Meta Variance Transfer: Learning to Augment from The Others

Seong-Jin Park [1] **Seungju Han** [1] **Jiwon Baek** [1] **Insoo Kim** [1] **Juhwan Song** [1] **Hae Beom Lee** [2]
Jae-Joon Han [1] **Sung Ju Hwang** [2]

Abstract

Humans have the ability to robustly recognize objects with various factors of variations such as nonrigid transformations, background noises, and changes in lighting conditions. However, training deep learning models generally require huge amount of data instances under diverse variations, to ensure its robustness. To alleviate the need of collecting large amount of data and better learn to generalize with scarce data instances, we propose a novel meta-learning method which learns to transfer factors of variations from one class to another, such that it can improve the classification performance on unseen examples. Transferred variations generate virtual samples that augment the feature space of the target class during training, simulating upcoming query samples with similar variations. By sharing the factors of variations across different classes, the model becomes more robust to variations in the unseen examples and tasks using small number of examples per class. We validate our model on multiple benchmark datasets for few-shot classification and face recognition, on which our model significantly improves the performance of the base model, outperforming relevant baselines.

1. Introduction

Humans can robustly recognize and understand a concept under various circumstances. For instance, we can learn a new visual concept only by observing a few images. Humans can also accurately recognize the same concept with large variations in shapes, lighting conditions, or with background changes. On the contrary, current deep learning frameworks generally require huge amount of data under diverse variations to robustly learn and recognize new concepts. However, collecting a complete dataset with all possible variations described for each instance is not feasible for real-world problems. This is particularly the case when dealing with large number of classes. In real-world settings, it is more common to collect an imbalanced dataset with a large discrepancy in the number of examples, or the coverage of the observable variations for each class. A deep neural network trained on such a dataset may not generalize well to unseen variations at test time.

A popular solution to this problem is perturbing existing samples during training, such that the perturbed examples could account for unseen variations. It is well known that conventional data augmentation methods such as random cropping, flipping, and color jittering of images are helpful for almost all computer vision applications. However, less studies have focused on applying semantic perturbations to training samples to simulate unseen variations in test data, particularly variations which cannot be described by conventional data augmentation and pre-defined transformations.

Some existing works (Goodfellow et al., 2014; Zhu et al., 2017; Karras et al., 2019) have addressed this problem by training generative models on a dataset with pre-defined modes of variations. Such methods generate images of a class with predefined variations. While this approach could generate realistic and diverse samples, there are still limitations. First, training such models requires an additional dataset with annotations for the required pre-defined variations. Since obtaining human annotations on the variations is costly, we can only obtain pre-defined types of variations, which may not cover all possible variations of a single class. Moreover, the images generated by the model do not guarantee improvements of the test accuracy, since it is trained separately from the classification model.

However, such a way of generating data with annotation to facilitate learning differs from the way of which humans learn. In fact, we presume that humans have a *meta-knowledge* that can transfer variations from one visual class to another, thus can imagine an unseen variation of a class. For example, even when we see an animal that we have never seen before, we can imagine how it will look with different poses in different surroundings. This is presumably due to the fact that there are certain types of variations

[1]Samsung Advanced Institute of Technology [2]KAIST. Correspondence to: Sung Ju Hwang <sjhwang82@kaist.ac.kr>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

that can be shared and transferred across different visual concept classes. However, a variation could not always be transferable from one class to another; for example, a mode that describes a bird flying will not transfer to a dog.

Motivated by this intuition, we propose a novel augmentation method which transfers observed variations from one class to another class, such that it could generate virtual samples for the target class to augment its feature space (Fig. 1). The proposed method alleviates the need of collecting huge datasets with an exhaustive set of possible variations. However, identifying meaningful directions of variations that can actually yield improvements on the test examples is a difficult problem, since the variations are different for each class and may not be compatible across classes. To resolve this issue, we propose a *meta-learning* framework that learns to transfer the variance of one class to another. It learns how and what should be transferred from the observed variations of a source class to the target class. Consequently, new samples are generated that simulate the upcoming queries of novel samples. The model learned with the generated samples improves the accuracy on test samples with variations unseen from original training data. We additionally perform a simple manifold regularization to facilitate knowledge transfer of dominant variations that can be easily shared across different categories. The key assumption in the success of this model is that, although observed variations for a class might be limited to fully represent the class, there could be numerous other classes that can account for its unseen variations, to help represent its distribution.

Our method differs from existing augmentation methods in that it learns both class representation and the meta-learner to transfer the variations *simultaneously*, such that it improves the test accuracy. Also, we transfer the information from the observed sample variance without attribute annotations through the learned parameters to select the meaningful variations. To confirm our assumption, we first perform a proof-of-concept experiment, and further validate the proposed method on multiple benchmark datasets for few-shot classification. The results show that our *Meta Variance Transfer* (MVT) significantly improves the performance of the base model.

Our contributions can be summarized as follows:

- We propose a novel meta-learning framework that transfers the observed variance of one class to another class such that it simulates upcoming queries and improves the test accuracy.
- We introduce an additional meta-learner that learns meaningful variance from one class to transfer it to another class without additional annotation. Accordingly, the feature space is augmented to better represent the data distribution.

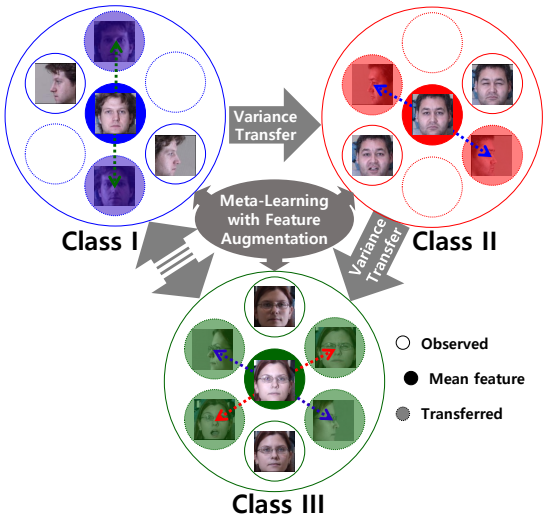

Figure 1. The key idea of the proposed Meta Variance Transfer. The observed variations are transferred from one class to other classes in the feature space. The feature space of each class is enclosed within a solid line circle. An imbalance in the dataset can be seen, as each class contains different variations. Class I, II, and III are mainly composed of features of exclusive variations in pose, expression, and illumination, respectively. The proposed method conceptually transfers the pose variance information from Class I to class II. Class II transfers expression variance information to Class III. In retrospect, each class generates the received variance-based feature around the mean feature value of the class itself.

- We additionally perform a simple manifold regularization to facilitate transferring dominant directions of variations through a linear auto-encoder with learnable parameters.
- We perform a proof of concept on a simplified problem on face recognition and validate the method on multiple benchmark datasets for few-shot classification that shows our model significantly improves the baseline performance.

2. Related work

Meta learning methods. The two most popular approaches for few-shot meta-learning are: metric-based and initialization-based methods. Metric-based meta learning methods (Koch, 2015; Vinyals et al., 2016; Snell et al., 2017; Oreshkin et al., 2018; Mishra et al., 2017) address the few-shot classification problem by learning a shared metric space. On the learned metric space, distance or similarity is used to determine the class label. Matching networks (Vinyals et al., 2016) employed cosine similarity to measure the similarity between the support and the query examples, while Prototypical networks (Snell et al., 2017) used Euclidean

distance. The other popular approach is based on the model initialization for fast adaptation. The methods (Finn et al., 2017; 2018; Li et al., 2017; Lee & Choi, 2018; Zintgraf et al., 2019) in this approach try to put a model in a proper initial point so that the model can rapidly adapt to new tasks with a few gradient descent updates. There are also several methods (Ravi & Larochelle, 2016; Munkhdalai & Yu, 2017) that learn an optimizer. Those replace the stochastic gradient decent optimizer or weight-update mechanism. In this paper, we adopt the metric learning approach (Vinyals et al., 2016) with a modification for our experiments.

Generative adversarial network. Generative adversarial networks (GANs) (Goodfellow et al., 2014) are generative models that are trained to generate realistic samples from a given dataset, via a minimax game between a generator and a discriminator. It is also relevant to our work since samples generated by GANs can be also used for data augmentation. There has been dramatic advance of GANs in image generation (Radford et al., 2015; Karras et al., 2017; 2019) which led to their successful application to various real-world problems (Zhu et al., 2017; Ledig et al., 2017; Pascual et al., 2017; Luc et al., 2016).

Some recent works (Zhang et al., 2018; Gao et al., 2018; Shmelkov et al., 2018; Shrivastava et al., 2017; Antoniou et al., 2017) have further investigated ways to use generated images from GANs to improve on the image classification performance of a target model. Especially, DAGAN (Antoniou et al., 2017) augments a classifier with fake images generated from a separately-trained generative adversarial network. The DAGAN performs transfer from *seen* classes to *unseen* classes, while our method performs transfer between *unseen* classes. In the generative model-based methods, sample generation and learning of the classifier is decoupled. On the other hand, ours is meta-learned to explicitly lower the classification loss on unseen classes.

Hallucination-based methods. The most relevant approach to the proposed method is the hallucination-based methods (Miller et al., 2000; Hariharan & Girshick, 2017; Wang et al., 2018; Schwartz et al., 2018; Chen et al., 2019c;b; Tsutsui et al., 2019). Early work by (Miller et al., 2000) used a density over transforms as a prior knowledge for the general deformations of different classes. Beyond the character recognition, (Hariharan & Girshick, 2017) proposed a method for generic categories to transfer modes of variation from base classes to novel ones. (Dixit et al., 2017) leveraged an additional dataset of images with hand-labeled attributes. However, those works require heuristic steps or human-labeled annotations. (Wang et al., 2018) proposed an end-to-end method in a meta-learning framework. A meta-learner takes both a random noise and an instance as input and then learns to generate new images.

There are several works (Chen et al., 2019c;b; Tsutsui et al., 2019) that learn to deform images by fusing two real images. (Chen et al., 2019c;b) deforms two real images to augment a support set. They additionally use samples from base classes during test phase. (Tsutsui et al., 2019) uses pre-trained image generator to generate slightly different images from the original image and fuse them for augmentation. In contrast, our method transfers variations in the feature domain.

Similarly to ours, there are some methods (Schwartz et al., 2018; Gao et al., 2018) that directly synthesizing new instances in the feature domain. (Schwartz et al., 2018) trains an encoder that learns transferable intra-class deformations between pairs of the same base class and synthesizes novel samples by applying the deformations to the samples of novel class during test phase. This method requires deformation for training samples from the classes seen during training and the generator is trained explicitly by the reconstruction loss. (Gao et al., 2018) trains an adversarial augmentation network that transfers the covariance of base class that is similar to the novel class. Ours differs from those in that we do not refer the information in base classes during test phase. Recently (Lee et al., 2020) proposed a meta-learning method that perturbs a sample to simulate the query data by adding noises to activations. The perturbation depends on the data itself, while the proposed method refers the actual observed variations of other classes.

3. Method

In this section, we first define the problem setting and the notations used throughout the paper. Then, we describe the proposed meta variance transfer method and its procedures in detail.

3.1. Few-shot learning problem

Here we define our few-shot learning problem setting for meta-learning to augment from the observed data. The goal of meta-learning is to learn a model that generalizes well over a task distribution $p(\mathcal{T})$, such that it can obtain a good performance on an unseen task from the same distribution. To this end, we train the model over larger number of episodes, where the model trains on the task $\mathcal{T}_\tau \sim p(\mathcal{T})$ at each episode τ. For few-shot classification, each episode is a w-way k-shot classification task, where the model requires to classify the given sample into one of the w classes with only k training instances per class. The detailed problem setting and the notations are as follows. For w-way k-shot learning, we have a training dataset $\mathcal{D} = \{(x_i, y_i)\}$, where x_i is the data instance and y_i is its corresponding label. Given the dataset, we first randomly select w classes and map them to one of the w labels $l \in \{1, 2, ..., w\}$ for w-way classification. Then, we sample k support and m query samples from each class. The training set for each

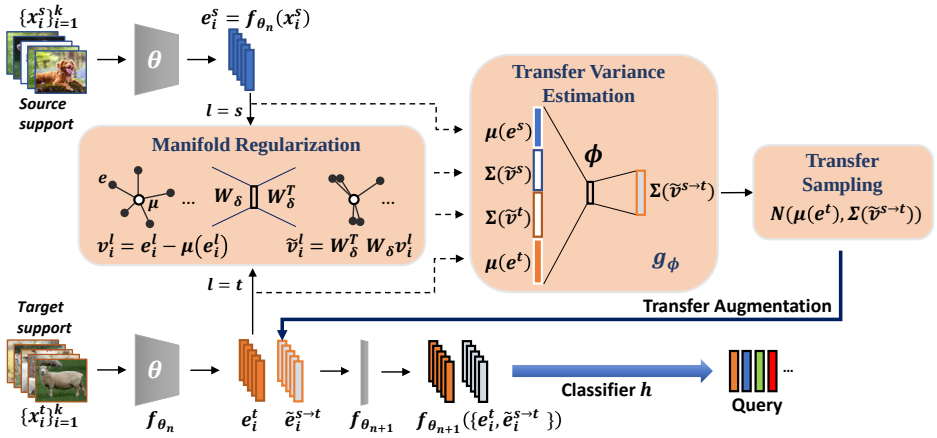

Figure 2. The overall framework of our Meta Variance Transfer from a source category (s) to a target category (t). This framework includes Manifold Regularization, Transfer Variance Estimation, Transfer Sampling, and Augmentation for the basic w-way / k-shot learning problem. $\tilde{\Sigma}^t$ denotes the transferred variance from the source s to the target t.

episode τ consists of the support set $\mathcal{S} = \{S_l\}_{l=1}^{w}$, where $S_l = \{(x_i, l)\}_{i=1}^{k}$ and the query set $\mathcal{Q} = \{Q_l\}_{l=1}^{w}$, where $Q_l = \{(x_i, l)\}_{i=1}^{m}$. The goal of few shot learning here is to learn from the support set $\mathcal{S}$ to correctly classify the queries $\mathcal{Q}$ to one of the classes in the support set. What makes the problem difficult is that the number of support samples is small. Thus, those samples in each class usually contain a limited set of variations for the given class, which makes it hard to learn the model that could generalize well to the queries with unseen variations. In the following subsections, we describe a method which alleviates this problem by transferring the variations observed in each class across classes to account for missing variations for each class.

3.2. Learning to augment from the others

The key idea of our method is that the observed variations of real data (e.g. geometric deformation, background changes, simple noise) could hint on *unseen* variations in other classes. Based on this idea, we introduce a meta learner that learns how and what to transfer the modes of variations between classes to improve the test accuracy, while simultaneously learning the representation space.

We consider the variations in the latent feature space which captures semantic information about the given sample. Formally, we denote the semantic variations $v_i^l = e_i^l - \mu(e^l)$ as the displacement between an i-th embedding vector e_i^l and the mean vector μ^l of class l in the feature space. Then the proposed model learns to transfer the variance in the feature space by *selecting* the variations that could be helpful in simulating the unseen test examples for the target class. The overall loss function consists of two terms:

$$\mathcal{L}_{\theta,\phi,\delta} = \mathcal{L}_{class} + \lambda \mathcal{L}_{manifold}, \tag{1}$$

where θ, ϕ, and δ represent the parameters for representation learning, variance transfer, and regularization (sec. 3.4), respectively. Each term and specific procedures are explained in the following subsections.

3.3. Meta Variance Transfer

The overall procedures of our MVT are as follows. Given the data in each episode τ, we first compute the embedding vector $e = f_\theta(x)$, where $f_\theta(\cdot)$ is a backbone network with the parameter θ, then compute the sample mean and covariances of the embedding vectors for each class. Then, we randomly sample a source and target class pair and feed the meta-learner $g_\phi(\cdot)$ the concatenated sample statistics of the source class (μ^s, Σ^s) and the target class (μ^t, Σ^t). The learner then learns from the statistics to output the transferred variation $\tilde{\Sigma}^t = g_\phi(cat[\mu^s, \Sigma^s, \mu^t, \Sigma^t])$ for the target class. We sample from a Gaussian distribution with the target mean and transferred variation $N(\mu^t, \tilde{\Sigma}^t)$ to augment the virtual support embedding vectors $\tilde{e}^t$ to the real support embedding vector set. Finally, we build a classifier $h(\cdot)$ using both real and virtual support vectors through any existing methods such as MatchingNet (Vinyals et al., 2016) and compute the cross entropy (CE) classification loss by

$$\mathcal{L}_{class} = \sum_{\mathcal{Q}} CE(h(f_\theta(x), \mathcal{S}; \theta, \phi, \delta), y). \tag{2}$$

For sampling, we use reparameterization trick to compute the gradients by following (Kingma & Welling, 2013). Although our variance transfer can be applied to any layer with a vector representation, as it is depicted in Figure 2, we transfer the embedding features before the final layer for our experiments unless otherwise noted. This is because while MVT aims to transfer semantic variations in the feature domains, the goal of final layer is to aggregate all the observed support samples to a point for each class such that the final feature eliminates the observed variations. For ex-

ample, reconstructing the original image from the final layer for face recognition would only yield frontal faces. On the other hand, features below the final layer capture sufficient amount of semantic variations for the variance transfer to be meaningful. We perform variance transfer not only at the meta-training but also at meta-test time, to augment the data for unseen classes. Further details of the method are described in section 3.5.

3.4. Manifold regularization

The second term, $\mathcal{L}_{manifold}$ regularizes the manifold of the features and the mode of variations. Without any regularization, the variations can be scattered arbitrarily on the feature space, which may make it difficult to learn and transfer the meaningful variations to the other classes. To better facilitate the variation transfer, we regularize and learn the directions of variation to be transferred. We simply learn a linear auto-encoder with the parameter δ by minimizing the following reconstruction loss:

$$\mathcal{L}_{manifold} = \sum_{\mathcal{S}} |v_i^l - \hat{v}_i^l| = \sum_{\mathcal{S}} |v_i^l - W_\delta^T W_\delta v_i^l|, \quad (3)$$

where $v_i^l = e_i^l - \mu(e^l)$ represents the feature displacement. W_δ is a matrix form of the FC layer having the parameter δ. Note that we omit the summation along feature dimension for simplicity. Instead of v, we then transfer the projected variation $\hat{v} = W_\delta^T W_\delta v$. This projection will suppress the variations that are difficult to be shared or less meaningful for the target class. Then, we replace $\Sigma(v)$ with $\Sigma(\hat{v})$ to compute g_ϕ in section 3.3. The overall framework of our MVT is depicted in detail in Figure 2.

3.5. Implementation details

Here we describe the implementation details of our method. As for the meta-learner ϕ which performs the variance transfer, we use an MLP with leakyReLU activation functions. Specifically, ϕ contains the parameter for the FC layer with the number of channels less than those of the feature vector, leakyReLU with the slope of 0.01, and another FC layer with the same number of channels with those of the original feature. For the sample covariance matrix, we assume a diagonal matrix to reduce the model complexity and simply compute the variance of each feature, from which the dimension for variance is reduced to the number of embedding feature dimension. Even though it may seem like a strong assumption, transferred variance can still properly represent semantic variations by combining different feature variances. We block the back-propagation through the g_ϕ to the embedding feature such that the transfer module can focus on learning to transfer given observed embedding features. In the PyTorch implementation, we *detach* the embedding features before feeding them to g_ϕ.

Algorithm 1 Meta Variance Transfer

Input: Training set $\mathcal{D} = \{(x_i, y_i)\}$
for each episode τ **do**
  Random sample $\mathcal{S} = \{S_l\}_{l=1}^w$ and $\mathcal{Q} = \{Q_l\}_{l=1}^w$
  Compute embedding features e_i^l for every sample i
  Compute μ^l and the feature displacement $v_i^l = e_i^l - \mu^l$
  Compute the manifold loss
  $L_{manifold} = \sum_{\mathcal{S}} |v_i^l - \hat{v}_i^l| = \sum_{\mathcal{S}} |v_i^l - W_\delta^T W_\delta v_i^l|$
  Compute $\Sigma^l(\hat{v})$
  for $s = 1$ **to** w **do**
    Sample target class $t \neq s$ randomly
    Compute $\tilde{\Sigma}^t = g_\phi([\mu^s, \Sigma^s(\hat{v}_i), \mu^t, \Sigma^t(\hat{v}_i)])$
    Sample z transferred samples $\tilde{e^t}$ from $N(\mu^t, \tilde{\Sigma}^t)$
    Augment $\tilde{e^t}$ to the embedding feature set of S_t
    Compute the final feature vectors f^t and $\tilde{f}^t$
  end for
  Compute the classification loss L_{class} in Eq. 5
  Optimize $\mathcal{L}_{\theta,\phi,\delta} = \mathcal{L}_{class} + \lambda \mathcal{L}_{manifold}$
end for

Our method is applicable to different types of classifiers h. In our experiments, we use a simpler version of MatchingNet (Vinyals et al., 2016) such that we compute the softmax over the Euclidean distance between the query and each support samples. Specifically, we first compute the average Euclidean distance d between a query q and the *augmented support set* for every classes. Then we compute the softmax over the negative of the distances to compute the cross entropy loss:

$$d_l(q) = \frac{1}{(k+z)} \{ \sum_{i=1}^{k} (||f(q) - f_i^l||) + \sum_{j=1}^{z} (||f(q) - \tilde{f}_j^l||) \}, \quad (4)$$

$$\mathcal{L}_{class} = \sum_{\mathcal{Q}} CE\left(\frac{exp(-d_y(q))}{\sum_l exp(-d_l(q))}\right), \quad (5)$$

where y in d_y represents the target label for the query and z is the number of augmented samples. Here we denote the final representations of the original and augmented samples by f^l and $\tilde{f}^l$, respectively. For the hyper-parameter λ in equation 1, we used 0.1 throughout the experiments. We found that our method is robust to changes in the λ value, if its scale is within certain ranges (from 0.01 to 1). The detailed procedures are summarized in Algorithm 1.

4. Experiments

Datasets. We extensively validate our method on diverse few-shot classification datasets. For fine-grained few-shot classification, we use CUB dataset (Wah et al., 2011) and face recognition datasets: VGGface2 (Cao et al., 2018) and CASIA-webface (Yi et al., 2014) datasets. We also use

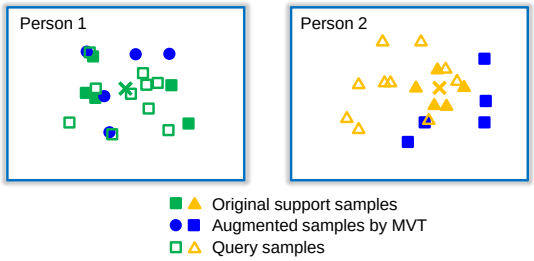

Figure 3. t-SNE visualization of the features trained with the proposed MVT of 2 subjects from the Multi-PIE dataset. Note that a few samples among the augmented samples well simulate the upcoming query samples.

miniImageNet dataset (Deng et al., 2009; Vinyals et al., 2016; Ravi & Larochelle, 2016) which is widely used as a few-shot meta-learning benchmark.

The CUB dataset consists of 200 bird classes with 11,788 images. Following (Chen et al., 2019a), we use 100, 50, and 50 classes for base, validation, and novel classes, respectively. The VGGface2 dataset originally contains around 3.3 million images of 9,131 subjects and the CASIA-webface dataset consists of 494,414 images with 10,575 subjects. We randomly select 300 subjects for each dataset and split them to 200, 50, and 50 subjects for seen, validation, and unseen datasets, respectively. For CASIA-webface we only sampled subjects with more than 50 images because the dataset contains subjects with less than 15 images. We call those two datasets as mini-VGG and mini-CASIA datasets. Finally, the miniImageNet contains 100 classes with 600 images for each class. For a fair comparison, we use the same split of (Ravi & Larochelle, 2016).

Training and evaluation protocols. To train our model, we follow the settings of (Chen et al., 2019a). We use 84×84 images for the Conv4 network, and 224×224 images for ResNet-based networks, as done in many previous works. Moreover, similarly to existing works, we apply simple data augmentations such as random crop and flip. We train the model by using the Adam optimizer (Kingma & Ba, 2014) for about 110k iterations with the initial learning rate of 0.001, and decay it by 0.1 at 70k and 90k iterations. We set the number of queries during both meta-training and meta-testing as 15 following the convention, and report 5-way results unless otherwise mentioned. We select the model that obtains the best accuracy on the validation dataset during meta-training and use it for the meta-test. We evaluate the model for 600 test episodes and report the mean accuracy as well as the 95% confidence intervals. We used a single NVIDIA P40 GPU for training the model. For the Conv4 network, we mainly use the Conv4 with an additional FC layer *(Conv4$_+$)* to aggregate the variations as we mentioned in section 3.3 and report the results of the original Conv4 as well. Right before the FC layer, we reduce the spatial dimension of the feature map with a 3×3 pooling with the stride of 2. We set the number of augmented samples z to the same number of real support samples k.

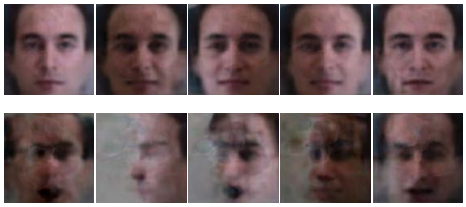

Figure 4. Decoded images of the original support features (Top) and those of the transferred and augmented features (Bottom).

4.1. Proof of concept

We first perform a proof-of-concept experiment to provide evidence for the validity of the proposed meta-learning framework for variance transfer. To this end, we constructed a synthetic dataset to clearly evaluate whether the proposed method works in the way we intend to. At each episode, we explicitly sample half of the classes with limited number of predefined types of variations, while we sample the other classes randomly from the whole data distribution. By this experiment, we can evaluate whether the variations in the classes sampled in half are indeed transferred to the classes with limited variations. For this proof-of-concept experiments, we adopt the multi-PIE face dataset (Gross et al., 2010), which contains images of subjects with 15 different viewpoints and 19 illumination conditions, with diverse facial expressions (e.g. smiling). We preprocess the dataset by cropping out only the face regions using a face detector (Zhang et al., 2016). We sample the half of k-way classes from the frontal images only with illumination and subtle facial expression changes.

We use two visualization methods to verify whether the proposed meta-learner is able to transfer the variations across classes as intended. First, we visualize the feature space by projecting the features onto a 2D space through t-SNE (Maaten & Hinton, 2008). Figure 3 shows this visualization. Secondly, we project the transferred features onto the image domain, by training an auxiliary decoder to reconstruct the original image from the feature embeddings augmented by our meta-variance transfer. Figure 4 shows the reconstruction of the augmented features.

Figure 3 shows the embeddings of the two subjects from the Multi-PIE dataset: The original support samples (filled markers), query samples (hollowed markers), and the additional support samples augmented by the proposed MVT (filled markers of different shape). From the figure, we can observe that most of the augmented samples overlap or are located close to the upcoming queries in the t-SNE 2D-projected space. This implies that the proposed MVT

Table 1. Few-shot classification accuracy on MiniImageNet and CUB dataset. $*$ denotes that the model is our reproduction of the original model, and $_+$ denotes that the model uses Conv4$_+$ network architecture as the backbone network, which has an additional FC layer to the base Conv4. We use our simple version of MatchingNet (*MATCHING** and *MATCHING*$^*_+$) as our baseline. We report the mean accuracy and the 95% confidence intervals for 600 test episodes.

METHOD	MINIIMAGENET 5 SHOT	MINIIMAGENET 10 SHOT	CUB 5 SHOT	CUB 10 SHOT
PROTONET (SNELL ET AL., 2017)	64.24±0.72	-	76.39±0.64	-
MAML (FINN ET AL., 2017)	64.55±0.52	-	75.75±0.76	-
META-SGD (LI ET AL., 2017)	65.55±0.56	-	-	-
REPTILE (NICHOL ET AL., 2018)	65.99±0.58	-	-	-
CAVIA (ZINTGRAF ET AL., 2019)	65.85±0.55	-	-	-
B++ (CHEN ET AL., 2019A)	66.43±0.63	-	79.34±0.61	-
MATCHING (VINYALS ET AL., 2016)	63.48±0.66	-	75.29±0.75	-
MATCHING*	64.63±0.70	68.90± 0.64	75.63±0.68	79.04± 0.60
MATCHING$^*_+$	64.88±0.68	68.85± 0.68	76.98±0.67	81.19± 0.59
MAML+META-DROP (LEE ET AL., 2020)	67.42±0.52	-	-	-
MAML+META-DROP*	67.06±0.74	70.72± 0.66	76.17±0.65	79.99± 0.59
MAML+META-DROP$^*_+$	67.24±0.68	70.72± 0.69	75.31±0.65	79.84± 0.57
MATCHING*+MVT (OURS)	65.24±0.68	71.81± 0.62	77.33±0.64	81.17± 0.58
MATCHING$^*_+$+MVT (OURS)	**67.67±0.70**	**71.83± 0.67**	**80.33±0.60**	**83.09± 0.51**

can simulate the test environment and reflect it during the meta-learning transfer stage. Even if the additional support samples augmented by the proposed MVT do not simulate the real-world data, it still can help obtain discriminative representations.

Figure 4 shows the examples of decoded images of the embeddings of a certain class (subject). Top row shows the decoded images of the original support features. The bottom row shows the decodings of the transferred and augmented features. The class originally contained a limited set of variations which only account for illumination changes. However, the transferred variations with MVT are decoded back to images with variations in the pose and facial expression. These results validate that our MVT can learn to transfer the semantic variations from one class to another. While the reconstruction is not perfect, note that generating actual image samples is not the focus of this paper. Rather, the main purpose of MVT is to transfer the variation information in the feature domain such that it can help improve classification accuracy without the overhead required to generate and train on additional image samples.

4.2. Few-shot classification evaluation

We now conduct a comparative study of our model against existing works in few-shot classification settings.

Table 1 reports the results of the proposed Meta-Variance Transfer compared to existing methods. As the proposed method requires to transfer the variances, we report the accuracy on 5-shot and 10-shot classification experiments rather than on one-shot classification. We used a simple MatchingNet (Matching*) as a base model (Sec. 3.5) due to its simplicity and compatibility to our method. We observe that our method outperforms all the baselines with the same backbone networks, including the original MatchingNet and the simplified version of MatchingNet (Matching*).

We also report the accuracy of the Matching network with an additional final FC layer (Matching$^*_+$), which obtains similar results on miniImageNet and achieves higher accuracy on CUB. We further compare against *Meta-dropout* (Lee et al., 2020), which is highly relevant to the proposed method since it also perturbs the training instances to help obtain improved performance on the meta-test case. The difference is that meta-dropout transfers from the tasks in meta-training to tasks in meta-test, while our MVT transfers across classes in the task given at meta-test time. We reproduce and report the results of Meta-drop under both the original setting and the same settings as ours. The results show that our MVT obtain the comparable or slightly better accuracy on the 5-shot miniImageNet classification, but outperforms it in all other cases. Even when considering that our baseline model (Matching$^*_+$) works better than MAML used in Meta-drop (MAML), the significant accuracy gap between the two approaches still validates the effectiveness of our method. For the 5-shot CUB, we also report the accuracy of the method (B++) proposed in (Chen et al., 2019c). Our MVT outperformed the other methods on both datasets. Note that the larger accuracy gap between the two of ours (Matching*+MVT and Matching $^*_+$+MVT) than those between the two baselines (Matching* and Matching $^*_+$) justifies the selection of our transfer layer mentioned in section 3.3 (i.e., the layer before the final layer). Our method also achieves more performance gain on CUB, since our variation transfer could be more helpful in fine-grained clas-

Table 2. Ablation study. Baseline is our simple MatchingNet. The *VAR* denotes the variation transfer by simply adding v_i of one class to the mean of another class without learning. The *MANIFOLD* represents the usage of manifold loss.

METHOD	MINIIMAGENET 5 SHOT	CUB 5 SHOT
BASELINE	64.88±0.68	76.98±0.67
BASELINE+VAR	32.76±0.58	29.09±0.49
BASELINE+MANIFOLD	64.37±0.70	76.74±0.65
BASELINE+MANIFOLD+VAR	31.59±0.57	32.21±0.56
META VARIANCE TRANSFER ($\lambda = 0$)	66.68±0.69	79.92±0.61
META VARIANCE TRANSFER (PROPOSED)	**67.67±0.70**	**80.33±0.60**

Table 3. Results of *MVT* with and without conventional augmentations. *AG.* denotes the random crop, flip, and color jittering augmentations.

DATASET	AG./MVT	RESNET-10$_+$ 5-SHOT	RESNET-10$_+$ 20-SHOT
CUB	×/×	70.28 ±0.81	74.05± 0.62
	×/√	**71.56±0.75**	**76.01± 0.65**
	√/×	84.01± 0.57	87.52± 0.45
	√/√	**85.35± 0.55**	**88.67± 0.42**
MINIVGG	×/×	90.23±0.47	92.89±0.38
	×/√	**91.22±0.39**	**93.24± 0.31**
	√/×	91.54± 0.43	93.78± 0.32
	√/√	**91.87± 0.41**	**94.85± 0.30**
MINICASIA	×/×	80.22±0.60	83.55±0.47
	×/√	**82.31±0.59**	**85.83± 0.45**
	√/×	87.25± 0.48	90.77± 0.37
	√/√	**88.90± 0.45**	**92.07± 0.34**

Table 4. Comparison with generative model-based augmentation.

METHOD	OMNIGLOT 20 WAY 2 SHOT	MINI-VGG 5 WAY 5 SHOT
MATCHING$^*_+$	96.86±0.18	85.95± 0.56
DAGAN	97.03±0.16	84.76± 0.57
MATCHING$^*_+$+MVT	**97.18±0.15**	**87.13± 0.52**

sification tasks where the classes share diverse semantic variations. However, we do not necessarily assume that the classes should be relevant because our method can flexibly control the amount of transfer between classes using the meta-learned transfer module, which explains why our method also works well on miniImageNet.

We further compare our MVT against DAGAN(Antoniou et al., 2017), which utilizes a generative model for data augmentation. The DAGAN first trains a generative adversarial network on the training data and trains a classifier with additional fake images generated from a separately-trained generator. We train a classifier with additionally generated images from DAGAN. To reduce the training time of the generator, we resized the images of miniVGG to 36×36. In

Table 5. Results with different backbones.

NETWORK	MVT	CUB 5 SHOT	CUB 10 SHOT
CONV4	×	76.98±0.67	81.19± 0.59
CONV4	√	**80.33±0.60**	**83.09± 0.51**
RESNET-10	×	84.01±0.57	85.67± 0.54
RESNET-10	√	**85.35±0.55**	**88.41± 0.44**
RESNET-34	×	85.24±0.55	88.71± 0.46
RESNET-34	√	**87.07±0.51**	**89.68± 0.41**

Table 4, we observe that DAGAN obtains no performance gain on VGGFace, which agrees with the results reported in the paper (Antoniou et al., 2017). This is because in DAGAN, the generator is separately trained from the classifier with the hope that it helps generalization on the target domains while MVT is meta-learned to transform and transfer only the meaningful variations that can lower the classification loss on unseen classes. Another obvious advantage of using MVT over DAGAN or any other image generation-based augmentation methods, is that MVT does not require a separate image generator which is costly to train and hold in memory, and can transfer variations in the latent feature space of the classifier in a single training phase.

4.3. Ablation study

In Table 2, we justify our design decisions through an ablation study. The baseline represents the simple MatchingNet. The *VAR* denotes the variation transfer by simply adding each variation v_i of one class to the mean vector of another class. The results of the second and fourth row show that such a simple variation transfer without learning causes the training to fail. This indicates that learning how and what to transfer is important. By comparing the first and third row, we can see that simply adding the manifold loss alone also does not improve the accuracy. The fifth row shows that the variance transfer with meta learning without the manifold regularization obtains improved accuracy, but still largely underperforms the full MVT model with manifold regularization, which achieves the best accuracy.

Table 6. Results with varying ways.

METHOD	MINIIMAGENET 10 WAY	MINIIMAGENET 20 WAY	CUB 10 WAY	CUB 20 WAY
MATCHING$^*_+$	49.76±0.45	35.29± 0.24	66.33±0.62	53.46± 0.33
MATCHING$^*_+$+MVT (OURS)	**52.56±0.44**	**38.01± 0.24**	**69.93±0.48**	**56.17± 0.32**

Table 7. Results with varying shots.

METHOD	MINIIMAGENET 2 SHOT	MINIIMAGENET 3 SHOT	MINIIMAGENET 4 SHOT	MINIIMAGENET 20 SHOT
MATCHING$^*_+$	56.12±0.73	60.15± 0.76	62.82±0.74	72.70± 0.60
MATCHING$^*_+$+MVT (OURS)	**57.74±0.77**	**61.48±0.76**	**64.75±0.71**	**75.20± 0.59**

METHOD	CUB 2 SHOT	CUB 3 SHOT	CUB 4 SHOT	CUB 20 SHOT
MATCHING$^*_+$	69.72±0.77	74.39± 0.74	76.07±0.66	83.03± 0.54
MATCHING$^*_+$+MVT (OURS)	**71.32±0.79**	**75.93±0.70**	**78.10±0.68**	**84.37± 0.51**

4.4. MVT with conventional augmentations

In Table 3, we show the comparative results with and without the conventional data augmentation methods. Here we additionally report the results of 20-shots and we use ResNet-10 for the results. We evaluated the results on fine-grained classification datasets, namely CUB, miniVGG, and miniCASIA. For the face datasets, we used 112×112 images, extracted the facial region using a face detector (Zhang et al., 2016), and skipped the max pooling to match the spatial dimension of the feature of 224×224 CUB data. During meta-training, we used random crop and flip on each image as the conventional augmentations for the CUB data. However, for the face dataset, we use color jittering instead of random crop because the face images are well aligned in all datasets. The results confirm that our MVT is orthogonal to conventional augmentations, and can bring in further performance enhancement when used together. On small datasets, the conventional augmentation is more effective, but its performance improvement diminishes on a larger dataset (miniVGG). On the contrary, the proposed MVT method achieves consistent performance improvements any datasets regardless of their sizes.

4.5. Generalization to different backbone networks

We validate if the proposed MVT could generalize to different backbone networks. In Table 5, we report the results of different backbone architectures with and without the proposed MVT. We used the network$_+$ for all the results. The results show that the proposed MVT method consistently improved the accuracy regardless of the architectures of the backbone networks. Interestingly, MVT-augmented models with 5-shot classification obtain comparable results with 10-shot classification performance of the backbone networks. For example, ResNet-10 with *5 shots* using MVT (85.35) obtain similar performance to ResNet-10 with *10 shots* (85.67). The results also show that the proposed MVT with a certain backbone results in performance comparable to a deeper version of that backbone. For example, ResNet-10 with 10 shots using MVT (88.41) obtain comparable performance to ResNet-34 with 10 shots (88.71).

4.6. Results with varying ways and shots

In addition to the common experimental setting of 5-way classification with 5- and 10- shots, we conducted experiments with varying ways and shots to see the robustness of our method to different settings. Table 6 shows the results with 10, 20-way using 5 shots, while the Table 7 shows the results of varying shots from 2 to 20 shots with 5-way. Note that our MVT requires at least two shots to compute a variance. Overall, the results show that our method robustly improves the accuracy in all settings.

5. Conclusion

We proposed a novel meta-learning framework for classification models, *Meta-Variance Transfer (MVT)* which learns to transfer factors of variations from one class to another, such that it improves the overall classification performance. Specifically, we capture the variations for each class as the difference of each sample from the class prototype, and meta-learn a network to transform it to another class such that the transformed virtual point helps lower the classification loss for the target class. We first perform a proof of concept of the model with a face recognition dataset, which shows that MVT is able to transfer meaningful factors of variations across classes. Further experimental validation on few-shot classification and face recognition shows that MVT significantly improves the performance of a model, orthogonally to existing data augmentation methods.

References

Antoniou, A., Storkey, A., and Edwards, H. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., and Zisserman, A. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 67–74. IEEE, 2018.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C., and Huang, J.-B. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019a.

Chen, Z., Fu, Y., Chen, K., and Jiang, Y.-G. Image block augmentation for one-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3379–3386, 2019b.

Chen, Z., Fu, Y., Wang, Y.-X., Ma, L., Liu, W., and Hebert, M. Image deformation meta-networks for one-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8680–8689, 2019c.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.

Dixit, M., Kwitt, R., Niethammer, M., and Vasconcelos, N. Aga: Attribute-guided augmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7455–7463, 2017.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135, 2017.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pp. 9516–9527, 2018.

Gao, H., Shou, Z., Zareian, A., Zhang, H., and Chang, S.-F. Low-shot learning via covariance-preserving adversarial augmentation networks. In *Advances in Neural Information Processing Systems*, pp. 975–985, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Gross, R., Matthews, I., Cohn, J., Kanade, T., and Baker, S. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

Hariharan, B. and Girshick, R. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3018–3027, 2017.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

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

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

Koch, G. Siamese neural networks for one-shot image recognition. 2015.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4681–4690, 2017.

Lee, H., Nam, T., Yang, E., and Hwang, S. J. Meta dropout: Learning to perturb latent features for generalization. In *International Conference on Learning Representations*, 2020.

Lee, Y. and Choi, S. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pp. 2933–2942, 2018.

Li, Z., Zhou, F., Chen, F., and Li, H. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Luc, P., Couprie, C., Chintala, S., and Verbeek, J. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*, 2016.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

Miller, E. G., Matsakis, N. E., and Viola, P. A. Learning from one example through shared densities on transforms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pp. 464–471. IEEE, 2000.

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

Munkhdalai, T. and Yu, H. Meta networks. In *International Conference on Machine Learning*, pp. 2554–2563, 2017.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Oreshkin, B., López, P. R., and Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 721–731, 2018.

Pascual, S., Bonafonte, A., and Serra, J. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. 2016.

Schwartz, E., Karlinsky, L., Shtok, J., Harary, S., Marder, M., Kumar, A., Feris, R., Giryes, R., and Bronstein, A. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *Advances in Neural Information Processing Systems*, pp. 2845–2855, 2018.

Shmelkov, K., Schmid, C., and Alahari, K. How good is my gan? In *Proceedings of the European Conference on Computer Vision*, pp. 213–229, 2018.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2107–2116, 2017.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

Tsutsui, S., Fu, Y., and Crandall, D. Meta-reinforced synthetic data for one-shot fine-grained visual recognition. In *Advances in Neural Information Processing Systems*, pp. 3057–3066, 2019.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wang, Y.-X., Girshick, R., Hebert, M., and Hariharan, B. Low-shot learning from imaginary data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7278–7286, 2018.

Yi, D., Lei, Z., Liao, S., and Li, S. Z. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10): 1499–1503, 2016.

Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., and Song, Y. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 2365–2374, 2018.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., and Whiteson, S. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pp. 7693–7702, 2019.