Variational Feature Disentangling for Fine-Grained Few-Shot Classification Supplementary Material

Jingyi Xu[1], Hieu Le[*2], Mingzhen Huang[1], ShahRukh Athar[1], and Dimitris Samaras[1]

[1]Department of Computer Science, Stony Brook University, NY, USA
[2]Amazon Robotics, MA, USA

1. Overview

We provide additional experiments and analyses of our proposed method. In particular:

- We provide the results of our method on the CIFAR-FS [6] and mini-ImageNet [17] datasets and compare it with the state-of-the-art methods in Section 2.
- We provide results of additional experiments that demonstrate the effectiveness of our method in dealing with the problem of an imbalanced training set with long-tail training classes in Section 3.
- We analyze of the sensitivity of our method to the number of training instances in Section 4.
- We provide a visualization of augmented samples on the MNIST dataset generated by our method using the posterior distribution in comparison with using Gaussian Noise in Section 5
- In Section 6, we provide additional experiments on the nearest "real sample" neighbors to show that augmented features from our methods lie close to the real ones.

2. Performance on the non-fine-grained few-shot datasets

Our method works particularly well on fine-grained datasets including the CUB[18], NAB[16], and Stanford Dogs[4] datasets where the intra-class variations are similar across classes. Here we provide additional results on the non-fine-grained few-shot datasets such as CIFAR-FS [6], and mini-Imagenet[17], summarized in Tables 1 and 2 respectively. On both datasets, our method outperforms other methods in 5-shot setting by a small margin and achieves competitive performance in 1-shot setting. For the aforementioned datasets, intra-class variations between classes are very different in nature, for e.g., the variation of the "*wok*" class would bear no resemblance to variations in the "*jellyfish*" class or the unicycle "*unicycle*" class of the mini-Imagenet dataset as they are very different objects. Therefore, constructing a common embedding space to model the intra-class variability, which is crucial to our method, is challenging. However, our method still gets competitive results. This implies that our method can be used even when we cannot be sure about the type of variability in the dataset, without a performance penalty

3. Performance on long-tailed training data

In the paper we have shown that the intra-class variance can be transferred from the base classes to to augment examples from novel classes. In this experiment, we investigate whether or not the intra-class variance can also be transferred across different training classes to deal with long-tailed training data, where the number of training instances of different classes is highly imbalanced.

[*]Work done prior to Amazon

Method	Backbone	1-shot	5-shot
MAML [3]	32-32-32-32	58.9 ± 1.9	71.5 ± 1.0
ProtoNet [13]	64-64-64-64	55.5 ± 0.7	72.0 ± 0.6
RelationNet [15]	64-96-128-256	55.0 ± 1.0	69.3 ± 0.8
R2D2 [1]	96-192-384-512	65.3 ± 0.2	79.4 ± 0.1
Shot-Free [10]	ResNet12	69.2 ± n/a	84.7 ± n/a
TEWAM [9]	ResNet12	70.4 ± n/a	81.3 ± n/a
ProtoNet[13]	ResNet12	72.2 ± 0.7	83.5 ± 0.5
MetaOptNet [6]	ResNet12	**72.6** ± 0.7	84.3 ± 0.5
Ours	ResNet12	72.3 ± 0.8	**84.9** ± 0.6

Table 1. Few-shot classification accuracy on the CIFAR-FS dataset. a-b-c-d denotes a 4-layer convolutional network with a, b, c, and d filters in each layer. The best performance is indicated in bold.

Method	Backbone	1-shot	5-shot
MatchingNet [17]	Conv4	43.56 ± 0.84	55.31 ± 0.73
ProtoNet [13]	Conv4	48.70 ± 1.84	63.11 ± 0.92
LEO [11]	WRN-28-10	**61.76** ± 0.08	77.59 ± 0.12
SNAIL [7]	ResNet12	55.71 ± 0.99	68.88 ± 0.92
TADAM [8]	ResNet12	58.50 ± 0.30	76.70 ± 0.30
MTL [14]	ResNet12	61.20 ± 1.80	75.50 ± 0.80
Variational FSL [19]	ResNet12	61.23 ± 0.26	77.69 ± 0.17
MetaOptNet [6]	ResNet12	60.33 ± 0.61	76.61 ± 0.46
Ours	ResNet12	61.40 ± 1.15	**81.10** ± 0.83

Table 2. Few-shot classification accuracy on the mini-Imagenet dataset. The best performance is indicated in bold.

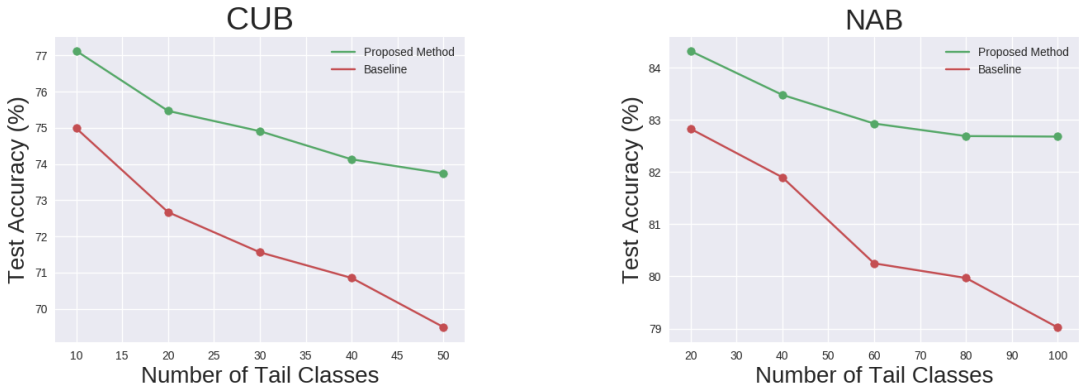

Figure 1. Few-shot classification accuracy on the CUB[18] and NAB [16] datasets as the number of tail classes increases. Compared to the baseline [2], our method is more resilient to long-tailed data due to the transfer of intra-class variance during the training stage.

We manually create some classes with insufficient training samples (10 samples each class) on the CUB[18] and NAB[16] datasets, and compare the performance of our method with the baseline [2] as the number of tail classes varies. As shown in Figure 1, the performance of both methods degrades as the number of tail classes increases. However, we observe that our method is more resilient to the number of classes with insufficient training data compared to the baseline[2]. We attribute such resilience to the augmented features generated in the training stage, which transfer the intra-class variance between training classes. Such an augmentation alleviates the issue of imbalanced training data and leads to more discriminative feature representations.

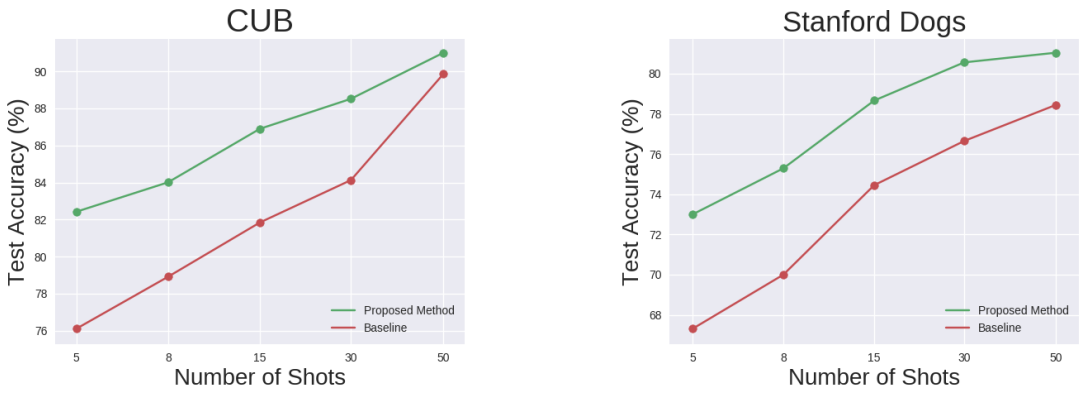

Figure 2. Few-shot classification accuracy on the CUB[18] and Stanford Dogs [4] datasets as the number of support samples per class increases. Compared to the baseline [2], our method is less sensitive to the number of shots.

4. Sensitivity to the number of support examples per class

Figure 2 shows how the classification accuracy varies as a function of the number of support examples per class (shots) on the CUB [18] and Stanford Dogs [4] datasets. We compare our method with the baseline method [2] with a Conv4 backbone. As expected, the number of support samples per class is highly correlated with the classification accuracy: more labeled samples per class would lead to better performance. However, we observe that our proposed method consistently outperforms the baseline method [2] and is less sensitive to the number of shots. Since our augmented samples can effectively enlarge the

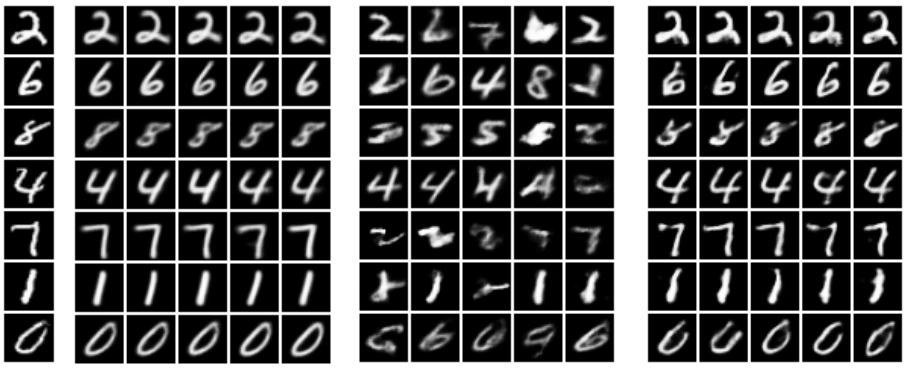

Figure 3. Visualization Results on the MNIST [5] dataset. The left-most images are the original images. (a) Images reconstructed from a traditional variational autoencoder model are basically identical to each other. (b) Adding Gaussian noise to the latent space of a traditional variational autoencoder produces meaningless outputs . (c) Sampling repeatedly from the disentangled intra-class variance of our model produces images which exhibit noticeable variations while preserving the attributes of the corresponding digit class.

intra-class variance, even with few labeled examples given, it will compensate the lack of training data and the performance will thus be less sensitive to the number of support samples per class.

5. Visualization Results on MNIST

We provide the visualization of the augmented features of our method on the MNIST dataset. As discussed in Section 4 of the paper, the representation of an input image is decomposed into two parts, the class-specific feature and the intra-class variance. We use four Convolutional layers to obtain a feature map of size $32\times4\times4$. An average pooling layer is applied

to the feature map to generate the class-specific feature and two fully connected layers are used to generate the intra-class variance. The decoder architecture is the transpose of the encoder. Unlike in the paper, the entire input image is reconstructed instead of the feature map.

Figure 3a shows the images reconstructed from a traditional variational autoencoder model (VAE), which are almost identical and have very little variation. Figure 3b shows the images reconstructed from a traditional VAE by adding Gaussian noise to the latent space. As can be seen from the figure, simply adding gaussian noise causes the decoder to generate images that do not lie on the MNIST image manifold. Figure 3c shows the images reconstructed from our model by sampling repeatedly from the learned intra-class variance distribution. As can be seen in Figure 3c, images reconstructed using our method show noticeable variation while preserving the attributes of the corresponding digit class. Consequently, such an augmentation can help augment training instances of the novel classes when training a classifier on them.

6. Analysis on nearest "real sample" neighbors.

We analyze the fidelity of augmented features to their classes via searching for their K-nearest "real-sample" neighbors (K=1,3, or 5). We train our method and delta-encoder[12] using the base classes and use the trained models to augment features of the novel classes. We then search for the nearest-neighbors of the augmented features in all the real samples in the novel classes of the CUB dataset [18]. Some nearest-neighbors in the image space are visualized in *Figure 1* in the main paper. The quantitative results are summarized in Table 3. As can be seen, only 8.7% of the augmented samples from delta-encoder have their nearest neighbors from the same classes as the original features. Even in the case when considering 3 and 5 nearest neighbors, only 27.9% and 32.8% of them belong to the same class as the original features. This suggest that the augmented features from delta-encoder do not tend to preserve the class-specific features of the original one.

Augmented features from our methods have significantly higher class fidelity: 35.4% of the features having their top nearest neighbors belong to the same classes as the original features. In the cases of 3-NN and 5-NN, 54.8% and 64.5% of the nearest neighbors have the same class as the original features, respectively. Note that the KNN algorithm is sensitive to "hard examples" that lie close to the class boundaries, as can be seen in the first row of Table 3.

	1-NN	3-NN	5-NN
Real Samples	43.2	63.9	71.7
Δ-Encoder [12]	8.7	27.9	32.8
Ours	35.4	54.8	64.5

Table 3. **Analysis of the nearest "real sample" neighbors**. The table shows the percentages of the nearest "real sample" neighbors of the augmented features that belong to the same class as the original features. We train methods on the base classes and test on the novel classes of the CUB [18] dataset.

References

[1] Luca Bertinetto, João F. Henriques, P. Torr, and A. Vedaldi. Meta-learning with differentiable closed-form solvers. *ArXiv*, abs/1805.08136, 2019. 2

[2] 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. 2, 3

[3] 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

[4] 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, 3

[5] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit databases. Technical Report, 2014. 3

[6] 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. 1, 2

[7] N. Mishra, Mostafa Rohaninejad, Xi Chen, and P. Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018. 2

[8] Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018. 2

[9] L. Qiao, Y. Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning, 2019. 2

[10] A. Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training, 2019. 2

[11] Andrei A. Rusu, D. Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. volume abs/1807.05960, 2019. 2

[12] 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. 4

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

[14] 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. 2

[15] 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

[16] 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

[17] 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. 1, 2

[18] 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, 3, 4

[19] 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