving state-of-the-art few-shot classification performance on miniImageNet and tieredImageNet datasets for both 1-shot and 5-shot settings. Code is available at: https://github.com/cvlab-stonybrook/fsl-rsvae.*

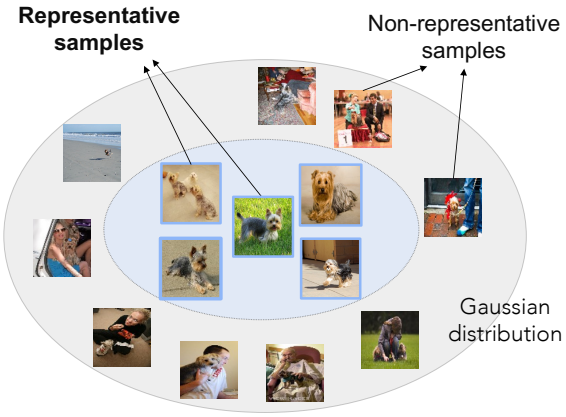



Figure 1. **Representative Samples.** We refer representative samples to the "easy-to-recognize" samples that faithfully reflect the key characteristics of the category. We identify those samples and then use them to train a VAE model for feature generation, conditioned on class-representative semantic embeddings. We show that the generated data significantly improves few-shot classification performance.

## 1. Introduction

Few-shot learning (FSL) methods aim to learn useful representations with limited training data. They are extremely useful for situations where machine learning solutions are required but large labelled datasets are not trivial to obtain (e.g. rare medical conditions [49, 71], rare animal species [75], failure cases in autonomous systems [42,43,58]). Generally, FSL methods learn knowledge from a fixed set of base classes with a surplus of labelled data and then adapt the learned model to a set of novel classes for which only a few training examples are available [73].

Many FSL methods [10, 23, 39, 65, 65, 77, 82] employ a prototype-based classifier for its simplicity and good performance. They aim to find a prototype for each novel class such that it is close to the testing samples of the same class and far away from testing samples for other classes. However, it is challenging to estimate a representative prototype just from a few available support samples [37,79]. An effective strategy to enhance the representativeness of the prototype is to employ textual semantic embeddings learned via NLP models [13, 46, 52, 53] using large unsupervised text corpora [77, 82]. These semantic embeddings implicitly associate a class name, such as "Yorkshire Terriers", with the class representative semantic attributes such as "smallest dog" or "long coat" [1] ( Fig. 1), providing strong and unbiased priors for category recognition.

For the most part, current FSL methods focus on learning to adaptively leverage the semantic information to complete the original biased prototype estimated from the few available samples. For example, the recent FSL method of Zhang *et al*. [82] learns to fuse the primitive knowledge and attribute features into a representative prototype, depending on the set of given few-shot samples. Similarly, Xing *et al*. [77] propose a method that computes an adaptive mixture coefficient to combine features from the visual and tex-

*Work done outside of Amazon

tual modalities. However, learning to recover an arbitrarily biased prototype is challenging due to the drastic variety of the possible combinations of few-shot samples.

In this paper, we propose a novel FSL method to obtain class-representative prototypes. Inspired by zero-shot learning (ZSL) methods [4, 18, 85], we propose to generate visual features via a variational autoencoder (VAE) model [66] conditioned on the semantic embedding of each class. This VAE model learns to associate a distribution of features to a conditioned semantic code. We assume that such association generalizes across the base and novel classes [3, 47]. Therefore, the model trained with sufficient data from the base classes can generate novel-class features that align with the real unseen features. We then use the generated features together with the few-shot samples to construct class prototypes. We show that this strategy achieves state-of-the-art results on both *mini*ImageNet and *tiered*ImageNet datasets. It works exceptionally well for 1-shot scenarios where our method outperforms state-of-the-art methods [76, 80] by $5 \sim 6\%$ in terms of classification accuracy.

Moreover, to enhance the representativeness of the prototype, we guide the VAE to generate more *representative* samples. Here we refer representative samples to the "*easy-to-recognize*" samples that faithfully reflect the key characteristics of the category (see Fig. 1). The embeddings of these representative samples often lie close to their corresponding class centers, which are particularly useful for constructing class-representative prototypes.

Specifically, we guide the VAE model to generate representative samples by selecting only representative data from the base classes for training it. In essence, our VAE model is trained to model the data distribution of the training set. As the training set contains only representative data, the trained VAE model outputs samples that are also representative. Specifically, to select those representative features, we first assume that the feature vectors of each class follow a multivariate Gaussian distribution and estimate this distribution for each base class. Based on these distributions, we compute the probability of each sample belonging to its corresponding category to measure the representativeness for the sample. We filter out the non-representative samples and train the VAE using only representative samples. Interestingly, we show that the representativeness of the training set highly corresponds to the accuracy of the few-shot classifier. We obtain the highest accuracy when training the VAE with the most representative samples. In this case, we only use a small percentage of the whole training set, e.g., 10% for the case of *mini*Imagenet dataset, to obtain the best results. Our analyses show that this approach consistently improves the FSL classification performance by $1 \sim 2\%$ across all benchmarks for three different baselines [10, 39, 65].

Our main contributions can be summarized as follows:

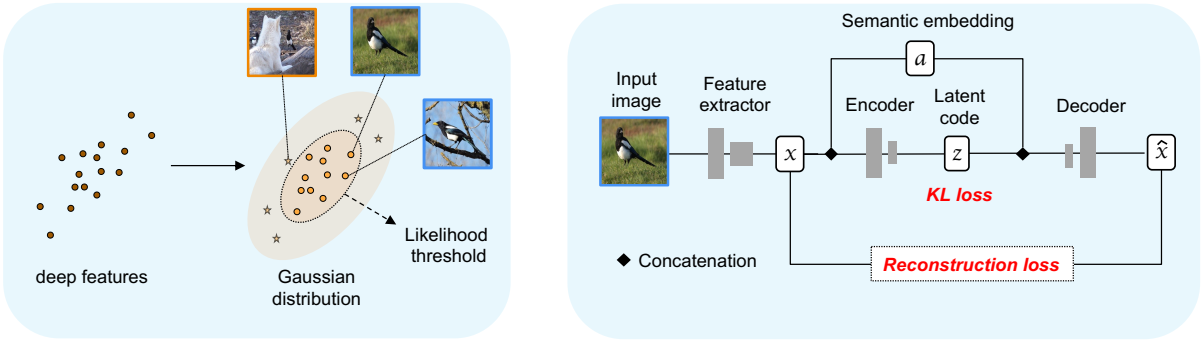
- We are the first to use a VAE-based feature generation approach conditioned on class semantic embeddings for few-shot classification.
- We propose a novel sample selection method to collect representative samples. We use these samples to train a VAE model to obtain reliable data points for constructing class-representative prototypes.
- Our experiments show that our methods achieve state-of-the-art performance on two challenging datasets, *tiered*ImageNet and *mini*ImageNet.

We summarize related FSL works in Section 2. Section 3 provides a rundown of our approach. Section 4 reports the main results obtained with our method. In section 5, we provide multiple analyses to clarify different aspects of our methods.

## 2. Related Work

**Few-shot Learning.** FSL is helpful when we only have limited labeled training data [7, 25–30]. Representative FSL approaches include metric learning based [65, 67, 68, 70, 79, 80, 83], optimization based [17, 31, 33, 34, 37, 54, 59, 62], and data augmentation based methods [2, 61, 74, 78]. Similar to our method, some FSL methods use semantic information to improve the few-shot classifiers [21, 51, 69, 77, 82]. Zhang *et al*. [82] and Xing *et al*. [77] propose methods that learn to adaptively combine the visual features and the semantic features to obtain an unified cross-modality representation for each class. These two methods focus on the fusing strategies that combine features of the two domains. Hu *et al*. [21] propose to disentangle the visual features into the sub-spaces that associate to different semantic attributes. The FSL method of Peng *et al*. [51] uses semantic information to infer a classifier for novel classes and adaptively combines this classifier with the few-shot samples. Our method is the first FSL method that uses a conditional VAE model to directly generate visual features, conditioned on the semantic embedding of each class.

**Conditional Variational Autoencoder.** The practice of using a conditional VAE to model a feature distribution has been used before in many computer vision tasks such as image classification [23, 60, 78, 84], image generation [16, 38], image restoration [14], or video processing [50]. Using VAE models for generating features conditioned on the corresponding semantic embedding is fairly common in ZSL methods [4, 18, 47, 60, 81, 85]. Mishra *et al*. [47] are the first to propose to use a conditional VAE for ZSL where they view ZSL as a case of missing data. They find that such an approach can handle well the domain shift problem. Similarly, Arora *et al*. [3] show that a conditional VAE can




**(a) Sample Selection method** **(b) Conditional VAE model**

Figure 2. **Overview –** The key aspect of our approach is to subset our training set to the most representative samples to train a conditional VAE model that generates more representative features. **(a)** To select representative samples, we assume that the features of each class follow a multivariate Gaussian distribution. We estimate the distribution parameters and compute a probability for each data point belonging to the class distribution. We identify a set of representative samples by setting a threshold on the probability. **(b)** We train a VAE to generate visual features, conditioned on the semantic embedding of each class. Using only representative samples (the output of the sample selection step) to train this VAE model improves the representativeness of the generated samples.

be used together with a GAN system to synthesize images for unseen classes effectively. Keshari *et al*. [22] focus on generating a specific set of *hard* samples which are closer to another class and the decision boundary. For the most part, ZSL methods aim to model the whole distribution of data [6,9,40,60], while our method focuses on modeling the distribution of representative samples useful for constructing the class-representative prototypes.

**Sample Selection.** To the best of our knowledge, we are the first to propose using a sample selection method for selecting training samples for a VAE model. Here we select only representative samples for training the VAE. This is a new sample selection regime since mainstream sample selection works mainly focus on identifying the most informative samples [5, 24] for training their models, which is widely used in active-learning [32, 63]. In FSL, Chang *et al*. [8] propose a method to select the most informative data that should be annotated for a few-shot text generation system. Zhou *et al*. [86] propose a method to select the useful *base classes* to train their model, while our work selects useful individual samples within an arbitrary set of base classes.

# 3. Method

## 3.1. Problem Definition

In a typical few-shot classification setting, we are given a set of data-label pairs D = $\{(x^i, y^i)\}$. Here $x^i \in R^d$ is the feature vector of a sample and $y^i \in C$, where $C$ denotes the set of classes. The set of classes is divided into base classes $C_b$ and novel classes $C_n$. The sets of class $C_b$ and $C_n$ are disjoint, *i.e.* $C_b \cap C_n = \emptyset$. For a $N$-way $K$-shot problem, we sample $N$ classes from the novel set $C_n$, and $K$ samples are available for each class. $K$ is often small (*i.e.*, $K = 1$ or $K = 5$). Our goal is to classify query samples correctly using the few samples from the support set.

## 3.2. Overall Pipeline

Fig. 2 gives an overview of our sample selection method and VAE training approach. We propose a method to select a set of representative samples from a set of base classes. We use these selected representative data to train a conditional VAE model for feature generation. To select representative samples, we assume that the features of each class follow a multivariate Gaussian distribution. We estimate the parameters for each class distribution and compute the probability for each data point belonging to its class. By setting a threshold on the probabilities, we identify a set of representative samples. We then use these selected representative samples to train a VAE model that generates samples conditioned on the semantic attributes of each class.

We train this VAE on the base classes and use the trained model to generate samples for the novel classes. The generated features are then used together with the few-shot samples to construct the prototype for each class. Our method is a simple plug-and-play module and can be built on top of any pretrained feature extractors. In our experiments, we show that our method consistently improves three baseline few-shot classification methods: Meta-Baseline [10], ProtoNet [65] and E3BM [39] by large margins.

#### 3.2.1 Class-representative Sample Selection

In this paper, we are interested in representative samples as they can serve as reliable data points for constructing a class-representative prototype [10, 65]. The main idea is to train a feature generator with only representative data to obtain more representative generated samples.

To select the representative features, we assume that the feature distribution of the base classes follows a Gaussian distribution and estimate the parameters of this distribution for each class. We calculate the Gaussian mean of a base class $i$ as the mean of every single dimension in the vector:

$$\mu^i = \frac{1}{n^i} \sum_{j=1}^{n^i} x^j, \tag{1}$$

where $x^j$ is a feature vector of the $j$-th sample from the base class $i$ and $n^i$ is the total number of samples in class $i$. The covariance matrix $\Sigma^i$ for the distribution of class $i$ is calculated as:

$$\Sigma^i = \frac{1}{n^i - 1} \sum_{j=1}^{n^i} (x^j - \mu^i)(x^j - \mu^i)^T. \tag{2}$$

Once we estimate the parameters of the Gaussian distribution using the adequate samples from the base classes, the probability density of observing a single feature, $x^j$, being generated from the Gaussian distribution of class $i$ is given by:

$$p(x^j|\mu^i, \Sigma^i) = \frac{\exp\{-\frac{1}{2}(x^j - \mu^i)^T {\Sigma^i}^{-1} (x^j - \mu^i)\}}{(2\pi)^{k/2} |\Sigma^i|^{1/2}}, \tag{3}$$

where $k$ is the dimension of the feature vector.

Here we assume that the probability of a single sample belongs to its category's distribution reflects the representativeness of the sample, *i.e.*, the higher the probability, the more representative the sample is. By setting a threshold $\epsilon$ on the estimated probability, we filter out those samples with small probabilities and get a set of representative features for class $i$:

$$\mathbb{D}^i = \{x^j \mid p(x^j|\mu^i, \Sigma^i) > \epsilon\}, \tag{4}$$

where $\mathbb{D}^i$ stores the features for class $i$ with the probabilities larger than a threshold $\epsilon$.

#### 3.2.2 Conditional VAE Model for Feature Generation

We use our sample selection method to select a set of representative samples and use them for training our feature generation model. We develop our feature generator based on a conditional variational autoencoder (VAE) architecture [66] (see Fig. 2b). The VAE is composed of an Encoder $E(x, a)$, which maps a visual feature $x$ to a latent code $z$, and a decoder $G(z, a)$ which reconstructs $x$ from $z$. Both $E$ and $G$ are conditioned on the semantic embedding $a$. The loss function for training the VAE for a feature $x^j$ of class $i$ can be defined as:

$$\begin{aligned} L_V(x^j) = &\text{KL}\left(q(z|x^j, a^i)||p(z|a^i)\right) \\ &- \log p(x^j|z, a^i), \end{aligned} \tag{5}$$

where $a^i$ is the semantic embedding of class $i$. The first term is the Kullback-Leibler divergence between the VAE posterior $q(z|x, a)$ and a prior distribution $p(z|a)$. The second term is the decoder's reconstruction error. $q(z|x, a)$ is modeled as $E(x, a)$ and $p(x|z, a)$ is equal to $G(z, a)$. The prior distribution is assumed to be $\mathcal{N}(0, I)$ for all classes.

The loss for training the feature generator is the loss over all selected representative training samples:

$$L_V = \sum_{i=1}^{C_b} \sum_{x \in \mathbb{D}^i} L_V(x) \tag{6}$$

#### 3.2.3 Constructing Class Prototypes

After the VAE is trained on the base set, we generate a set of features for a class $y$ by inputting the respective semantic vector $a^y$ and a noise vector $z$ to the decoder $G$:

$$\mathbb{G}^y = \{\hat{x} | \hat{x} = G(z, a^y), z \sim \mathcal{N}(0, I)\}. \tag{7}$$

The generated features along with the original support set features for a few-shot task is then served as the training data for a task-specific classifier. Following our baseline methods, we compute the prototype for each class and apply the nearest neighbour classifier. Specifically, we first compute two separated prototypes: one using the support features and the other using the generated features. Each prototype is the mean vector of the features of each group. We then take a weighted sum of the two prototypes to obtain the final prototype $\text{p}^y$ for class $y$:

$$\text{p}^y = w_g * \frac{1}{|\mathbb{G}^y|} \sum\nolimits_{\hat{x}^j \in \mathbb{G}^y} \hat{x}^j + w_s * \frac{1}{|\mathbb{S}^y|} \sum\nolimits_{x^j \in \mathbb{S}^y} x^j, \tag{8}$$

where $\mathbb{S}^y$ is the support set features and $(w_g, w_s)$ are the coefficients of the generated feature prototype and the real feature prototype, respectively. We classify samples by finding the nearest class prototype for an embedding query feature. We conduct further analysis to show that our generated features can benefit all types of classifiers (see Section 5.2). Compared to the methods that correct the original biased prototype, our model does not require any carefully designed combination scheme.

| Method | Backbone | *mini*ImageNet | | *tiered*ImageNet | |
|---|---|---|---|---|---|
| | | 1-shot | 5-shot | 1-shot | 5-shot |
| Matching Net [70] | ResNet-12 | 65.64 ± 0.20 | 78.72 ± 0.15 | 68.50 ± 0.92 | 80.60 ± 0.71 |
| MAML [17] | ResNet-18 | 64.06 ± 0.18 | 80.58 ± 0.12 | - | - |
| SimpleShot [72] | ResNet-18 | 62.85 ± 0.20 | 80.02 ± 0.14 | 69.09 ± 0.22 | 84.58 ± 0.16 |
| CAN [20] | ResNet-12 | 63.85 ± 0.48 | 79.44 ± 0.34 | 69.89 ± 0.51 | 84.23 ± 0.37 |
| S2M2 [44] | ResNet-18 | 64.06 ± 0.18 | 80.58 ± 0.12 | - | - |
| TADAM [48] | ResNet-12 | 58.50 ± 0.30 | 76.70 ± 0.30 | 62.13 ± 0.31 | 81.92 ± 0.30 |
| AM3 [77] | ResNet-12 | 65.30 ± 0.49 | 78.10 ± 0.36 | 69.08 ± 0.47 | 82.58 ± 0.31 |
| DSN [64] | ResNet-12 | 62.64 ± 0.66 | 78.83 ± 0.45 | 66.22 ± 0.75 | 82.79 ± 0.48 |
| Variational FSL [84] | ResNet-12 | 61.23 ± 0.26 | 77.69 ± 0.17 | - | - |
| MetaOptNet [31] | ResNet-12 | 62.64 ± 0.61 | 78.63 ± 0.46 | 65.99 ± 0.72 | 81.56 ± 0.53 |
| Robust20-distill [15] | ResNet-18 | 63.06 ± 0.61 | 80.63 ± 0.42 | 65.43 ± 0.21 | 70.44 ± 0.32 |
| FEAT [80] | ResNet-12 | 66.78 ± 0.20 | 82.05 ± 0.14 | 70.80 ± 0.23 | 84.79 ± 0.16 |
| RFS [68] | ResNet-12 | 62.02 ± 0.63 | 79.64 ± 0.44 | 69.74 ± 0.72 | 84.41 ± 0.55 |
| Neg-Cosine [36] | ResNet-12 | 63.85 ± 0.81 | 81.57 ± 0.56 | - | - |
| FRN [76] | ResNet-12 | 66.45 ± 0.19 | 82.83 ± 0.13 | 71.16 ± 0.22 | 86.01 ± 0.15 |
| Meta-Baseline [10] | ResNet-12 | 63.17 ± 0.23 | 79.26 ± 0.17 | 68.62 ± 0.27 | 83.29 ± 0.18 |
| Meta-Baseline + SVAE (Ours) | ResNet-12 | 69.96 ± 0.21 | 79.92 ± 0.16 | 73.05 ± 0.24 | 83.96 ± 0.18 |
| Meta-Baseline + R-SVAE (Ours) | ResNet-12 | 72.79 ± 0.19 | 80.70 ± 0.16 | 73.90 ± 0.24 | 84.17 ± 0.18 |
| ProtoNet [80] | ResNet-12 | 62.39 | 80.53 | 68.23 | 84.03 |
| ProtoNet + SVAE (Ours) | ResNet-12 | 73.01 ± 0.24 | 83.13 ± 0.40 | 76.36 ± 0.65 | 85.65 ± 0.50 |
| ProtoNet + R-SVAE(Ours) | ResNet-12 | **74.84 ± 0.23** | **83.28 ± 0.40** | 76.98 ± 0.65 | 85.77 ± 0.50 |
| E3BM [39] | ResNet-12 | 64.09 ± 0.37 | 80.29 ± 0.25 | 71.34 ± 0.41 | 85.82 ± 0.29 |
| E3BM + SVAE (Ours) | ResNet-12 | 73.07 ± 0.39 | 80.82 ± 0.31 | 79.85 ± 0.43 | 86.82 ± 0.32 |
| E3BM + R-SVAE(Ours) | ResNet-12 | 73.35 ± 0.37 | 80.95 ± 0.31 | **80.46 ± 0.43** | **86.99 ± 0.32** |

Table 1. **Comparison to prior works on *mini*ImageNet and *tiered*ImageNet**. Average 5-way 1-shot and 5-way 5-shot accuracy (%) with 95% confidence intervals. SVAE denotes our method using the VAE trained with all features in the base set. R-SVAE denotes the one trained with only representative features. The **best** performance is highlighted in bold.

# 4. Experiments

## 4.1. Experimental Settings

**Datasets.** We evaluate our method on two widely-used benchmarks for few-shot learning, *mini*ImageNet [55] and *tiered*ImageNet [57]. ***mini*ImageNet** is a subset of the ILSVRC-12 dataset [12]. It contains 100 classes and each class consists of 600 images. The size of each image is 84 × 84. Following the evaluation protocol of [56], we split the 100 classes into 64 base classes, 16 validation classes, and 20 novel classes for pre-training, validation, and testing. ***tiered*ImageNet** is a larger subset of ILSVRC-12 dataset, which contains 608 classes sampled from hierarchical category structure. The average number of images in each class is 1281. It is first partitioned into 34 super-categories that are split into 20 classes for training, 6 classes for validation, and 8 classes for testing. This leads to 351 actual categories for training, 97 for validation, and 160 for testing.

**Baseline methods.** Our method can be used as a simple plug-and-play module for many existing few-shot learning methods without fine-tuning their feature extractors. We investigate three baseline few-shot classification methods used in conjunction with our method: ProtoNet [80], Meta-Baseline [10] and E3BM [39]. ProtoNet is known as a strong and classic prototypical approach. In our experiments, we use the ProtoNet implementation of Ye *et al*. [80]. Meta-Baseline [10] uses a ProtoNet model to fine-tune a generic classifier via meta-learning. E3BM [39] meta-learns the ensemble of epoch-wise models to achieve robust predictions for FSL. For each baseline method, we extract the corresponding feature representations to train our feature generation VAE model. We then use the trained VAE to generate features and obtain the class prototypes for few-shot classification.

**Evaluation protocol.** We use the top-1 accuracy as the evaluation metric to measure the performance of our method. We report the accuracy on standard 5-way 1-shot and 5-shot settings with 15 query samples per class. We randomly sample 2000 episodes from the test set and report the mean accuracy with the 95% confidence interval.

## 4.2. Implementation Details

All the three baselines use ResNet12 backbone as the feature extractor. The feature representation is extracted by average pooling the final residual block outputs. The dimension of the feature representation is 640 for ProtoNet [80], 512 for Meta-Baseline [10], and 640 for E3BM [39]. For our feature generation model, both the encoder and the decoder are two-layer fully-connected (FC) networks with 4096 hidden units. LeakyReLU and ReLU [19] are the non-

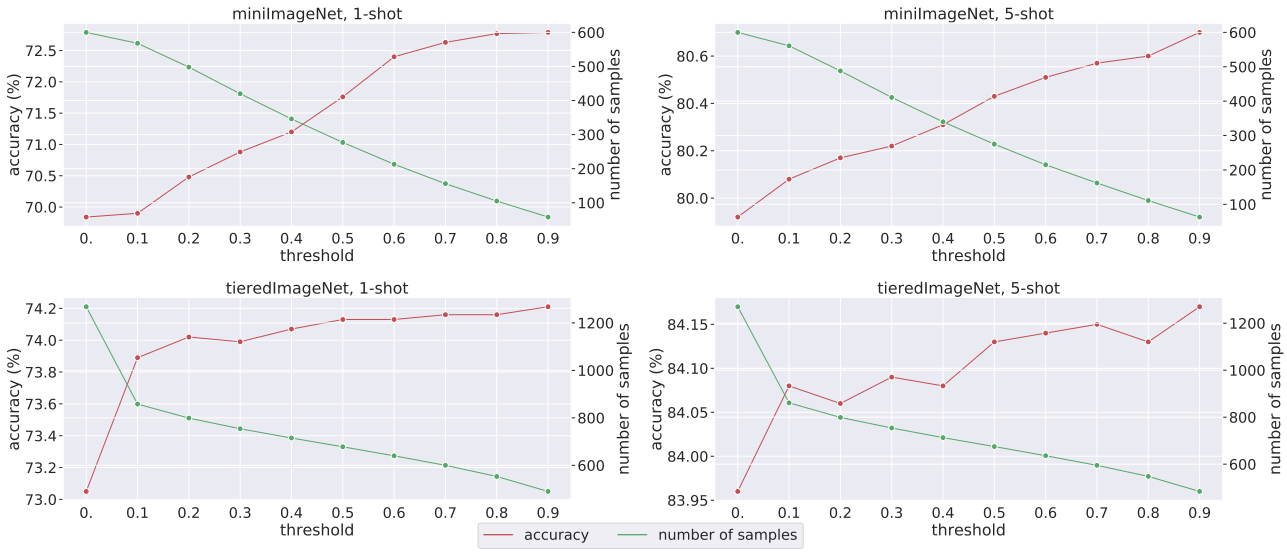



Figure 3. **Few-shot classification results with different probability thresholds.** We report the classification accuracy (%) (red) and the number of samples (green) when setting different thresholds for the probabilities. A higher threshold means we select samples that are more representative, resulting in a less amount of training data points. In general, the classification performance increases when the number of training samples decreases with increasing representativeness thresholds.

linear activation functions in the hidden and output layers, respectively. The dimensions of the latent space and the semantic vector are both set to be 512. The network is trained using the Adam optimizer with $10^{-4}$ learning rate. Our semantic embeddings are extracted from CLIP [53]. We empirically set the combination weights $[w_g, w_s]$ in Equation 8 to $[\frac{1}{2}, \frac{1}{2}]$ for 1-shot settings and to $[\frac{1}{6}, \frac{5}{6}]$ for 5-shot settings. We set the probability threshold to 0.9 for the main experiments and discuss the performance under different values of this threshold in Section 5.1.

### 4.3. Results

Table 1 presents the 5-way 1-shot and 5-way 5-shot classification results of our methods on *mini*ImageNet and *tiered*ImageNet in comparision with previous FSL methods. Here all methods use ResNet12/ResNet18 architectures as feature extractors with input images of size $84 \times 84$. Thus, the comparison is fair. For the rest of the paper, we denote our VAE trained with all data as **SVAE** (Semantic-VAE) and the model trained with only representative data as **R-SVAE** (Representative-SVAE).

We apply our methods on top of the Meta-Baseline [10], ProtoNet [80], and E3BM [39]. Our methods consistently improve all three baselines under all settings and for all datasets. They work particularly well under the 1-shot settings, in which sample bias is a more pronounced issue. Using the model trained on all data - SVAE, we report $6.8\% \sim 10\%$ 1-shot accuracy improvements for all three baselines. Our 1-shot performance for all the baselines outperforms the state-of-the-art method [76] by large margins. In 5-shot, our method consistently brings a $0.5 \sim 2.7\%$ performance gains to all baselines.

Using representative samples to train our VAE model further improves the three baseline methods under all settings and for all datasets. Compared to SVAE, training on strictly representative data improves the 1-shot classification accuracy by $0.3\% \sim 2.8\%$ and the 5-shot classification accuracy by $0.2\% \sim 0.8\%$. R-SVAE achieves state-of-the-art few-shot classification on *mini*ImageNet dataset with the ProtoNet baseline and on *tiered*ImageNet dataset with the E3BM baseline.

## 5. Analyses

All the following analyses use the feature extractor from the Meta-Baseline method [10].

### 5.1. Analysis on the Probability Threshold

In our main setting, we set a threshold of 0.9 on the probabilities to select those class-representative samples as the training data for our VAE model (the higher, the more representative). In this section, we conduct experiments with different threshold values to see how it affects the classifier's performance. Fig. 3 shows the classification accuracy under different thresholds on *mini*ImageNet and *tiered*ImageNet datasets. As the threshold increases, more non-representative samples are filtered out, resulting in less training data for R-SVAE. Interestingly, we observe that the model generally performs better with higher threshold val-

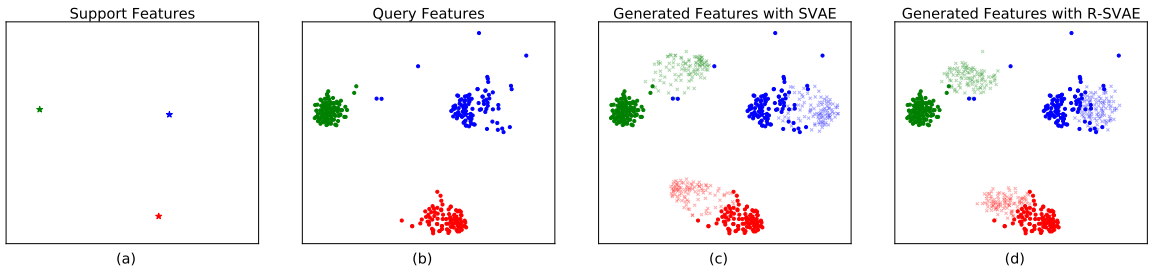



Figure 4. **Feature Visualization.** We show the t-SNE visualization of the original features (marked as dark points) and our generated features (marked as transparent points) on *tiered*ImageNet dataset. Different colors represent different classes. From left to right, we show the original support set (a), the query set (b), the features generated by SVAE (c), and the features generated by R-SVAE (d).

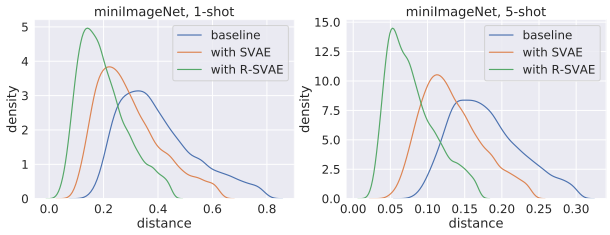



Figure 5. **Distance Distributions.** Kernel Density Estimation of the distance between the estimated prototypes and the ground truth prototype. A smaller value means the estimated prototypes are closer to the ground truth prototypes.

ues under both 1-shot and 5-shot settings. For example, under the 1-shot setting on *mini*ImageNet dataset, we only use 58 images per class on average when setting the threshold to 0.9. Training the VAE model with this small set of images improves the performance by $2.95\%$ compared with the model trained using all data in the base set with 600 images per class on average. The results suggest that the performance of our method strongly corresponds to the representativeness of training data. Moreover, it shows that our sample selection method provides a reliable measurement for the representativeness of the training samples.

## 5.2. Performance with Different Classifiers

In our main experiments, we classify samples by finding the nearest neighbor among class prototypes. In this section, we apply another three different types of classifiers: 1-nearest neighbor classifier (1-N-N), Support Vector Machine (SVM), and Logistic Regression (LR).

Table 2 shows the 1-shot performance of different classifiers using our generated features on *mini*ImageNet and *tiered*ImageNet datasets. It shows that the features generated by our VAEs improve the performance of all three classifiers. For example, the 1-shot accuracy on *mini*ImageNet using LR is improved by $8.8\%$ with SVAE and by $10.1\%$ with R-SVAE. The consistent performance improvements show that our generated features can benefit different types of classifiers.

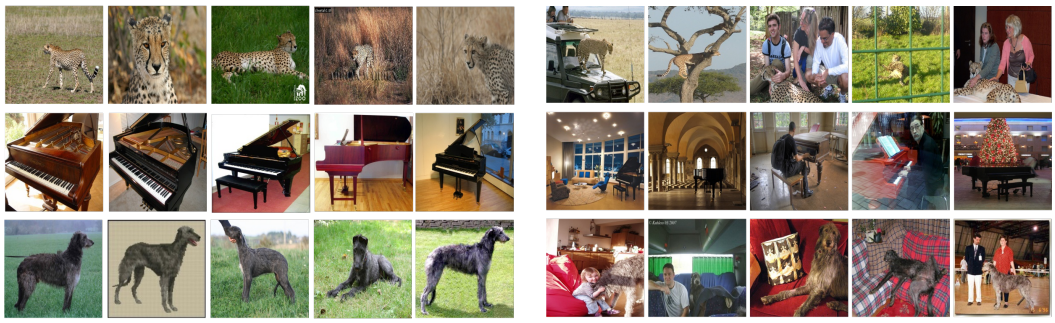
## 5.3. Feature Distribution Analysis

In Fig. 4, we show the t-SNE representation [41] of different sets of features for three classes from the novel set of *tiered*ImageNet dataset. From left to right, we visualize the distribution of the original support set (a), the query set (b), the features generated by SVAE (c), and the features generated by R-SVAE (d). Note that our methods do not rely on the support features to generate features.

Fig. 4(c) and (d) visualize the effect of our sample selection method. Fig. 4(c) visualizes features generated from our method trained with all available data from the base classes, which consist of 1281 images per class on average. In Fig. 4(d), we train the same model with only 484 representative images per class on average. Our model trained with a representative subset of data generates features that lie closer to the real features, showing the effectiveness of our sample selection method.

Moreover, we plot the distance distributions between the estimated prototypes and the ground truth prototypes of each class. Specifically, for each class, we first obtain the ground-truth prototype by taking the mean of all the features of the class. Then we calculate the $L_2$ distance between the ground truth prototype and three different prototypes: 1) Baseline: the prototype was estimated using only the support samples. 2) SVAE: the prototype was estimated using the support samples and the generated samples from our SVAE model. 3) R-SVAE: the prototype was estimated using the support samples and the generated samples from our R-SVAE model.

We sample 2400 tasks from *mini*ImageNet dataset under both 5-way 1-shot and 5-way 5-shot settings. For each task, we obtain five distances, one distance per class. Then we plot the probability density distribution of the distance, shown in Fig. 5. The probability density is calculated by binning and counting observations and then smoothing them with a Gaussian kernel, namely, Kernel Density Esti-



Representative Non-representative

Figure 6. **Examples of representative samples (left) and non-representative samples (right)**. We visualize 5 images with high probabilities and 5 images with small probabilities computed via our proposed method for 3 classes from *tiered*ImageNet dataset.

| | *mini*ImageNet | | | *tiered*ImageNet | | |
|---|---|---|---|---|---|---|
| Classifier | support samples | + SVAE | + R-SVAE | support samples | + SVAE | +R-SVAE |
| Prototype [10] | 63.17 ± 0.23 | 69.96 ± 0.21 | **72.79 ± 0.19** | 68.62 ± 0.27 | 73.05 ± 0.24 | **73.90 ± 0.24** |
| 1-N-N | 63.28 ± 0.23 | 67.25 ± 0.20 | **69.27 ± 0.19** | 68.73 ± 0.26 | 68.05 ± 0.25 | **69.82 ± 0.24** |
| SVM | 63.41 ± 0.23 | 70.30 ± 0.20 | **72.84 ± 0.19** | 68.88 ± 0.25 | 69.26 ± 0.25 | **71.28 ± 0.24** |
| LR | 63.33 ± 0.22 | 72.11 ± 0.20 | **73.41 ± 0.19** | 69.15 ± 0.25 | 74.99 ± 0.23 | **75.98 ± 0.23** |

Table 2. **Choices of the classifiers**. One-shot classification accuracy on *mini*ImageNet and *tiered*ImageNet using different types of classifiers, *i.e.*, 1-N-N, SVM and LR. All methods use the feature extractor from the Meta-Baseline method [10].

mation [11]. As can be seen the Fig., our estimated class prototypes are much closer to the ground truth prototypes, compared to the baseline.

### 5.4. Sample Visualization

In Fig. 6, we visualize some representative samples and non-representative samples based on the representativeness probability computed via our method. The samples on the left panel are images with high probabilities. These images mostly contain the main object of the category and are easy to recognize. On the contrary, the samples on the right panel are those with small probabilities. They contain various class-unrelated objects and can lead to noisy features for constructing class prototypes.

### 5.5. Performance with Different Semantic Embedding

We use CLIP features in our main experiments. The performance of our method trained with Word2Vec [45] features are shown in Table 3. Note that CLIP model is trained with 400M pairs (image and its text title) collected from the web while Word2Vec is trained with only text data. Our model outperforms state-of-the-art methods in both cases.

| | 1-shot | 5-shot |
|---|---|---|
| Meta-Baseline | 63.17 ± 0.23 | 79.26 ± 0.17 |
| Meta-Baseline + SVAE | 67.39 ± 0.21 | 79.77 ± 0.17 |
| Meta-Baseline + R-SVAE | **68.03 ± 0.22** | **79.93 ± 0.16** |

Table 3. **Classification accuracy using Word2Vec [45] as the semantic feature extractor.**

## 6. Limitations and Discussion

We propose a feature generation method using a conditional VAE model. Here we focus on modeling the distribution of the representative samples rather than the whole data distribution. To accomplish that, we propose a sample selection method to collect a set of strictly representative training samples for training our VAE model. We show that our method brings consistent performance improvements over multiple baselines and achieves state-of-the-art performance on both *mini*ImageNet and *tiered*ImageNet datasets. Our method requires a pre-trained NLP model to obtain the semantic embedding of each class. It might also inherit some potential biases from the textual domain. Note that our method does not aim to generate diverse data with large intra-class variance [35, 78]. Building a system that can generate both representative and non-representative samples can greatly benefit various downstream computer vision tasks and is an interesting direction to extend our work.

**Acknowledgements.** Jingyi Xu is partially supported by a research grant from Zebra Technologies and the SUNY2020 ITSC grant. Hieu Le is funded by Amazon Robotics to attend the conference. We thank Tran Truong, Kien Huynh, and Bento Gonçalves for proofreading the paper.

## References


[1] https://www.hillspet.com/dog-care/dog-breeds/yorkshire-terrier. 1

[2] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial eetworks. In *arXiv preprint arXiv:1711.04340*, 2018. 2

[3] Gundeep Arora, Vinay Kumar Verma, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4281–4289, 2018. 2

[4] Jimmy Ba, Kevin Swersky, Sanja Fidler, and Ruslan Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4247–4255, 2015. 2

[5] Jawadul H. Bappy, S. Paul, Ertem Tuncel, and Amit K. Roy-Chowdhury. The impact of typicality for informative representative selection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 771–780, 2017. 3

[6] Nihar Bendre, Kevin Desai, and Peyman Najafirad. Generalized zero-shot learning using multimodal variational auto-encoder with semantic concepts. *ArXiv*, abs/2106.14082, 2021. 3

[7] Alex Borowicz, Hieu Le, Grant Humphries, G. Nehls, Caroline Höschle, V. Kosarev, and H. Lynch. Aerial-trained deep learning networks for surveying cetaceans from satellite imagery. *PLoS ONE*, 14, 2019. 2

[8] Ernie Chang, Xiaoyu Shen, Hui-Syuan Yeh, and Vera Demberg. On training instance selection for few-shot neural text generation. *ArXiv*, abs/2107.03176, 2021. 3

[9] Yu chao Gu, Le Zhang, Yun Liu, Shao-Ping Lu, and Ming-Ming Cheng. Generalized zero-shot learning via vae-conditioned generative flow. *ArXiv*, abs/2009.00303, 2020. 3

[10] Yinbo Chen, Zhuang Liu, Huijuan Xu, Trevor Darrell, and Xiaolong Wang. Meta-baseline: Exploring simple meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9062–9071, 2021. 1, 2, 3, 4, 5, 6, 8

[11] Yen-Chi Chen. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*, 1:161 – 187, 2017. 8

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 1

[14] Yingjun Du, Jun Xu, Xiantong Zhen, Ming-Ming Cheng, and Ling Shao. Conditional variational image deraining. *IEEE Transactions on Image Processing*, 29:6288–6301, 2020. 2

[15] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. pages 3722–3730, 2019. 5

[16] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018. 2

[17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning(ICML)*, 2017. 2, 5

[18] Jingcai Guo and Song Guo. A novel perspective to zero-shot learning: Towards an alignment of manifold structures via semantic feature expansion. *IEEE Transactions on Multimedia*, 23:524–537, 2021. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 5

[20] Ruibing Hou, Hong Chang, Bingpeng Ma, S. Shan, and Xilin Chen. Cross attention network for few-shot classification. In *NeurIPS*, 2019. 5

[21] Ping Hu, Ximeng Sun, Kate Saenko, and Stan Sclaroff. Weakly-supervised compositional featureaggregation for few-shot recognition. volume abs/1906.04833, 2019. 2

[22] Rohit Keshari, Richa Singh, and Mayank Vatsa. Generalized zero-shot learning via over-complete distribution. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13297–13305, 2020. 3

[23] Junsik Kim, Tae-Hyun Oh, Seokju Lee, Fei Pan, and In So Kweon. Variational prototyping-encoder: One-shot learning with prototypical images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[24] Hieu Le, Bento Goncalves, Dimitris Samaras, and Heather Lynch. Weakly labeling the antarctic: The penguin colony case. In *CVPR Workshops*, June 2019. 3

[25] Hieu Le, Vu Nguyen, Chen-Ping Yu, and D. Samaras. Geodesic distance histogram feature for video segmentation. *ACCV*, 2016. 2

[26] Hieu Le and Dimitris Samaras. Physics-based shadow image decomposition for shadow removal. *IEEE TPAMI*. 2

[27] Hieu Le and Dimitris Samaras. Shadow removal via shadow image decomposition. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[28] Hieu Le and Dimitris Samaras. From shadow segmentation to shadow removal. In *European Conference on Computer Vision(ECCV)*, 2020. 2

[29] Hieu Le, Tomas F. Yago Vicente, Vu Nguyen, Minh Hoai, and Dimitris Samaras. A+D Net: Training a shadow detector with adversarial shadow attenuation. In *European Conference on Computer Vision(ECCV)*, 2018. 2

[30] Hieu Le, Chen-Ping Yu, Gregory Zelinsky, and Dimitris Samaras. Co-localization with category-consistent features and geodesic distance propagation. In *ICCV Workshop*, 2017. 2

[31] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 5

[32] X. Li and Yuhong Guo. Adaptive active learning for image classification. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–866, 2013. 3

[33] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. In *arXiv preprint arXiv:1707.09835*, 2017. 2

[34] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. pages 9250–9259, 2019. 2

[35] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In *European Conference on Computer Vision(ECCV)*, 2018. 8

[36] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In *ECCV*, 2020. 5

[37] Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. In *ECCV*, 2020. 1, 2

[38] Ming-Yu Liu, Thomas M. Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017. 2

[39] Yaoyao Liu, Bernt Schiele, and Qianru Sun. An ensemble of epoch-wise empirical bayes for few-shot learning. In *ECCV*, 2020. 1, 2, 3, 5, 6

[40] Peirong Ma and Xiao Hu. A variational autoencoder with deep embedding model for generalized zero-shot learning. In *AAAI*, 2020. 3

[41] L. V. D. Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. In *Journal of Machine Learning Research*, 2008. 7

[42] Anay Majee, Kshitij Agrawal, and A. Subramanian. Few-shot learning for road object detection. *ArXiv*, abs/2101.12543, 2021. 1

[43] Anay Majee, A. Subramanian, and Kshitij Agrawal. Meta guided metric learner for overcoming class confusion in few-shot road object detection. *ArXiv*, abs/2110.15074, 2021. 1

[44] Puneet Mangla, Mayank Kumar Singh, Abhishek Sinha, Nupur Kumari, Vineeth N. Balasubramanian, and Balaji Krishnamurthy. Charting the right manifold: Manifold mixup for few-shot learning. pages 2207–2216, 2020. 5

[45] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. 8

[46] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38:39–41, 1992. 1

[47] Ashish Mishra, M. Shiva Krishna Reddy, Anurag Mittal, and Hema A. Murthy. A generative model for zero shot learning using conditional variational autoencoders. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2269–22698, 2018. 2

[48] Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018. 5

[49] Cheng Ouyang, Carlo Biffi, Chen Chen, Turkay Kart, Huaqi Qiu, and Daniel Rueckert. Self-supervision with superpixels: Training few-shot medical image segmentation without annotation. In *ECCV*, 2020. 1

[50] Junting Pan, Chengyu Wang, Xu Jia, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. Video generation from single semantic label map. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3728–3737, 2019. 2

[51] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. Few-shot image recognition with knowledge transfer. pages 441–449, 2019. 2

[52] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 1

[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 6

[54] Aravind Rajeswaran, Chelsea Finn, S. Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *NeurIPS*, 2019. 2

[55] Sachin Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 5

[56] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 5

[57] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, H. Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. 2017. 5

[58] Mahdi Rezaei and Mahsa Shahidi. Zero-shot learning and its applications from autonomous vehicles to covid-19 diagnosis: A review. *Intelligence-Based Medicine*, 3:100005 – 100005, 2020. 1

[59] Adam Santoro, Sergey Bartunov, M. Botvinick, Daan Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 2

[60] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero- and few-shot learning via aligned variational autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[61] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Rogerio Feris, Abhishek Kumar, Raja Giryes, and Alex M. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, June 2018. 2

[62] Tyler R. Scott, K. Ridgeway, and M. Mozer. Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning. In *NeurIPS*, 2018. 2

[63] Burr Settles. Active learning literature survey. 2009. 3

[64] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Tafazzoli Harandi. Adaptive subspaces for few-shot learning. pages 4135–4144, 2020. 5

[65] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1, 2, 3, 4

[66] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, 2015. 2, 4

[67] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018. 2

[68] Yonglong Tian, Yue Wang, Dilip Krishnan, J. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? volume abs/2003.11539, 2020. 2, 5

[69] Pavel Tokmakov, Yu-Xiong Wang, and Martial Hebert. Learning compositional representations for few-shot recognition. pages 6371–6380, 2019. 2

[70] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 2, 5

[71] Wenji Wang, Qing Xia, Zhiqiang Hu, Zhennan Yan, Zhuowei Li, Yang Wu, Ning Huang, Yue Gao, Dimitris N. Metaxas, and Shaoting Zhang. Few-shot learning by a cascaded framework with shape-constrained pseudo label assessment for whole heart segmentation. *IEEE Transactions on Medical Imaging*, 40:2629–2641, 2021. 1

[72] Yan Wang, Wei-Lun Chao, Kilian Q. Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. volume abs/1911.04623, 2019. 5

[73] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *arXiv: Learning*, 2019. 1

[74] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018. 2

[75] 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. 1

[76] Davis Wertheimer, Luming Tang, and Bharath Hariharan. Few-shot classification with feature map reconstruction networks. pages 8008–8017, 2021. 2, 5, 6

[77] Chen Xing, Negar Rostamzadeh, Boris N. Oreshkin, and Pedro H. O. Pinheiro. Adaptive cross-modal few-shot learning. In *NeurIPS*, 2019. 1, 2, 5

[78] Jingyi Xu, Hieu Le, Mingzhen Huang, ShahRukh Athar, and Dimitris Samaras. Variational feature disentangling for fine-grained few-shot classification. In *ICCV*, 2021. 2, 8

[79] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. volume abs/2101.06395, 2021. 1, 2

[80] Han-Jia Ye, Hexiang Hu, D. Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. pages 8805–8814, 2020. 2, 5, 6

[81] Yunlong Yu, Zhong Ji, Jungong Han, and Zhongfei Zhang. Episode-based prototype generating network for zero-shot learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14032–14041, 2020. 2

[82] Baoquan Zhang, Xutao Li, Yunming Ye, Zhichao Huang, and Lisai Zhang. Prototype completion with primitive knowledge for few-shot learning. In *CVPR*, pages 3754–3762, 2021. 1, 2

[83] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. pages 12200–12210, 2020. 2

[84] Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. Variational few-shot learning. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 5

[85] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3010–3019, 2017. 2

[86] Linjun Zhou, Peng Cui, Xu Jia, Shiqiang Yang, and Qi Tian. Learning to select base classes for few-shot classification. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4623–4632, 2020. 3