
CONTENTS

| | |
|---|----------|
| A Appendix | 1 |
| A.1 Implementation Details | 1 |
| A.1.1 Meta-training Hyper-parameters | 1 |
| A.1.2 Dataset Details | 1 |
| A.2 Additional Results | 3 |
| A.2.1 Additional Meta-training Settings | 3 |
| A.2.2 Hyper-parameter Studies | 5 |
| A.2.3 Additional Experiments on Different Clustering Structures | 6 |
| A.3 Additional Discussion on the Learned Clustering | 7 |

A APPENDIX

A.1 IMPLEMENTATION DETAILS

A.1.1 META-TRAINING HYPER-PARAMETERS

We summary the hyper-parameters for meta-training in our experiments in Table 1. In general, we follow the basic settings in (Yao et al., 2019). The base learner is a standard four-block convolutional neural network. The number of nodes for the hierarchical task clustering network is set to 4,4,1 to accommodate larger clustering capacity on OOD meta-testing tasks. The constructed pool has 16 clusters, each of which has a capacity of 320 images (20 classes * 16 images per class). Every 2 epochs, we re-calculate the probability scores for all images in the pool to keep the pool up-to-date. We adopt a warm-start strategy by sampling tasks from the pool after 3 epochs, since it makes no sense to regularize with a hypervolume loss on the basis of a random pool. We structurize 4 tasks (i.e., 2 \mathcal{T}_m s and 2 \mathcal{T}_p s) by designing the random variable λ from $Beta(5, 2)$ and $Beta(2, 5)$ for 2 \mathcal{T}_m s, respectively. In this way, we can ensure that the generated mixed tasks are not too close (or far away) with each other, so as to better imitate OOD tasks. We meta-train our model on a single RTX 2080-Ti GPU. We summarize the whole framework of our proposed method in Algorithm 1.

A.1.2 DATASET DETAILS

In this section, we briefly introduce the datasets we use in our experiments. All images are converted into (84×84) pixels of widths and heights with RGB channels. We randomly sample 16 images for each dataset as illustrated in Figure 1.

- **Meta-Dataset** (Triantafillou et al., 2019) is a cross-domain image datasets including 10 sub-datasets from real to hand-drawn images.
 - **Fine-Grained Visual Classification of Aircraft (Aircraft)** (Maji et al., 2013). We follow the same setting as in (Yao et al., 2019), that meta-training/meta-validation/meta-testing sets are split to contain 64/16/20 classes. Each aircraft variant contains 100 images.
 - **Caltech-UCSD Birds-200-2011 (Birds)** (Wah et al., 2011). We follow the same setting as in (Yao et al., 2019), that meta-training/meta-validation/meta-testing sets are split to contain 64/16/20 classes. Each bird species contains 60 images.
 - **Describable Textures (Textures)** (Cimpoi et al., 2014). We follow the same setting as in (Yao et al., 2019), that meta-training/meta-validation/meta-testing sets are split to contain 30/7/10 classes. Each texture class contains 120 images.
 - **FGVCx-Fungi (Fungi)** (Kaggle, 2018). We follow the same setting as in (Yao et al., 2019), that meta-training/meta-validation/meta-testing sets are split to contain 64/16/20 classes. Each mushroom species contains 150 images.
 - **ILSVRC-2012 (ImageNet)** (Russakovsky et al., 2015) is a well-established comprehensive dataset for image classification. Here, we do not use the full dataset. In practice, we use the

Algorithm 1: Meta-training of the Proposed Framework

```
1: Require: termination condition  $T$ ; outer learning rate  $\beta$ ; meta batch size  $B$ ; shot  $K$ ; way  $N$ 
2: Require: cluster number  $C$ ; pool update period  $T_u$ 
3: Require: hypervolume loss weight  $\alpha$ ; reference point  $\mathcal{Z}$ 
4: Initialize pool  $\mathcal{C} = \{\mathcal{C}_c\}_{c=1}^C \leftarrow \{\emptyset\}_{c=1}^C$ 
5: Randomly initialize the clustering network and base learner  $\theta_{all} = \{\theta_{cn}, \theta_{bl}\}$ 
6: for  $t = 1$  to  $T$  do
7:   Sample a batch of tasks  $\{\mathcal{T}_i\}_{i=1}^B$ 
8:   Compute  $\mathcal{L}_{train}$  by HSML
9:
10:  /* Clustering Pool Construction */
11:  if  $\text{mod}(t, T_u) == 0$  then
12:    for  $c = 1$  to  $C$  do
13:      for  $\hat{x}$  in  $\mathcal{C}_c$  do
14:        Update  $p_{\hat{x}}$  in Equation (1)
15:      end for
16:    end for
17:  end if
18:   $P \leftarrow \mathcal{C}$ 
19:  for each  $\mathcal{T}_i$  do
20:    for  $x$  in  $\mathcal{T}_i$  do
21:      Construct auxiliary task  $\mathcal{T}_x$  with  $\mathcal{D}_{\mathcal{T}_x}^{(s)} \leftarrow \{(x, y_j)\}_{j=1}^{NK}$ 
22:      Calculate  $p_x$  in Equation (1)
23:       $P \leftarrow P \cup p_x$ 
24:    end for
25:  end for
26:  Apply  $k$ -means on  $P$  to have  $\{\mathcal{C}_c\}_{c=1}^C \leftarrow P$ 
27:
28:  /* Task Sampling from Pool */
29:  Sample  $\mathcal{C}_i, \mathcal{C}_j$  from  $\mathcal{C}$ 
30:  Sample  $\mathcal{T}_{pi}, \mathcal{T}_{pj}, \mathcal{T}_{mi}, \mathcal{T}_{mj}, \mathcal{T}_{oi}, \mathcal{T}_{oj}$  from  $\mathcal{C}_i, \mathcal{C}_j$  in subsection 4.3
31:
32:  /* Conflict Loss Computation */
33:  Compute the multi-objective query loss matrix  $\mathcal{L}_{mo}$  in Equation (3)
34:  Compute  $\mathcal{L}_{HV}(\mathcal{L}_{mo}, \mathcal{Z})$ 
35:
36:  /* Meta Training */
37:  Compute  $\nabla \mathcal{L}_{total} = \nabla_{\Theta} \mathcal{L}_{train} + \alpha \nabla_{\theta_{cn}} \mathcal{L}_{HV}$ 
38:  Update  $\Theta \leftarrow \Theta - \beta \nabla \mathcal{L}_{total}$ 
39: end for
```

commonly used subset **Mini** (Vinyals et al., 2016) as a substitution. We randomly select 20 classes for meta-testing, each containing 600 images.

- **Omniglot** (Lake et al., 2015) contains 1623 hand-written characters from different alphabets. We randomly select 659 characters for meta-testing, each containing 20 images.
- **VGG Flower** (Nilsback & Zisserman, 2008) contains 102 flower categories. We randomly select 16 classes for meta-testing, each containing around 100 images.
- **Quickdraw** (Jongejan et al., 2016) contains 345 online hand-drawn categories. We use a subset of 500 images for each class described in DomainNet. We randomly select 100 categories for meta-testing.
- **Traffic Signs** (Houben et al., 2013) contains 43 classes of German road signs. Images are in different illumination conditions and blurs. All classes are used for meta-testing.
- **MSCOCO** (Lin et al., 2014) contains 80 classes of objects localized in bounding boxes of original images. All classes are used for meta-testing.

Table 1: Hyper-parameters summary.

| | Hyper-parameters | Values |
|---------------------------|---|-------------------|
| Base learner | Meta batch size | 4 |
| | Inner loop learning rate | 0.01 |
| | Outer loop learning rate | 0.0001 |
| | Inner step | 3 |
| | Outer step | 15 |
| | CNN block number | 4 |
| | CNN filter number | 48 |
| Clustering network | Node number | (4, 4, 1) |
| | Hidden dim | 128 |
| | Reconstruction loss weight | 0.01 |
| Pool construction | Cluster capacity (image number) | 320 |
| | Cluster number C | 16 |
| | Pool update period (epoch) | 2 |
| Task sampling | Start sampling epoch | 3 |
| | $\mathcal{T}_p, \mathcal{T}_m, \mathcal{T}_o$ numbers | (2, 2, 2) |
| | CutMix bounding box size | (25, 25) |
| | Beta parameter (a, b) | (5, 2) and (2, 5) |
| Conflict loss calculation | Hypervolume loss weight α | 0.1 |
| | Reference point \mathcal{Z} | $[1.5, 1.5]^\top$ |
| Dataset | Class number N | 5 |
| | Shot number K | 1 |
| | Query sample number $n^{(a)}$ | 75 |
| | Image shape | (84, 84, 3) |

- **DomainNet** (Triantafillou et al., 2019) is a multi-source datasets including 6 distinct domains (i.e., **Clipart**, **Infograph**, **Painting**, **Quickdraw**, **Real**, **Sketch**) with similar class labels. We randomly select 100 classes for each domain, each containing around 500 images.
- **CIFAR-100** (Krizhevsky et al., 2009) is a low resolution image dataset containing 100 fine-grained categories. All classes are used for meta-testing.
- **Stanford Cars (Cars)** (Krause et al., 2013) contains 196 car classes. Different from the given default image-level splitting, we randomly select 49 classes for meta-testing, each containing around 40 images.
- **Oxford-IIIT Pets (Pets)** (Parkhi et al., 2012) contains 37 dog and cat categories. Each image has a ground truth bounding box around the head of the animal. We randomly select 20 classes for meta-testing, each containing 100 images.
- **Stanford Dogs (Dogs)** (Khosla et al., 2011) contains 120 breeds of dogs. We randomly select 30 classes for meta-testing, each containing hundreds of images.

A.2 ADDITIONAL RESULTS

A.2.1 ADDITIONAL META-TRAINING SETTINGS

After testing the effectiveness of our proposed framework on the commonly used 1-shot 5-way meta-training scenario, we further apply it to additional meta-training settings.

Testing on a base learner with less capacity. We report the average meta-testing accuracy in Table 2, 3 when decreasing the number of filters to 32. Our method achieves similar performance on ID datasets (i.e., 49.62% accuracy on average comparing with HSML 49.29% accuracy) but also shows consistently outperforming accuracy (i.e., 42.85% on average comparing with HSML 41.77% accuracy). Comparing with the results in Table ??, we can observe a smaller improvement on average (i.e., 1.08% vs 2.01%) between Ours and HSML. We can conclude that a more disentangled clustering is of benefit to generalize to OOD tasks for a base learner with higher capacity.

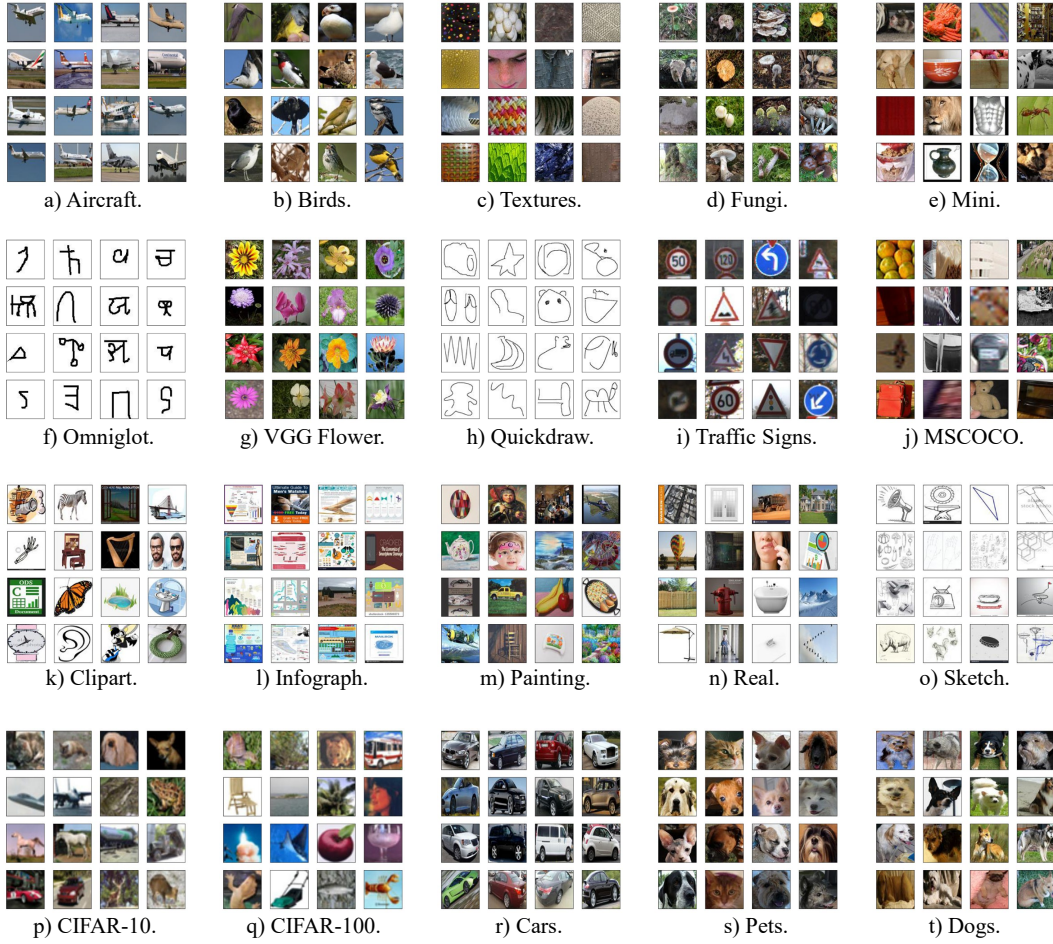


Figure 1: Image examples from all datasets used in the experiments.

Table 2: ID meta-testing accuracy comparison of our method to HSML meta-trained with Aircraft, Birds, Textures, and Fungi. Base learners have 32 filters in each layer. Accuracy (standard deviation) are reported.

| Test Dataset | Aircraft | Birds | Textures | Fungi | ID Average |
|--------------|----------------------|----------------------|----------------------|----------------------|---------------|
| HSML | 55.92%(0.30%) | 62.45%(0.39%) | 33.71%(0.30%) | 45.10%(0.21%) | 49.29% |
| Ours | 56.48%(0.37%) | 62.12%(0.35%) | 34.83%(0.32%) | 45.06%(0.18%) | 49.62% |

Table 3: OOD meta-testing accuracy comparison of our method to HSML meta-trained with Aircraft, Birds, Textures, and Fungi. Base learners have 32 filters in each layer. Accuracy (standard deviation) are reported.

| Test Dataset | Mini | Traffic Signs | Real | CIFAR-100 | Pets | OOD Average |
|--------------|----------------------|----------------------|----------------------|----------------------|----------------------|---------------|
| HSML | 37.10%(0.28%) | 44.48%(0.36%) | 42.08%(0.23%) | 39.49%(0.31%) | 45.72%(0.31%) | 41.77% |
| Ours | 38.40%(0.29%) | 45.37%(0.35%) | 43.01%(0.31%) | 40.93%(0.27%) | 46.55%(0.28%) | 42.85% |

A.2.2 HYPER-PARAMETER STUDIES

Effect of different objective numbers. The number of objectives is the number of randomly sampled columns from the pool in each iteration. A larger number indicates a larger scope considered to encourage disentanglement simultaneously. We investigate this effect in Figure 2. We do not observe better OOD performance in the 3-objective case, which supports our claim that it is computationally efficient to enhance pair-wise cluster difference.

Effect of different mixed task numbers. We study the effectiveness of our framework when varying the number of mixed tasks generated in each iteration. The meta-testing accuracy is reported in Figure 3. We do not observe a clear tendency when increasing the number of mixed tasks. Regarding the computational cost, we use 2 mixed tasks in our main experiments.

Effect of hypervolume loss weights. The weight of hypervolume loss α controls the importance between the meta-training loss and the hypervolume loss. We investigate the effect of hypervolume loss weights in Table 4. Note that, the zero weight equals to the standard HSML. For ID datasets, increasing the weight does not produce a better meta-testing accuracy, which shows that the learned clustering in HSML is enough for distinguishing ID datasets. However, this can be further promoted for OOD datasets with our hypervolume loss, since the meta-testing accuracy for OOD datasets shows a significant increasing trend when increasing the hypervolume loss weight.

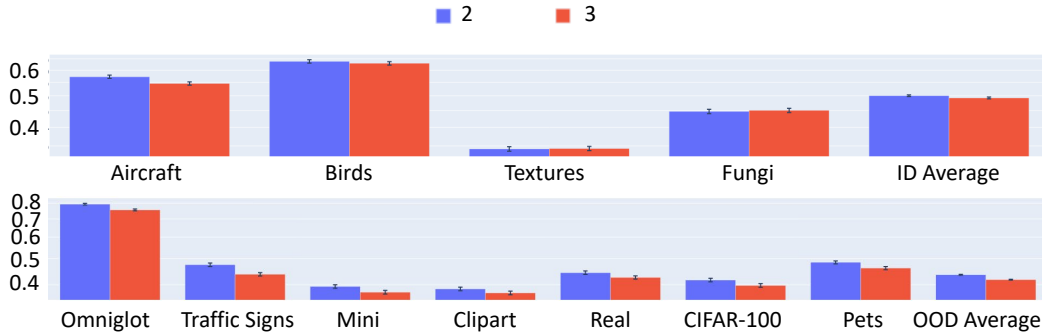


Figure 2: Meta-testing accuracy for varying number of objectives (blue: 2-objective, red: 3-objective) on 1-shot 5-way experiments meta-trained with Aircraft, Birds, Textures, and Fungi datasets. The number of mixed tasks generated in each iteration is set to 2 and 3 for 2-objective and 3-objective cases, respectively.

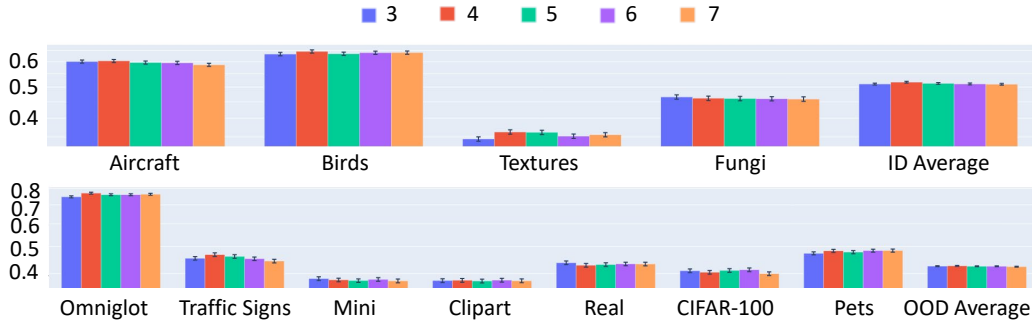


Figure 3: Meta-testing accuracy for varying number of mixed tasks on 1-shot 5-way experiments meta-trained with Aircraft, Birds, Textures, and Fungi datasets. The number of objectives is set to 2.

Table 4: Comparison of different settings of hypervolume loss weights on meta-testing accuracy over 1000 tasks for each dataset. Models are all meta-trained with Aircraft, Birds, Textures, and Fungi datasets.

| α | Aircraft | Birds | Textures | Fungi | Mini | Traffic Signs | VGG Flower | Omniglot |
|----------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| 0.00 | 59.54% | 64.18% | 34.84% | 46.82% | 36.40% | 44.38% | 68.06% | 77.88% |
| 0.01 | 57.17% | 63.56% | 35.32% | 46.51% | 36.83% | 44.65% | 67.12% | 76.08% |
| 0.05 | 60.41% | 64.25% | 35.73% | 46.36% | 37.71% | 45.39% | 68.64% | 75.06% |
| 0.10 | 57.75% | 63.38% | 34.96% | 46.40% | 38.12% | 45.16% | 67.15% | 78.15% |
| 0.50 | 46.90% | 59.49% | 33.63% | 42.73% | 37.79% | 48.45% | 68.92% | 79.35% |
| 1.00 | 42.55% | 57.36% | 31.75% | 41.94% | 36.81% | 47.99% | 69.54% | 78.31% |

Effect of different mixing methods. Mixed tasks are essential components in *Task Sampling*, which are generated to mimic OOD tasks from meta-training ID tasks. To this end, our method performs CutMix (Yun et al., 2019) task augmentation to generate mixed tasks. We investigate the effect of MixUp (Zhang et al., 2017) task augmentation. For each image-pair $(\tilde{x}_{1i}, \tilde{x}_{2i})$, we calculate the mixed image $\tilde{x}_i = \lambda \tilde{x}_{1i} + (1 - \lambda) \tilde{x}_{2i}$. Note that we sample λ using the same strategy as described in *Task Sampling* part. We further develop a variant of MixUp (named MixUp-R), which is to mix the task representations of each image-pair rather than the images themselves.

We compare CutMix, MixUp, MixUp-R on meta-testing accuracy over 1000 tasks for each dataset. The 5-way 1-shot experiment results are shown in Table 5. We can not observe significant difference among these methods, but CutMix works better in general.

A.2.3 ADDITIONAL EXPERIMENTS ON DIFFERENT CLUSTERING STRUCTURES

Different clustering network architectures. In order to show the benefit of a larger capacity of the clustering network, we evaluate three different architectures (i.e., (4,2,1), (4,4,1), and (8,4,1) structures with 8, 16, and 32 clusters in the pool, respectively). The meta-testing accuracy is reported in Figure 4 with some representative OOD datasets (i.e., Traffic Signs, Mini, Clipart, Real, CIFAR-100, and Dogs) and the average of all OOD datasets. It is clear that a larger capacity leverages improvement on OOD meta-testing.

SpectralNet. Recent studies on SpectralNet (Shaham et al., 2018; Yang et al., 2019) show promising results on promoting disentangled clustering. We compare our method with a HSML variant (named HSML-SN) that use SpectralNet (Yang et al., 2019) as a substitution of the clustering network. We use a meta batch size of 256, which is much larger than the meta batch size we use for HSML and our method (i.e., 4), so as to well capture the structure of the data for each task batch. The dimension of the network output (i.e., cluster number) is set to the same number w.r.t. hierarchical clustering network in HSML (i.e., 16).

Table 5: Comparison of different settings of mixing methods on meta-testing accuracy over 1000 tasks for each dataset. Models are all meta-trained with Aircraft, Birds, Textures, and Fungi datasets.

| Method | Aircraft | Birds | Textures | Fungi | Mini | Traffic Signs | VGG Flower | Omniglot |
|---------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| CutMix | 56.48% | 62.12% | 34.83% | 45.06% | 38.40% | 45.37% | 66.62% | 76.08% |
| MixUp | 54.94% | 62.01% | 34.28% | 44.93% | 37.62% | 45.29% | 58.14% | 75.34% |
| MixUp-R | 54.85% | 62.05% | 34.28% | 44.41% | 38.23% | 44.19% | 68.44% | 75.74% |

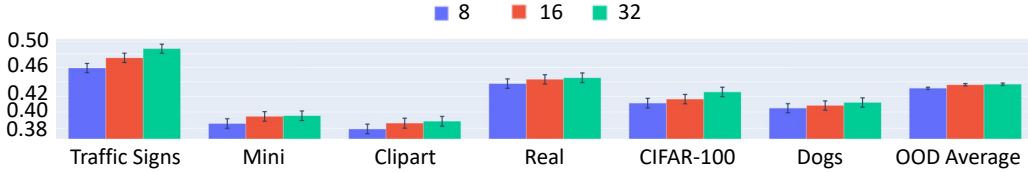


Figure 4: Meta-testing accuracy for different numbers of clusters on 1-shot 5-way experiments meta-trained with Aircraft, Birds, Textures, and Fungi datasets. The clustering network architecture for 8, 16, and 32 are (4, 2, 1), (4, 4, 1), and (8, 4, 1), respectively.

We compare HSML-SN with HSML as well as our method in terms of the meta-testing accuracy over 1000 tasks for each OOD dataset. The 5-way 1-shot experiment results are shown in Table 6. SpectralNet does not bring better OOD meta-testing performance than hierarchical clustering network in HSML and our method within limited meta-training iterations. Our method outperforms HSML-SN on most of OOD datasets, which hints the advanced clustering learned by our method.

A.3 ADDITIONAL DISCUSSION ON THE LEARNED CLUSTERING

We analyse the learned clustering of HSML and our method using the pool described in *Clustering Pool Construction*. We visualize images whose probability scores are top-16 closest to 16 clustering centers in Figure 5. It can be clearly observed that the learned features (clusters) are different for HSML and our method. HSML has some duplicated clusters (i.e., 2 similar Birds and 2 similar Texture clusters). Our method tends to learn more implicit features than HSML.

REFERENCES

- Mircea Cimpoi, Subhansu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.
- Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, pp. 1–8. Ieee, 2013.
- Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. The quick, draw!-ai experiment. *Mount View, CA, accessed Feb*, 17(2018):4, 2016.
- Kaggle. 2018 fgcvx fungi classification challenge, 2018.
- Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR Workshop on Fine-Grained Visual Categorization*, volume 2. Citeseer, 2011.
- Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *4th International IEEE Workshop on 3D Representation and Recognition*, Sydney, Australia, 2013.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Table 6: Comparison of our method with HSML-SN on meta-testing accuracy over 1000 tasks for each OOD dataset. Models are all meta-trained with Aircraft, Birds, Textures, and Fungi datasets.

| Model | Mini | Traffic Signs | VGG Flower | Omniglot |
|---------|---------------|---------------|---------------|---------------|
| HSML | 36.62% | 47.53% | 67.14% | 74.03% |
| HSML-SN | 30.95% | 42.98% | 64.03% | 70.96% |
| Ours | 38.92% | 48.52% | 69.19% | 80.12% |

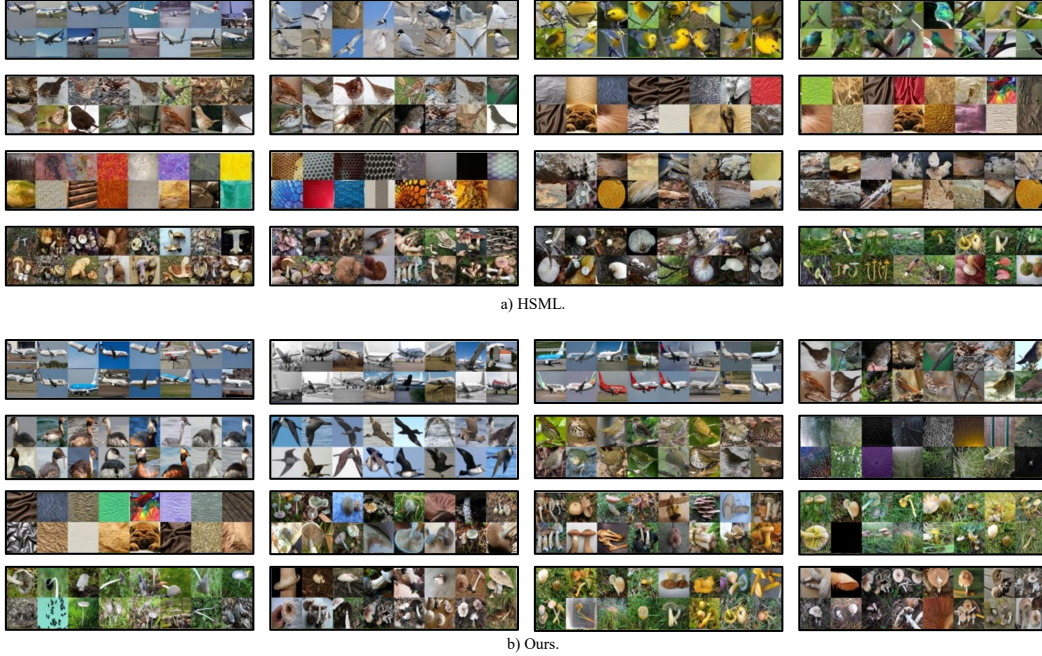


Figure 5: Image examples from learned pool.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. In *Computer Vision and Pattern Recognition*, 2013.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Uri Shaham, Kelly Stanton, Henry Li, Boaz Nadler, Ronen Basri, and Yuval Kluger. Spectralnet: Spectral clustering using deep neural networks. *arXiv preprint arXiv:1801.01587*, 2018.

-
- Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2019.
- Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29, pp. 3630–3638, 2016.
- Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep Spectral Clustering Using Dual Autoencoder Network. In *Computer Vision and Pattern Recognition*, June 2019.
- Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *International Conference on Machine Learning*, pp. 7045–7054. PMLR, 2019.
- Sangdo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.
- Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.