

Variational Feature Disentangling for Fine-Grained Few-Shot Classification

Jingyi Xu¹, Hieu Le^{*2}, Mingzhen Huang¹, ShahRukh Athar¹, and Dimitris Samaras¹

¹Department of Computer Science, Stony Brook University, NY, USA

²Amazon Robotics, MA, USA

Abstract

Data augmentation is an intuitive step towards solving the problem of few-shot classification. However, ensuring both discriminability and diversity in the augmented samples is challenging. To address this, we propose a feature disentanglement framework that allows us to augment features with randomly sampled intra-class variations while preserving their class-discriminative features. Specifically, we disentangle a feature representation into two components: one represents the intra-class variance and the other encodes the class-discriminative information. We assume that the intra-class variance induced by variations in poses, backgrounds, or illumination conditions is shared across all classes and can be modelled via a common distribution. Then we sample features repeatedly from the learned intra-class variability distribution and add them to the class-discriminative features to get the augmented features. Such a data augmentation scheme ensures that the augmented features inherit crucial class-discriminative features while exhibiting large intra-class variance. Our method significantly outperforms the state-of-the-art methods on multiple challenging fine-grained few-shot image classification benchmarks. Code is available at: <https://github.com/cvlab-stonybrook/vfd-iccv21>

1. Introduction

Fine-grained visual data are hard to collect and costly to annotate [49, 46, 15]. Fine-grained visual datasets often become quite long-tailed and lead to classifiers overfitting to the abundant classes when trained in vanilla settings. Fine-grained few-shot learning (FSL) methods alleviate this problem since they learn discriminative class features, among visually similar classes, using as few as 5 or 1 training instances.

Augmenting the few-shot classes by generating addi-

^{*}Work done prior to Amazon

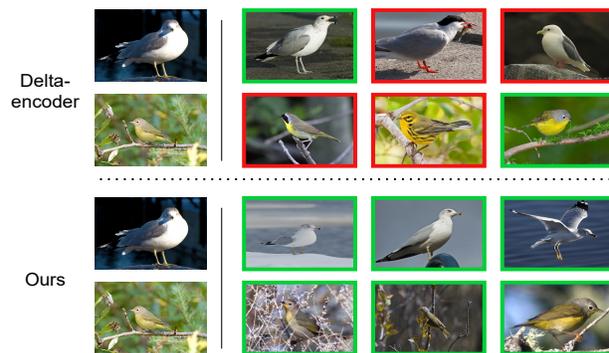


Figure 1: **Nearest “real sample” neighbors of the augmented examples.** We train data augmentation methods using the base classes and search for the nearest-neighbors of the generated samples in the novel classes of the CUB dataset. The input images are shown in the first column. Each row shows the nearest neighbors of some augmented features computed from: Δ -encoder [38] (1st and 2nd rows) and our method (3rd and 4th rows). Green borders indicate that the images have the same class as the input image and red borders indicate otherwise.

tional data is a straightforward way to mitigate issues of overfitting in FSL. Nevertheless, generating diverse data reliably remains an open question [19, 40]. The generated samples should contain the class-discriminative features while exhibiting high intra-class diversity. A typical data synthesis approach is generating new samples based on adversarial frameworks [45, 25, 11, 55, 2, 20, 21, 22]. However, these methods suffer from a lack of diversity in the generated samples as adversarial training often mode-collapses. Another approach is the feature transfer that transfers the intra-class variance from the base classes, which have many training samples, to augment features for the novel classes, in which only few samples are available [38, 52, 12]. These methods are based on a common assumption that intra-class variations induced by poses, back-

grounds, or illumination conditions are shared across categories. The intra-class variations are either modelled as low-level statistics [52] or pairwise variations [38, 12] and are applied directly on the novel samples. In this paper, we discuss two potential issues with these approaches. First, these transformations can introduce certain class-discriminative features that could alter the class-identity of the transformed features. For example, only 8.7% of the augmented features using the Δ -encoder[38] have their nearest “real sample” neighbors belong to the same classes as the original samples (see Fig. 1). Second, the extracted variations might not be relevant to a specific novel sample, i.e., some bird species would never appear in sea backgrounds. Applying irrelevant variations would result in noisy or meaningless samples and degrade classification results (see Sec. 6.1). These two issues are more pronounced for fine-grained classification since a small change in feature space might change the category of the feature due to the small inter-class distances.

We address these issues in this paper via a novel data augmentation framework. First, we disentangle each feature into two components: one that captures the intra-class variance, which we refer as intra-class variance features, and the other that encodes the class-discriminative features. Second, we model intra-class variance via a common distribution from which we can easily sample the new intra-class variations that are relevant for diversifying a specific instance. We show that both the feature disentanglement and the distribution of intra-class variability can be approximated using data from the base classes and it generalizes well to the novel classes. The two key supervision signals that drive the training of our framework are: 1) A classification loss that ensures that the class-discriminative features contain class specific information, 2) A Variational Auto-Encoder (VAE) [18] system that explicitly models intra-class variance via an isotropic Gaussian distribution.

Our method works especially well for fine-grained datasets where the intra-class variations are similar across classes, achieving state-of-the-art few-shot classification performances on the CUB[49], NAB[46], and Stanford Dogs[15] datasets, outperforming previous methods [38, 24] by a large margin. We show in our analyses that the data generated by our method lies closely to the real-and-unseen features of the same class and can closely approximate the distribution of the real data.

To sum up, our contributions are:

1. We are the first to propose a VAE-based feature disentanglement method for fine-grained FSL.
2. We show that we can train such a system using sufficient data from the base classes. We can sample from the learnt distribution to obtain relevant variations to diversify novel training instances in a reliable manner.

3. Our method outperforms state-of-the-art FSL methods in multiple fine-grained datasets by a large margin.

2. Related Work

FSL methods can be broadly organized into three categories: metric learning based, optimization based and data augmentation based.

Metric learning based methods [47, 41, 43, 51, 50, 44, 53] utilize the similarities between images to regularize the embedding space. Matching Networks [47] use an attention mechanism over a learned embedding of the labeled set of examples to predict classes for the unlabeled points. The Prototypical Network [41] learns to classify query samples based on their Euclidean distance to prototype representations of each class. Sung *et al.* [43] propose to measure the distance metric with a CNN-based relation module.

Optimization based methods [10, 27, 24, 28, 36, 34, 39] aim to design models that can generalize to new tasks efficiently. MAML [10] uses a meta-learner to find an initialization which can be adapted to new categories within few gradient updates using small training data. Meta-SGD [27] learns to learn not only the learner initialization but also the learner update direction and learning rate. Lee *et al.* propose MetaOptNet [24], which uses discriminatively trained linear predictors as base learners to learn feature representations for FSL.

Data augmentation based methods [2, 38, 48] generate additional training examples to alleviate the problem of data insufficiency. DAGAN [2] uses conditional generative adversarial network (GAN) to transform image features, which can be applied to novel unseen classes of data. Wang *et al.* [48] propose to combine a meta-learner with a hallucinator, which can effectively hallucinate novel instances of new classes. Some methods transfer intra-class variance from the base classes to the novel class. The Δ -encoder [38] extracts transferable intra-class deformations from image pairs of the same class and uses them to augment samples of the novel classes. Differently, our method learns a common distribution of the intra-class variability without pairing any images.

Our method uses a VAE to model intra-class variance. VAEs have been used in FSL [54, 37, 16]. Zhang *et al.* [54] estimate a distribution for each class and compute the probability of a novel input to perform classification. Schonfeld *et al.* [37] learn a shared latent space of image features and class embeddings via aligned VAEs. Instead of modelling the whole image feature, we only use variational inference to model the intra-class variance for our data augmentation framework. The concept of feature disentanglement using VAEs has been used in various applications such as person re-identification [9, 57], metric learning [29], or multi-domain image translation [30, 23]. We are the first to pro-

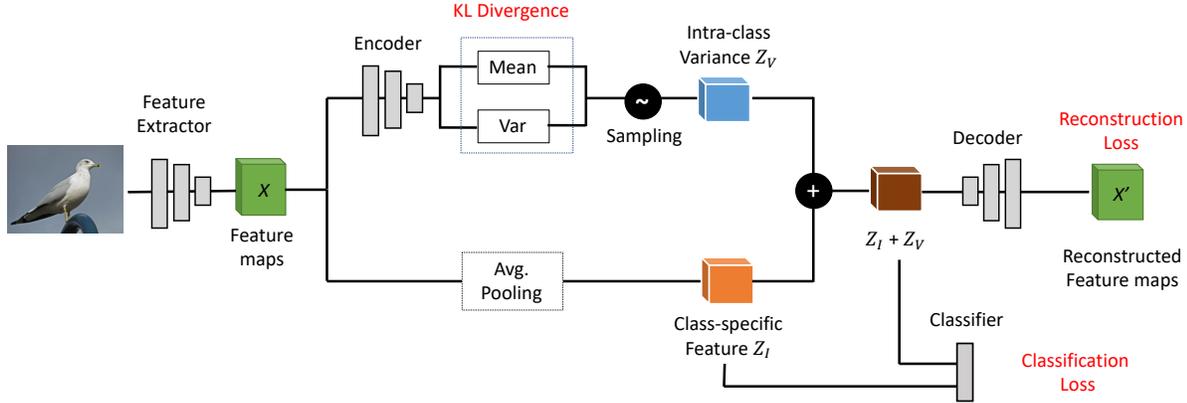


Figure 2: **The pipeline of our proposed method.** The input image is mapped into the image feature maps X . We input X into an Encoder to obtain the mean and variance of the intra-class variability distribution that are used to sample the intra-class variance feature z_V . The class-specific feature z_I is obtained by max-pooling X . z_V is forced to follow an isotropic multivariate Gaussian distribution. Both z_I and the combined features are used to train a classifier. We sample from the learned distribution repeatedly to get multiple z_V and add them to the class-specific feature z_I to get the augmented features. These augmented features are used together with the original ones to train a more robust classifier.

pose such a VAE-based feature disentanglement scheme for FSL problems. We show that such a model trained on base classes can be used to effectively augment data for the novel classes and significantly improve the classification results.

3. Few-shot Learning Preliminaries

In FSL, abundant labeled images of base classes and a small number of labeled images of novel classes are given. Our goal is to train a classifier that can correctly classify novel class images with the few given examples. The standard FSL procedure includes a training stage and a fine-tuning stage. During the training stage, we use base class images to train a feature extractor and the classifier. Then in the fine-tuning stage, we freeze the parameters of the pre-trained feature extractor and train a new classifier head using the few labeled examples in the novel classes. In the testing stage, the learned classifier predicts labels on a set of unseen novel class images.

Since the available samples during the fine-tuning stage are scarce and lack diversity, the learned classifier tends to overfit to the few samples and thus performs poorly on the test images. To address this, we augment the training samples with our proposed data augmentation method, which significantly improves the performance of the baseline.

4. Method

Our goal is to generate additional features of the few novel class images which contain larger intra-class variance. Fig. 2 illustrates the pipeline of our proposed method. We decompose the feature representation of an input image into two components, the class-specific feature z_I and the

intra-class variance feature z_V . z_V is constrained to follow a prior distribution. Then we repeatedly sample new intra-class variance features \tilde{z}_V from the distribution and add them to the class-specific feature z_I to get augmented features. The augmented features are used together with the original features to train the final classifier. In the following sections, we will describe how we model the distribution of intra-class variability via variational inference and how we use it to diversify samples from the novel set.

4.1. Variational Inference for Intra-class Variance

Given an input image (i), we first use a feature extractor to map it into a feature map $X^{(i)}$. We then compute the intra-class variance feature $z_V^{(i)}$ and the class-specific feature $z_I^{(i)}$ from $X^{(i)}$ such that the embedding of the input image, $z^{(i)}$, can be expressed as:

$$z^{(i)} = z_I^{(i)} + z_V^{(i)}. \quad (1)$$

Here we assume that the intra-class variance feature is generated from some conditional distribution $p(z_V)$ and the feature map $X^{(i)}$ is generated from some conditional distribution $p(X|z)$.

The class-specific feature $z_I^{(i)}$ can be learned by minimizing the cross-entropy loss given the class label $y^{(i)}$:

$$L_{cls}(X^{(i)}) = L_{cross-entropy}(W(z_I^{(i)}), y^{(i)}) \quad (2)$$

where W is a classifier with a single fully connected layer.

We use variational inference to model the posterior distribution of the variable z_V . Specifically, we approximate

the true posterior distribution $p(z_V|X)$ with another distribution $q(z_V|X)$. The Kullback-Leibler divergence between the true distribution and the approximation is:

$$KL[q(z_V|X)||p(z_V|X)] = \int_Z q(Z|X) \log \frac{q(Z|X)}{p(Z|X)}. \quad (3)$$

Since the Kullback-Leibler divergence is always greater than or equal to zero, maximizing the marginal likelihood $p(X^{(i)})$ is equivalent to maximizing the evidence lower bound (ELBO) defined as follows:

$$ELBO^{(i)} = E_{q(z_V^{(i)}|X^{(i)})} [\log p(X^{(i)}|z_V^{(i)})] - KL(q(z_V^{(i)}|X^{(i)})||p(z_V)). \quad (4)$$

Prior work [29, 52, 54] has shown that the distribution of intra-class variability can be modelled with a Gaussian distribution. Here we set the prior distribution of z_V to be a centered isotropic multivariate Gaussian: $p(z_V) = \mathcal{N}(0, I)$. For the posterior distribution, we set it to be a multivariate Gaussian with diagonal covariance:

$$q(z_V^{(i)}|X^{(i)}) = \mathcal{N}(\mu^{(i)}, \sigma^{(i)}), \quad (5)$$

where $\mu^{(i)}$ and $\sigma^{(i)}$ are computed by a probabilistic encoder. With the reparameterization trick, we obtain $z_V^{(i)}$ as follows:

$$z_V^{(i)} = \mu^{(i)} + \sigma^{(i)} * \epsilon, \epsilon \sim \mathcal{N}(0, I). \quad (6)$$

Since $z_I^{(i)}$ is deterministic given $X^{(i)}$, we have $p(X^{(i)}|z_V^{(i)}) = p(X^{(i)}|z_V^{(i)}, z_I^{(i)}) = p(X^{(i)}|z^{(i)})$. To estimate the maximum likelihood $p(X^{(i)}|z^{(i)})$, we use a decoder to reconstruct the original feature map from $z^{(i)}$ and minimize the $L2$ distance between the original feature map and the reconstructed one.

From Eq. 4, we now derive the loss function for the modeling of intra-class variance:

$$L_{intra}(X^{(i)}) = \|X^{(i)} - \hat{X}^{(i)}\|^2 + KL(q(z_V^{(i)}|X^{(i)})||p(z_V)), \quad (7)$$

where $\hat{X}^{(i)}$ is the reconstructed feature map synthesized from the sum of class-specific feature $z_I^{(i)}$ and intra-class variance feature $z_V^{(i)}$ sampled from the distribution $\mathcal{N}(\mu^{(i)}, \sigma^{(i)})$.

The L_{intra} loss includes two terms. The first term is the reconstruction term, which ensures that the encoder extracts meaningful information from the inputs. The second term is a regularization term, which forces the latent code, $z_V^{(i)}$, to follow a standard normal distribution. Here, instead of minimizing the Kullback-Leibler divergence directly, we decompose it into three terms as in [5]:

$$KL[q(z_V|X)||p(z_V)] = KL(q(z_V, X)||q(z_V)p(X)) + KL(q(z_V)||\prod_j q(z_{V_j})) + \sum_j KL(q(z_{V_j})||p(z_{V_j})), \quad (8)$$

where z_{V_j} denotes the j -th dimension of the latent variable.

The three terms in Eq. 8 are referred to as the *index-code mutual information*, *total correlation*, and *dimension-wise KL* respectively. Prior work [5, 1, 4] has shown that penalizing the index-code mutual information and total correlation terms leads to a more disentangled representation while the dimension-wise KL term ensures that the latent variables do not deviate too far from the prior. Similar to [5], we penalize the total correlation with a weight α and rewrite L_{intra} as follows:

$$L_{intra}(X^{(i)}) = \|X^{(i)} - \hat{X}^{(i)}\|^2 + KL(q(z_V^{(i)}, X^{(i)})||q(z_V^{(i)})p(X)) + \alpha * KL\left(q(z_V^{(i)})||\prod_j q(z_{V_j}^{(i)})\right) + \sum_j KL(q(z_{V_j}^{(i)})||p(z_{V_j})). \quad (9)$$

The combination of L_{cls} and L_{intra} drives the model to extract discriminative class-specific features $z_I^{(i)}$ and model the distribution of intra-class variability simultaneously.

4.2. Objective Function

Given the distribution of intra-class variability, we can generate additional samples for the base classes during the training stage. For input image (i) with extracted class-specific feature $z_I^{(i)}$ and intra-class variability mean and variance $\mu^{(i)}$ and $\sigma^{(i)}$ respectively, we sample new intra-class variance features, $\tilde{z}_V^{(i)}$, for this image from the distribution $\mathcal{N}(\mu^{(i)}, \sigma^{(i)})$ and add them to $z_I^{(i)}$ to obtain the augmented features $\tilde{z}^{(i)} = z_I^{(i)} + \tilde{z}_V^{(i)}$. We use these features to train our system using the following cross-entropy loss:

$$L_{aug}(X^{(i)}) = L_{cross-entropy}(W(\tilde{z}^{(i)}), y^{(i)}) \quad (10)$$

The overall loss function in the training stage is a weighted combination of the aforementioned terms:

$$L = L_{cls} + L_{intra} + \beta * L_{aug} \quad (11)$$

where β is the coefficient of L_{aug} .

4.3. Diversifying Samples for Few-Shot Classes

In this section, we discuss how to use our model to diversify samples for few-shot classes. Our intra-class variance is modelled by an isotropic Gaussian distribution. Sampling from this distribution would result in an arbitrary intra-class variance feature. However, we conjecture that such an arbitrary feature may not be relevant for all instances, i.e., some birds never appear with a background of the sea. Note that here as all intra-class variations are mapped into a common continuous embedding space via variational inference and closely related or similar intra-class variations likely form local neighborhoods in the embedding space. Thus, instead

of sampling from the zero-mean and unit-variance distribution, we only sample from the mean and variance estimated directly from the conditional sample to obtain the likely relevant intra-class variations to this sample.

Specifically, given an image of novel class $(i)^*$ with class label $y^{(i)^*}$, we first extract the feature map $X^{(i)^*}$, the class-specific feature $z_I^{(i)^*}$, and the mean and variance of the intra-class variability distribution $\mu^{(i)^*}$ and $\sigma^{(i)^*}$ for this instance. We then generate additional features by adding the class-specific features $z_I^{(i)^*}$ with a biased term sampled from the distribution of intra-class variability.

$$\tilde{z}^{(i)^*} = z_I^{(i)^*} + \tilde{z}_V^{(i)^*}, \tilde{z}_V^{(i)^*} \sim N(\mu^{(i)^*}, \sigma^{(i)^*}), \quad (12)$$

where $\tilde{z}^{(i)^*}$ is the augmented feature and $\tilde{z}_V^{(i)^*}$ is sampled from the posterior distribution $N(\mu^{(i)^*}, \sigma^{(i)^*})$. By sampling from $N(\mu^{(i)^*}, \sigma^{(i)^*})$ multiple times, we get multiple augmented features $\tilde{z}^{(i)^*}$ that can be used to train the classifier. In Sec. 6.1, we verify the effectiveness of this sampling scheme.

5. Experiments

5.1. Datasets

We evaluate our method on three fine-grained image classification datasets: Caltech UCSD Birds (CUB) [49], North America Birds (NAB) [46] and Stanford Dogs [15]. The CUB dataset contains 11,788 bird images from 200 bird species in total. Following the setup introduced in [49], we sample the base classes from the 100 classes provided for training, and sample the novel set from the 50 classes provided for testing. The NAB dataset contains 48,527 bird images with 555 classes, which is four times larger than CUB. Similar to [45], we adopt a 2:1:1 training, validation and test set split. The Stanford Dogs dataset is a subset of the Imagenet dataset designed for fine-grained image classification with 90 categories for training and validation and 30 testing categories.

5.2. Implementation Details

We conduct experiments with two architectures of our feature extractor: ResNet12 and Conv4 for fair comparisons with other methods using similar architectures. **ResNet12** [13] contains 4 Residual blocks. Each residual block is composed of 3 *conv* layers with 3×3 kernels. A 2×2 max-pooling layer is applied at the end of each residual block. **Conv4** consists of 4 layers with 3×3 convolutions and 32 filters, followed by batch normalization (BN), a ReLU non-linearity, and 2×2 max-pooling.

The class-specific features are calculated by average-pooling the output of the feature extractor. The encoder consists of three *conv* blocks followed by two fully-connected heads that output the μ and $\log\sigma^2$ respectively. The decoder

consists of a fully connected layer followed by three Convolutional blocks.

Training policies. The whole network is trained from scratch in an end-to-end manner. In the training stage, we use the Adam optimizer [17] on all datasets with initial learning rate 0.001. We train our model for 100 epochs in total with a batch size of 16 and reduce the learning rate by 0.1 at the 40-th and 80-th epochs. We empirically set $\alpha = 4$ in Eq. 9 and $\beta = 1$ in Eq. 11.

We follow a standard few-shot evaluation scheme. In the fine-tuning stage, we select 5 classes from the novel classes randomly. For each class, we pick k instances as the support set and 16 instances for the query set for a k -shot task. The extracted features of all support set images along with the augmented features are used to train a linear classifier for 100 iterations with a batch size of 4. For each feature extracted from a support image, we obtain five augmented features. The final results are averaged over 600 experiments. For data augmentation, we adopt random cropping, horizontal flipping and color jittering as in [6]. The final size of the input images is 84×84 .

5.3. Results

Tab. 1 summarizes the 5-way classification accuracy of various methods using ResNet12 backbones. The results are obtained using the publicly available code of each method. Our proposed method outperforms the previous methods by a large margin for both 1-shot and 5-shot settings on all three datasets. Compared with the Δ -encoder [38], another data augmentation based method, our proposed method achieves 7.40%, 9.20% and 7.65% performance gain for the 1-shot setting and 5.88%, 2.90% and 9.40% performance gain for the 5-shot setting on the three datasets respectively. It can be seen that our improvement in the 1-shot setting is more pronounced than in the 5-shot setting since the 1-shot setting is a more extreme case of data scarcity, in which augmenting the training data tends to be more useful.

We compare with methods using Conv4 architectures as the backbone networks in Tab. 2. Here the majority of methods only report their results on the CUB and Stanford Dogs datasets. Our proposed method achieves state-of-the-art performance for both the 1-shot and 5-shot settings. Especially for the 1-shot setting, our method obtains 2.12% performance gain for the CUB and 2.19% gain for the Stanford Dogs over MattML [56], a newly proposed method that is aimed specifically at fine-grained few-shot visual recognition.

Our method also achieves competitive few-shot classification performances on non fine-grained datasets such as CIFAR-FS[3] and mini-ImageNet[47, 35]. More details can be found in the supplementary material.

Method	CUB		NAB		Stanford Dogs	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Baseline [6]	63.90 ± 0.88	82.54 ± 0.54	70.36 ± 0.89	87.91 ± 0.49	63.53 ± 0.89	79.95 ± 0.59
Baseline++ [6]	68.46 ± 0.85	81.02 ± 0.46	76.00 ± 0.85	90.99 ± 0.41	58.30 ± 0.35	73.77 ± 0.68
MAML [10]	71.11 ± 1.00	82.08 ± 0.72	80.08 ± 0.93	88.87 ± 0.54	66.56 ± 0.66	79.32 ± 0.35
MatchingNet [47]	72.62 ± 0.90	84.14 ± 0.50	73.91 ± 0.72	88.17 ± 0.45	65.87 ± 0.81	80.70 ± 0.42
ProtoNet [41]	71.57 ± 0.89	86.37 ± 0.49	73.60 ± 0.83	89.72 ± 0.41	65.02 ± 0.92	83.69 ± 0.48
RelationNet [43]	70.20 ± 0.84	84.28 ± 0.46	67.41 ± 0.82	85.47 ± 0.43	59.38 ± 0.79	79.10 ± 0.37
MTL [42]	73.31 ± 0.92	82.29 ± 0.51	78.69 ± 0.78	87.74 ± 0.34	54.96 ± 1.03	68.76 ± 0.65
Δ -encoder [38]	73.91 ± 0.87	85.60 ± 0.62	79.42 ± 0.77	92.32 ± 0.59	68.59 ± 0.53	78.60 ± 0.78
MetaOptNet [24]	75.15 ± 0.46	87.09 ± 0.30	84.56 ± 0.46	93.31 ± 0.22	65.48 ± 0.49	79.39 ± 0.25
Ours	79.12 ± 0.83	91.48 ± 0.39	88.62 ± 0.73	95.22 ± 0.32	76.24 ± 0.87	88.00 ± 0.47

Table 1: Few-shot classification accuracy on the CUB [49], NAB [46], and Stanford Dogs [15] dataset. All experiments are from 5-way classification with the same backbone network (ResNet12). The best performance is indicated in bold.

Method	CUB		Stanford Dogs	
	1-shot	5-shot	1-shot	5-shot
MatchingNet [47]	45.30 ± 1.03	59.50 ± 1.01	35.80 ± 0.99	47.50 ± 1.03
ProtoNet [41]	37.36 ± 1.00	45.28 ± 1.03	37.59 ± 1.00	48.19 ± 1.03
RelationNet [43]	58.99 ± 0.52	71.20 ± 0.40	43.29 ± 0.46	55.15 ± 0.39
MAML [10]	58.13 ± 0.36	71.51 ± 0.30	44.84 ± 0.31	58.61 ± 0.30
adaCNN [33]	56.76 ± 0.50	61.05 ± 0.44	42.16 ± 0.43	54.12 ± 0.39
CovaMNet [31]	52.42 ± 0.76	63.76 ± 0.64	49.10 ± 0.76	63.04 ± 0.65
DN4 [26]	53.15 ± 0.84	81.90 ± 0.60	45.73 ± 0.76	61.51 ± 0.85
LRPABN [14]	63.63 ± 0.77	76.06 ± 0.58	45.72 ± 0.75	60.94 ± 0.66
MattML [56]	66.29 ± 0.56	80.34 ± 0.30	54.84 ± 0.53	71.34 ± 0.38
Ours	68.42 ± 0.92	82.42 ± 0.61	57.03 ± 0.86	73.00 ± 0.66

Table 2: Few-shot classification accuracy on the CUB [49] and Stanford Dogs [15] dataset. All experiments are from 5-way classification with the same backbone network (Conv4). The best performance is indicated in bold.

6. Additional Analyses

In this section, we provide additional experiments to clarify different aspects of our methods.

6.1. Analysis on the generated intra-class variations

We conduct a simple experiment to verify the effectiveness of our sampling method (Sec.4.3). Instead of sampling from the instance-conditioned mean and variance, we sample the intra-class variance feature from the zero-mean and unit-variance distribution.

Fig. 3 summarizes the results of this experiment for 5-way 1-shot classification on the CUB and NAB dataset. As can be seen, intra-class variance features sampled from zero-mean and unit-variance do not improve the results (red lines). In contrast, our method of sampling from the instance-conditioned posterior distribution generates features that consistently improve classification performance as the number of augmented samples increases.

6.2. Comparison to other data augmentation based methods

We compare our method with two other data augmentation based FSL methods: MetaIRNet[45] and Δ -encoder[38]. MetaIRNet uses a pre-trained image generator

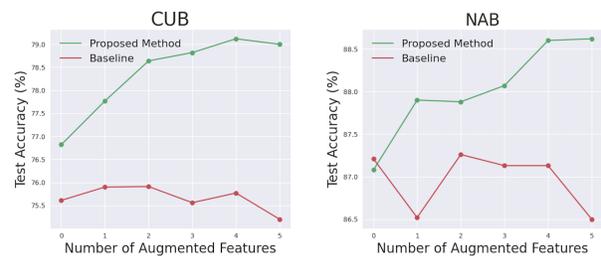


Figure 3: **Analysis on the generated intra-class variations.** We augment samples with the intra-class variance features sampled from the estimated mean and variance (green lines) or from the zero-mean and unit-variance (red lines). Our sampling scheme generates features that consistently improve classification.

to synthesize additional images and combine them with the original images to form additional training samples. The Δ -encoder learns to synthesize transferable non-linear deformations between pairs of examples of seen classes and apply these deformations to the few provided samples of novel categories.

We use the additional samples synthesized by both of these methods to train three types of classifiers: K-nearest

neighbors (KNN), Support Vector Machine (SVM), and Logistic Regression (LR), which are then used to classify novel images. The comparisons between these methods and our method are shown in Tab. 3. The superior performance of our method demonstrates that the augmented features obtained by our framework is beneficial for various types of classifiers. Note that for MetaIRNet [45], the results in Tab. 3 are lower than their numbers reported in the original paper since they pre-trained the backbone network on ImageNet while here all methods are trained from scratch.

Method	KNN		SVM		LR	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MetaIRNet [45]	63.18	74.82	63.76	76.77	63.53	79.95
Δ -Encoder [38]	67.31	82.67	76.02	82.87	76.22	85.17
Ours	75.46	83.17	79.07	87.59	78.34	89.30

Table 3: **Analysis of different classifiers.** Few-shot classification accuracy on the CUB [49] dataset in 1-shot and 5-shot settings with different types of classifiers.

In Tab. 4, we directly compare our method with the Δ -encoder using K-NN classifiers (K=1). Interestingly, it can be seen that the augmented features generated using the delta-encoder decrease classification performance. In fact, we observe that the majority (91.3%) of the nearest neighbors of the Δ -encoder’s generated features belong to different classes (some are visualized in Fig. 1), suggesting that the pairwise transformations extracted from this method might alter the class-identities of the transformed features. On the other hand, our generated features preserve well the class identity and mildly improve the classification results.

Method	Δ -Encoder		Ours	
	w/o Aug	w/ Aug	w/o Aug	w/ Aug
5-way	69.37	67.31	74.95	75.46
10-way	58.69	52.19	62.05	63.17
20-way	48.10	38.84	50.19	50.72

Table 4: **Effect of augmented features on 1NN classifier.** Few-shot classification accuracy on the CUB [49] dataset using 1NN classifier with original features vs augmented features. The original features of the Δ -Encoder are from a pre-trained ResNet18 network.

6.3. Comparisons to other methods to model intra-class variance

We assume that the intra-class variance can be modelled with an isotropic multivariate Gaussian distribution in a latent space. In this section, we compare this method with other methods that model the intra-class variance including Gaussian mixture variational autoencoder (GMVAE) [8], covariance matrix [52], and a baseline model where we

do not disentangle intra-class variance features from class-discriminative features.

Tab. 5 summarizes the results. The first row shows the results for GMVAE. This method enforces that the latent space is divided into distinct clusters for different classes. However, for this model, the accuracy drops by 6.15% and 2.13% for the 1-shot setting and 5.02% and 0.50% for the 5-shot setting on the CUB and NAB datasets respectively. The results align with our assumption that the intra-class variance is shared across different classes. Thus, enforcing a multi-modal prior distribution would lead to performance degradation.

The second row shows the results for the method proposed in [52] based on covariance matrices. Specifically, this method assumes a Gaussian prior on the distribution of intra-class variability across different classes which can be transferred from the base classes to the rare classes. However, instead of modelling the distribution by variational inference, [52] uses a covariance matrix to estimate the feature variance distribution. Here we apply this method on our extracted features to generate additional features on the CUB and NAB datasets under both 1-shot and 5-shot settings. Compared with the non-parametric estimate of the Gaussian distribution, modelling intra-class variance via variational inference in an end-to-end manner brings 6.03% and 1.91% improvement for the 1-shot setting and 3.64% and 1.03% improvement for the 5-shot setting on the CUB and NAB dataset respectively. Last, we provide the results for our method without feature disentanglement, denoted as “No disentanglement” in the third row. In spirit, this model is similar to [54] which models each point as a distribution via variational inference. Given a new sample, we augment it via sampling repeatedly from the estimated mean and variance. Without feature disentanglement and explicit modelling of the intra-class variance, this model does not achieve comparable results compared to other methods.

Intra-class distribution model	CUB		NAB	
	1-shot	5-shot	1-shot	5-shot
Gaussian Mixture Model [8]	75.16	86.46	86.49	94.72
Covariance Matrix [52]	75.28	87.84	84.71	94.19
No disentanglement [54]	73.40	86.60	81.83	92.83
Isotropic Gaussian (Proposed)	79.12	91.48	88.62	95.22

Table 5: Few-shot classification accuracy on the CUB [49] and NAB [46] dataset in 1-shot and 5-shot setting with different methods to model intra-class variance.

6.4. Data Distribution Analysis

We compare the data distributions between the real data and the generated data from our method in comparison to other state-of-the-art data generation methods [52, 38]. Here we measure the average intra-class variance, the distances between classes (inter-class distances), and the

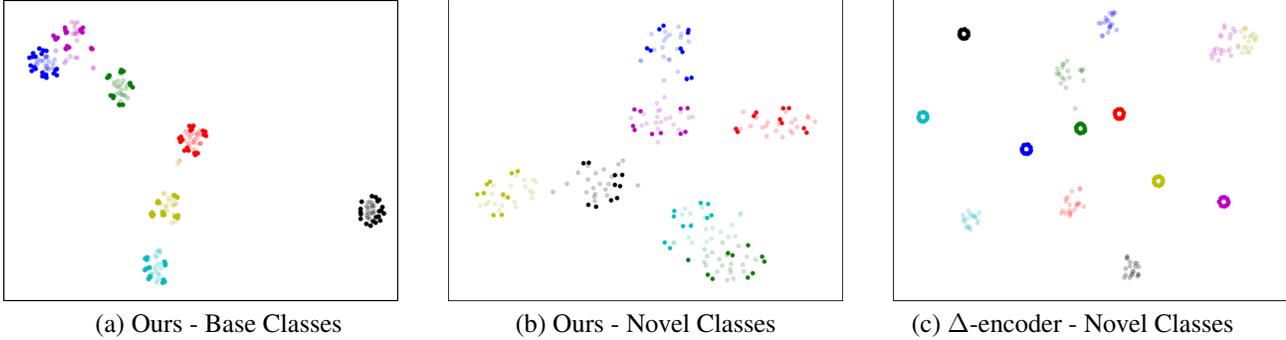


Figure 4: Visualization of the original features (marked as dark points) and the augmented features (marked as transparent points) on the CUB dataset using t-SNE. (a) Base real features and features augmented by our method. (b) Novel real features and features augmented by our method. (c) Novel real features and features augmented by the Δ -encoder. Our augmented features better mimic the distribution of real features.

data clusterability via the Davies–Bouldin index (DBI) [7]. Specifically, the DBI for a cluster i is calculated by:

$$DBI_i = \max_{i \neq j} \frac{Intra_{(i)} + Intra_{(j)}}{Inter_{(i,j)}} \quad (13)$$

where $Intra_{(i)}$ is the intra-class variance of cluster i , calculated by taking the average of squared deviations from the class center. $Inter_{(i,j)}$ is the distance between the two class centers of clusters i and j . The lower the value of the DBI, the better the separation between the clusters and the “tightness” inside the clusters.

Tab. 6 shows the average values of the intra-class variance, inter-class distances, and the DBI (denoted as D_{intra} , D_{inter} , and DBI respectively) across all novel classes of the CUB dataset. The inter-class distances are averaged across all pairs of classes. As shown in the table, features from the support set exhibit smaller intra-class variance compared to features from all data. All methods augment features from the support set. Interestingly, both sets of generated features using the method proposed in [52] and the Δ -encoder [38] decrease intra-class variance. On the other hand, the set of features augmented by our method closely approximate the data distribution of the set of all real features.

Fig. 4 demonstrates how real samples and generated samples from our method are distributed in a 2D space in comparison with the Δ -encoder [38] using t-SNE [32]. The original features are marked as light colors while the augmented features are marked as dark colors. Different colors denote different classes. The visualization for the base classes with the augmented features from our method is shown in Fig. 4a. Fig. 4b visualizes the real features and the generated features of our method for the novel classes. Our method generates samples that follow closely the real samples. The visualization for the novel classes and the generated features from the Δ -encoder is shown in Fig. 4c.

	D_{intra}	D_{inter}	DBI
Support data (5 samples)	21.52	32.77	2.21
All data	28.97	35.89	3.02
Covariance matrix [52]	17.98	35.24	1.79
Encoder-based Model [38]	10.34	11.69	1.77
Ours	27.27	34.12	2.53

Table 6: **Data Distribution analysis for different sets of features.** We augment features using our method and other data generation method based on covariance matrices [52] or the Δ -encoder [38]. All methods augment features from the support set (first row).

As can be seen, the generated data from each novel class forms into a new cluster and does not lie close to the actual data points.

7. Conclusion

We have proposed a simple, yet effective, feature augmentation method via feature disentanglement and variational inference to address the data scarcity problem in few-shot fine-grained classification. The generated features enlarge the intra-class variance for novel set images while preserving the class-discriminative features. The consistent performance improvement with the increase of the number of augmented samples suggests that the learned features are meaningful and nontrivial. The higher accuracy compared with other data augmentation based methods further demonstrate the superiority of our method. While this work mainly focuses on few-shot recognition problems, a promising future direction is to apply the feature transfer idea to other data-scarce or label-scarce tasks.

Acknowledgements. This work was partially supported by Zebra Technologies, the Partner University Fund, the SUNY2020 ITSC, and a gift from Adobe.

References

- [1] Achille Alessandro and Soatto Stefano. Emergence of invariance and disentanglement in deep representations. In *J. Mach. Learn. Res.*, 2018. 4
- [2] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. In *arXiv preprint arXiv:1711.04340*, 2018. 1, 2
- [3] Luca Bertinetto, João F. Henriques, P. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. *ArXiv*, abs/1805.08136, 2019. 5
- [4] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β -vae. In *arXiv: Machine Learning*, 2018. 4
- [5] Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *arXiv preprint arXiv:1802.04942*, 2018. 4
- [6] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Machine Learning(ICML)*, 2019. 5, 6
- [7] David L. Davies and Donald W. Bouldin. A cluster separation measure. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1979. 8
- [8] Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C.H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. In *arXiv preprint arXiv:1611.02648*, 2017. 7
- [9] Chanho Eom and Bumsub Ham. Learning disentangled representation for robust person re-identification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2
- [10] 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, 6
- [11] Hang Gao, Zheng Shou, Alireza Zareian, Hanwang Zhang, and Shih-Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 1
- [12] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *IEEE International Conference on Computer Vision (ICCV)*, June 2017. 1, 2
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 5
- [14] Huaxi Huang, Junjie Zhang, Jian Zhang, Jingsong Xu, and Qiang Wu. Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification. In *arXiv preprint arXiv:1908.01313*, 2019. 6
- [15] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization(FGVC), IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2011. 1, 2, 5, 6
- [16] 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. 2
- [17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 5
- [18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. 2
- [19] Hieu Le and Dimitris Samaras. Shadow removal via shadow image decomposition. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 1
- [20] Hieu Le and Dimitris Samaras. From shadow segmentation to shadow removal. In *European Conference on Computer Vision(ECCV)*, 2020. 1
- [21] Hieu Le and Dimitris Samaras. Physics-based shadow image decomposition for shadow removal. In *arXiv preprint arXiv:2012.13018*, 2020. 1
- [22] 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. 1
- [23] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *European Conference on Computer Vision(ECCV)*, 2018. 2
- [24] 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, 6
- [25] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. Adversarial feature hallucination networks for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1
- [26] Wenbin Li, Lei Wan, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6
- [27] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-grad: Learning to learn quickly for few-shot learning. In *arXiv preprint arXiv:1707.09835*, 2017. 2
- [28] Yann Lifchitz, Yannis Avrithis, Sylvaine Picard, and Andrei Bursuc. Dense classification and implanting for few-shot learning. pages 9250–9259, 2019. 2
- [29] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In *European Conference on Computer Vision(ECCV)*, 2018. 2, 4
- [30] Alexander H. Liu, Yen-Cheng Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. A unified feature disentangler for multi-domain image translation and manipulation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2
- [31] Wei Luo, Xitong Yang, Xianjie Mo, Yuheng Lu, Larry S Davis, Jun Li, Jian Yang, and Ser-Nam Lim. Cross-x learn-

- ing for fine-grained visual categorization. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 6
- [32] L. V. D. Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. In *Journal of Machine Learning Research*, 2008. 8
- [33] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning (ICML)*, 2018. 6
- [34] Aravind Rajeswaran, Chelsea Finn, S. Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *NeurIPS*, 2019. 2
- [35] 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
- [36] Adam Santoro, Sergey Bartunov, M. Botvinick, Daan Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 2
- [37] 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
- [38] 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. 1, 2, 5, 6, 7, 8
- [39] 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
- [40] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1
- [41] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2, 6
- [42] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6
- [43] 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, 6
- [44] 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
- [45] Satoshi Tsutsui, Yanwei Fu, and David Crandall. Meta-Reinforced Synthetic Data for One-Shot Fine-Grained Visual Recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, June 2019. 1, 5, 6, 7
- [46] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 2, 5, 6, 7
- [47] 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, 6
- [48] 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
- [49] 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, 2, 5, 6, 7
- [50] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. volume abs/2101.06395, 2021. 2
- [51] 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
- [52] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for deep face recognition with under-represented data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 4, 7, 8
- [53] 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
- [54] 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, 4, 7
- [55] Ruixiang ZHANG, Tong Che, Zoubin Ghahramani, and Yangqiu Song Yoshua Bengio and. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 1
- [56] Yaohui Zhu, Chenlong Liu, and Shuqiang Jiang. Multi-attention meta learning for few-shot fine-grained image recognition. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, 2020. 5, 6
- [57] Yang Zou, Xiaodong Yang, Zhiding Yu, B. V. Kumar, and J. Kautz. Joint disentangling and adaptation for cross-domain person re-identification. In *European Conference on Computer Vision (ECCV)*, 2020. 2