A MULTI-OBJECTIVE PERSPECTIVE TOWARDS IMPROVING META-GENERALIZATION

Anonymous authors
Paper under double-blind review

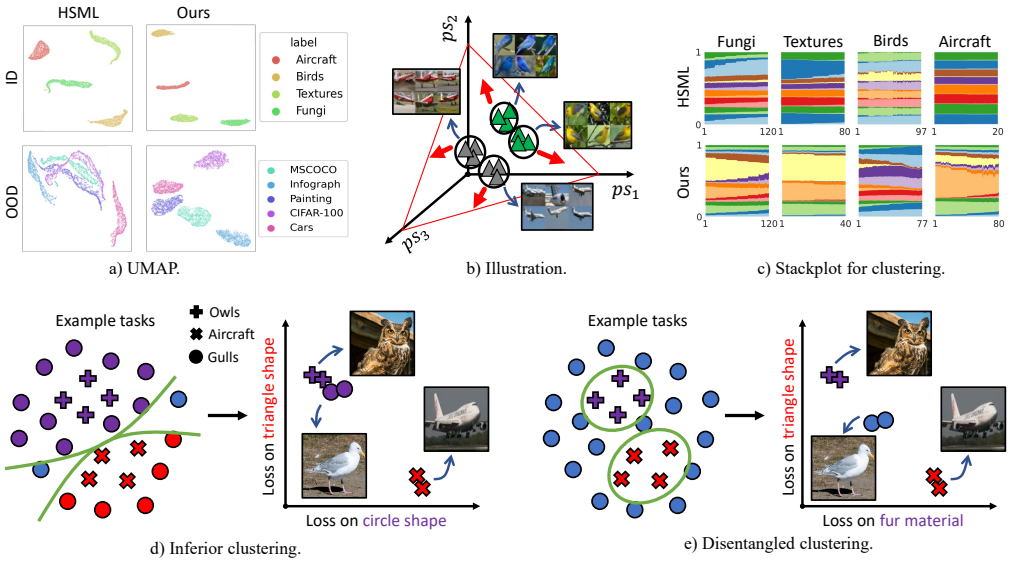
ABSTRACT

To improve meta-generalization, i.e., accommodating out-of-domain meta-testing tasks beyond meta-training ones, is of significance to extending the success of meta-learning beyond standard benchmarks. Previous heterogeneous meta-learning algorithms have shown that tailoring the global meta-knowledge by the learned clusters during meta-training promotes better meta-generalization to novel meta-testing tasks. Inspired by this, we propose a novel multi-objective perspective to sharpen the compositionality of the meta-trained clusters, through which we have empirically validated that the meta-generalization further improves. Grounded on the hierarchically structured meta-learning framework, we formulate a hypervolume loss to evaluate the degree of conflict between multiple cluster-conditioned parameters in the two-dimensional loss space over two randomly chosen tasks belonging to two clusters and two mixed tasks imitating out-of-domain tasks. Experimental results on more than 16 few-shot image classification datasets show not only improved performance on out-of-domain meta-testing datasets but also better clusters in visualization.

1 INTRODUCTION

Meta-learning (Hospedales et al., 2020) has been a very active and burgeoning area of research, with an eye toward human-level intelligence that learns from prior experience (i.e., *meta-training tasks*) to quickly adapt to novel tasks (i.e., *meta-testing tasks*) with minimal supervision. Solutions in literature have pursued three major categories of methods, including metric-based (Cao et al., 2020; Snell et al., 2017; Vinyals et al., 2016), model-based (Mishra et al., 2018; Ravi & Larochelle, 2017), and optimization-based (Antoniou et al., 2019; Finn et al., 2017; Li et al., 2017; Nichol et al., 2018; Rusu et al., 2019; Song et al., 2020) , which learn the transferable meta-knowledge of a metric space, a feed-forward model, and an initialization or optimizer, respectively. The globally shared meta-knowledge, unfortunately, is far from adequate to accommodate a heterogeneous and growing assortment of tasks in the wild (Yao et al., 2019; Yu et al., 2020): (1) meta-training tasks themselves belong to multiple clusters, i.e., being *heterogeneous*; (2) meta-testing tasks are likely out-of-domain (OOD) of meta-training tasks, giving rise to novel and *growing* clusters.

The quest for improving meta-generalization from meta-training to meta-testing tasks under the two real-world scenarios drives the following three strands of works. The first line tackles task heterogeneity either by taking an ensemble of multiple base learners (Dvornik et al., 2020) or by adapting the feature extractor via task-conditioning (Triantafillou et al., 2021; Liu et al., 2021); unfortunately, they require a priori the cluster a task belongs to, which is often inaccessible beyond benchmark datasets. The second line avoids the use of such prior knowledge by directly applying task-specific conditioning onto the base learner (Lee et al., 2020; Lee & Choi, 2018; Oreshkin et al., 2018; Rusu et al., 2019; Vuorio et al., 2019; Wang et al., 2020; Yoon et al., 2018; 2019), unfavorably sacrificing the meta-generalization among a cluster of closely related tasks (Yao et al., 2019). Thus, ours are in line with the third group of works (Yao et al., 2019; 2020) which automatically learns the underlying clusters of meta-training tasks and performs cluster-specific conditioning instead. Specifically, the state-of-the-art HSML (Yao et al., 2019) trains a hierarchical clustering network to obtain a C-dimensional clustering probability score in a C-simplex (e.g., the 3-simplex in Figure 1b), so that the cluster-conditioned initialization for the base learner achieves minimal losses on query sets of this cluster of tasks. Notwithstanding outstanding clustering of in-domain (ID) meta-training tasks as shown in Figure 1a by visualizing the clustering probability scores, HSML struggles to



a) UMAP. b) Illustration. c) Stackplot for clustering.

d) Inferior clustering. e) Disentangled clustering.

Figure 1: Illustration of the core contribution and idea of our method. a) shows the UMAP visualization for clustering probability scores of both ID and OOD meta-testing tasks. Although HSML learns outstanding clustering of ID meta-training tasks, it struggles to differentiate between OOD tasks, while ours succeeds on both ID and OOD tasks. b) illustrates the objective of our method to learn disentangled clusters in the clustering probability score space. c) shows the stackplots of 16 clusters of clustering probability scores for 4 groups of sampled tasks by HSML and ours, respectively. The learned clusters by HSML share overlap within each group, while ours learns more disentangled clustering. Note that the meta-training tasks in a) and c) are sampled from Aircraft, Birds, Textures and Fungi. d) and e) show an intuitive comparison of inferior and disentangled clustering in both feature and loss spaces, where an OOD task of bird recognition is confused/differentiated with the task of owl recognition under disentangled/inferior clustering.

differentiate between OOD tasks and thereby results in poor meta-generalization performance on OOD meta-testing tasks. This is largely attributed to the homogeneity of cluster centers that share considerable overlap, evidenced in almost identical probability scores for 16 clusters across all 4 groups of tasks in Figure 1c, though a subtle difference in clustering probability scores is sufficient for HSML tell them apart. Figure 1d provides a perspective in the feature and loss space, where the two largely overlapping cluster centers (i.e., circle shape and triangle shape) suffice to push the losses of the base learner conditioned by one cluster (i.e., owl) on these two clusters of tasks far away from those conditioned by the other cluster (i.e., aircraft). Undesirably, a novel cluster of OOD tasks (i.e., gull) is likely close to some of the clusters learned during meta-training (i.e., owl).

Taking inspiration from these preliminary experiments (i.e., Figure 1a, c) and also the significance of compositionality (Russin et al., 2020; 2019) for promoting the generalization, we seek a solution that maximally disentangles the cluster centers so that clustering probability scores vary significantly from cluster to cluster (see Figure 1b). This is, however, non-trivial provided that tasks are trained in a batch-wise manner in meta-learning. Concretely, we for the first time formulate task clustering enabled meta-learning as a multi-objective optimization problem, theoretically supported by (Jin & Sendhoff, 2008) which states that empirical losses on different subsets of data can be regarded as a multi-objective point of view. In this multi-objective formulation, each objective is the loss of the base learner with respect to a cluster of tasks; for example, the two objectives in the loss space of Figure 1e are losses on tasks belonging to the clusters of fur material and triangle shape, respectively. The base learner conditioned by different clusters will obviously lead to a handful of points in the loss space; disentangling the clusters boils down to enforcing a diverse distribution of these points, opposed to inferior clustering in Figure 1d. Therefore, we obtain these points by conditioning the base learner with both (1) tasks affiliated to varying clusters of tasks and (2) OOD clusters mimicked by mixup of ID tasks, and regularize meta-training with a hypervolume loss maximizing which encourages

a diversified distribution of these points. Figure 1a) verifies the effectiveness of the proposed loss, where ours successfully differentiates between OOD tasks with more disentanglement of the cluster centers (see Figure 1c).

We summarize our main contributions as follows.

- To the best of our knowledge, we are the first to improve meta-generalization from a multi-objective perspective, through which we propose a novel regularizer and obtain multiple disentangled clusters.
- We evaluate the meta-model empowered by the proposed disentangled task clustering on comprehensive OOD datasets including different grains (i.e., fine-grained and coarse-grained) and styles (e.g., real, draw, infograph). The experimental results show that our model significantly outperforms other state-of-the-art approaches on both ID and OOD meta-testing accuracy.

2 Related Works

Meta-learning for task heterogeneity. Prior approaches that deal with task heterogeneity can be broadly divided into three categories. First, specified with the ground-truth cluster that a task belongs to, SUR (Dvornik et al., 2020) trains multiple base learners, each of which targets a cluster of tasks, while the works (Liu et al., 2021; Suo et al., 2020; Triantafillou et al., 2021) learns a feature extractor comprised of shared universal parameters and task-specific parameters, to achieve single-network multi-domain representation for tasks. To alleviate the cluster information which is usually inaccessible, the second line directly modulates a generalized model to be task-specific, via learning a binary mask (Lee & Choi, 2018), a mapping to meta-training tasks (Wang et al., 2020), and a task embedding (Lee et al., 2020; Oreshkin et al., 2018; Rusu et al., 2019; Vuorio et al., 2019), respectively. Although these task-specific conditioning methods are powerful for knowledge customization, it fails to take the underlying clusters of tasks into consideration, which further boosts the meta-generalization to OOD tasks (Yao et al., 2019). Thus, the third line resorts to an additional network to learn the tree-based (Yao et al., 2019) or graph-based cluster (Yao et al., 2019) assignment of a task and modulate the base learner with the assigned cluster. Another algorithm of TSA-MAML (Zhou et al., 2021) learns how to cluster tasks via applying k-means on the model parameter space, while it trains multiple individual models for each cluster, which is very time-consuming and memory-inefficient. Moreover, as we have illustrated in the Introduction, none of these methods explicitly regularizes the distribution of cluster centers, which however is crucial to meta-generalization.

Multi-objective optimization for machine learning. Multi-objective optimization has been frequent for tackling multi-task learning, where each task presents one objective (e.g., robustness, model complexity, mean squared loss, etc. (Jin & Sendhoff, 2008)) to be solved. There have been a multitude of gradient-based multi-objective optimization methods attempting to improve the diversity and Pareto optimality of the solutions, including MGDA (Sener & Koltun, 2018), EPO (Mahapatra & Rajan, 2020), HV Maximization (Deist et al., 2021), ParetoMTL (Lin et al., 2019). A recent study(Ye et al., 2021) simultaneously optimize the two objectives of meta-training loss and robustness in meta-learning. However, our focus of improving meta-generalization in meta-learning is significantly different from prior attempts. To the best of our knowledge, our work is the first to formulate the losses on different clusters of tasks as multiple objectives, based on which we pursue disentangled clusters that will contribute to diverse solutions in the objective space and thereby improved generalization.

3 Preliminaries

In this paper, we focus on *meta-learning* to address the commonly studied K-shot N-way learning tasks, where we assume a set of N^{tr} meta-training tasks $\{T_i\}_{i=1}^{N^{tr}}$ sampled from a task distribution $p(\mathcal{T})$. Each i-th task $T_i = \{\mathcal{D}_i^s, \mathcal{D}_i^q\}$ consists of a support set $\mathcal{D}_i^s = \{\mathbf{x}_{ij}, y_{ij}\}_{j=1}^{NK}$ and a query set $\mathcal{D}_i^q = \{\mathbf{x}_{ij}, y_{ij}\}_{j=1}^{n^q}$, with NK support examples and n^q query examples, respectively. In the meta-training phase, the meta-knowledge is learned from the meta-training tasks; afterwards, the meta-knowledge facilitates the learning of a meta-testing task $T_t = \{\mathcal{D}_t^s, \mathcal{D}_t^q\}$ in the meta-testing phase.

MAML (Finn et al., 2017) as the pioneering optimization-based meta-learning algorithm meta-learns a well-generalized initialization θ of the base learner as the meta-knowledge, so that θ quickly solves a task $\mathcal{T}_i$ within only a few gradient steps. Concretely, the adaptation in each task follows $\phi_i = \theta - \alpha \nabla_\theta \mathcal{L}(\theta; \mathcal{D}_i^s)$, where without loss of generality we illustrate with only single gradient

step. The initialization θ, in turn, is therefore optimized by evaluating the loss of all adapted models $\{\phi_i\}_{i=1}^{N^{tr}}$ over all query sets, i.e., $\theta = \theta - \beta \nabla_\theta \sum_{i=1}^{N^{tr}} \mathcal{L}(\phi_i; \mathcal{D}_i^q)$. α, β denote the learning rates for adaptation within a task and optimization of θ, respectively.

In real-world applications, it is frequent to have a heterogeneous task distribution $\{p_c(\mathcal{T})\}_{c=1}^{C}$ composed of C diverse domains, e.g., bird classification and dog classification, where $p_c(\mathcal{T})$ denotes the c-th domain of tasks. MAML with a single initialization θ, unfortunately, struggles to accommodate various domains of tasks, according to the "no-free-lunch" theory.

HSML (Yao et al., 2019), one of the state-of-the-art approaches to addressing task heterogeneity, learns a clustering network to perform hierarchical clustering of tasks, besides the global initialization θ. As shown in Figure 2, the clustering network conditions on the representation of a task, i.e., $\mathbf{g}_i = \frac{1}{NK} \sum_{j=1}^{NK} \mathcal{F}(\mathbf{x}_{ij}, y_{ij})$, where $\mathcal{F}(\cdot)$ is the image-level embedding function. Besides the mean pooling over embeddings of NK support examples, a recurrent autoencoder aggregator also applies to arrive $\mathbf{g}_i$. Upon the representation $\mathbf{g}_i$ of the i-th task, the clustering network learns N_k clustering centers $\{\mathbf{a}_k\}_{k=1}^{N_k}$ and outputs the probability of the task belonging to the k-th cluster as $p_i^k = \frac{\exp\left(-||(\mathbf{h}_i - \mathbf{a}_k)/\sigma||_2^2/2\right)}{\sum_{k'=1}^{N_k} \exp\left(-||(\mathbf{h}_i - \mathbf{a}_k)/\sigma||_2^2/2\right)}$. Note that $\boldsymbol{p_i} = [p_i^1, \cdots, p_i^{N_k}] \in \Delta_{N_k}$ and $\sum_{k=1}^{N_k} p_i^k = 1$ always holds. The cluster-specific representation $\mathbf{h}_i = \sum_{k=1}^{N_k} p_i^k \tanh\left(\mathbf{W}^k \mathbf{g}_i + \mathbf{b}^k\right)$, therefore, modulates the global initialization θ to be the cluster-specific one, i.e., $\theta_i = \theta \circ \text{FC}_{W_g}^{\sigma}(\mathbf{h}_i)$. $\text{FC}_{\mathbf{W}_g}^{\sigma}$ denotes a fully-connected layer parameterized by $\mathbf{W}_g$ and activated by σ. Here we illustrate a single level of clustering, though HSML allows multiple levels by recursively using $\mathbf{h}_i$ as the input for the next level of clustering. HSML trains the clustering network, the image-level embedding function $\mathcal{F}$ as well as the initialization θ via the following objective,

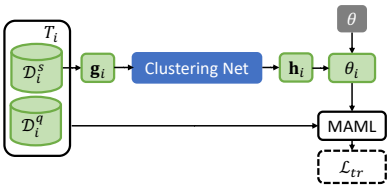


Figure 2: Brief illustration of the HSML framework.

$$\mathcal{L}_{tr} = \mathcal{L}(\theta_i - \alpha \nabla_{\theta_i} \mathcal{L}(\theta_i; \mathcal{D}_i^s); \mathcal{D}_i^q) + \xi \mathcal{L}_r(\mathcal{D}_i^s), \tag{1}$$

where $\mathcal{L}_r$ is the reconstruction loss. For more details, please kindly refer to (Yao et al., 2019).

4 Proposed Framework

The proposed framework as shown in Figure 3 sets out to push ahead with learning as disentangled clusters as possible from meta-training tasks, so that the initialization modulated by the composition of them well generalizes to OOD tasks. Grounded on HSML that empowers task clustering, the proposed framework formulates meta-training as a multi-objective optimization problem which will be introduced in Section 4.1. Such a formulation lays the foundation for our major proposal, i.e., the conflict loss that regularizes the clusters to be as distinct as possible. Section 4.2 and Section 4.3 will introduce the two components of pool construction and task sampling, which prepare for computing the conflict loss. Section 4.4 concludes with the detailed derivation of the conflict loss as well as the overall meta-training objective.

4.1 Multi-objective Formulation of Meta-learning with Heterogeneous Tasks

Multi-objective query losses. The vanilla MAML as stated above assumes N^{tr} meta-training tasks all sampled from a homogeneous task distribution $p(\mathcal{T})$, which justifies a naive average of all query losses on the query sets of all tasks, i.e., $\mathcal{L}(\theta) = \sum_{i=1}^{N^{tr}} \mathcal{L}(\phi_i; \mathcal{D}_i^q)$. Provided with a heterogeneous task distribution $\{p_c(\mathcal{T})\}_{c=1}^{C}$, however, the average query loss over all meta-training tasks raises a major issue that all C domains are treated equally without discrimination; as long as the average loss keeps decreasing, poor performance on some challenging domains is ignored. To this end, we propose the multi-objective formulation $\mathcal{L}^{(mo)}(\theta) \in \mathbb{R}^{N_u}$ which evaluates the losses of θ with respect to different clusters of tasks, i.e.,

$$\mathcal{L}^{(mo)}(\theta) = [\mathcal{L}^1(\theta), \ldots, \mathcal{L}^{N_u}(\theta)]^\top = [\sum_{T_i \in \mathcal{U}^1} \mathcal{L}(\phi_i; \mathcal{D}_i^q), \cdots, \sum_{T_i \in \mathcal{U}^{N_u}} \mathcal{L}(\phi_i; \mathcal{D}_i^q)]^\top. \tag{2}$$

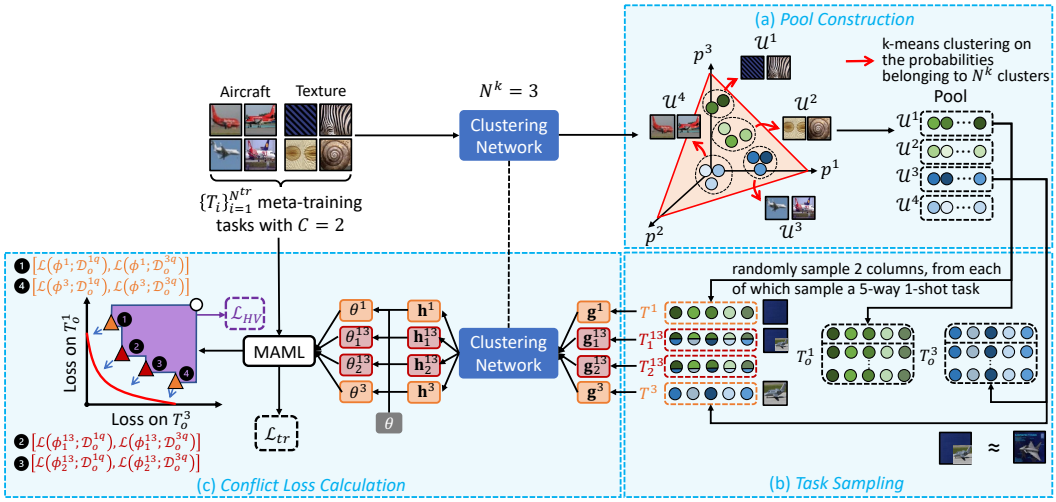


Figure 3: Illustration of the proposed framework that consists of three major components. (a) **Pool construction**: in the N_k-dimensional simplex by the clustering network, we perform k-means clustering (with N_u clusters) on the probabilities of all images, and construct a pool of N_u (4 in the figure) columns each of which stores images belonging to it. The color represents domains (e.g., blue for aircraft) and the color degree denotes class labels (e.g., dark blue for Boeing 777-300er). (b) **Task sampling**: we randomly sample two columns from the pool in each iteration, and from each column sample a pure task with only 5-way 1-shot support examples (T^1,T^3 in the figure) and an objective task with both support and query examples (T_o^1,T_o^3 in the figure). Besides, we generate mixed tasks that mimic OOD tasks (e.g., the mixed example by CutMix is similar to one real example from the domain of Infograph). (c) **Conflict loss calculation**: we obtain four initializations that are specific to two pure ID columns and two OOD tasks, i.e., θ^1, θ^3, θ_1^{13}, θ_2^{13}, respectively. By evaluating their losses on the two objective tasks T_o^1 and T_o^3, we calculate the conflict loss (namely the hypervolume loss $\mathcal{L}_{HV}$) to regularize meta-training with more disentangled clusters.

The reasons why we evaluate on N_u clusters instead of C domains mentioned above are as follows. First, C remains inaccessible in most of real-world cases, where tasks arrive in sequentially without a domain label. Secondly, a subjectively defined domain does not necessarily represent a single cluster of tasks; for example, the domain of Aircraft contains airbuses and helicopters. In Section 4.2, we will detail how to obtain N_u clusters of tasks, i.e., $\mathcal{U}^1, \cdots, \mathcal{U}^{N_u}$.

Note that this idea of multi-objective formulation is inspired from (Jin & Sendhoff, 2008) which uses losses on subsets of data in conventional machine learning as multiple objectives. Despite the difficulty in choosing subsets from a dataset in (Jin & Sendhoff, 2008), it is highly intuitive and straightforward to approach task heterogeneity in meta-learning by formulating multiple losses on tasks from different distributions as multiple objectives.

Multi-objective query loss matrix. In fact, our framework established on HSML allows cluster-specific initializations $\{\theta^u\}_{u=1}^{N_u}$, by modulating the global initialization θ with the representation of a task belonging to the u-th cluster, i.e., T^u. Assuming the availability of multiple tasks from N_{ood} OOD domains $\{T^u\}_{u=N_u+1}^{N_u+N_{ood}}$ during meta-training, we similarly obtain their cluster-specific initialization $\{\theta^u\}_{u=N_u+1}^{N_u+N_{ood}}$. Consequently, we obtain a multi-objective query loss matrix $\mathcal{L}^{(mo)}(\Theta) = [\mathcal{L}^{(mo)}(\theta^1), \ldots, \mathcal{L}^{(mo)}(\theta^{N_u+N_{ood}})]^\top \in \mathbb{R}^{(N_u+N_{ood})\times N_u}$, according to Eqn (2).

The multi-objective query loss matrix serves the cornerstone for calculating the conflict loss in Section 4.4, where the key insight lies in that more diverse distribution of $N_u + N_{ood}$ losses in the N_u-dimensional space promotes the separation of N_u clusters (*a.k.a.* disentangled clusters). In practice, unfortunately, we do not have access to tasks from OOD domains during meta-training; therefore, we will introduce how to construct mixed tasks to mimic the tasks from OOD domains in Section 4.3. Another advantage of such a multi-objective formulation is to make visualization of the model $\{\theta_u\}_{u=1}^{N_u}$ by the state-of-the-art manifold representing methods (e.g., t-SNE (Van der Maaten & Hinton, 2008), UMAP (McInnes et al., 2018)) more robust. Rather than using the original θ_u in a very high dimensional space, representing θ_u with N_u query losses greatly reduces the dimension.

4.2 CLUSTERING POOL CONSTRUCTION

Our method resorts to the multi-objective loss matrix to assist meta-training, which requires some tasks for generating and evaluating Θ. These tasks should be able to thoroughly represent learned clusters under the current clustering strategy (i.e., θ^i achieves smallest/best query losses on tasks in cluster $\mathcal{U}^i$, while having higher/worse query losses on tasks in $\mathcal{U}^j, j \neq i$, as illustrated in Figure 4a and b). On the other hand, the soft-assignment of the task enables the model to cluster this task into multiple clusters. As a result, it is sometimes difficult to encounter a meta-training task, one hundred percent belonging to one specific cluster, since tasks usually consist of multiple images with different labels.

In order to efficiently construct tasks that well-represent one cluster, we propose to store images in the pool rather than tasks. To counter the effect of the label shift, for an image x, we construct an auxiliary task $\mathcal{T}_x$, with only the support set $\mathcal{D}^s_{\mathcal{T}_x} = \{(x, y_s)\}_{s=1}^{NK}$. We obtain the embedded representation h_x and the probability score $\boldsymbol{p_x}$ for image x. K-means clustering is performed on $\boldsymbol{p_x}$s for historical images and pool maintains N_u clusters $\{\mathcal{U}^u\}_{u=1}^{N_u}$ of historical images, according to the distances between $\boldsymbol{p_x}$ to k-means clustering centers. Moreover, we re-obtain $\boldsymbol{p_x}$ for outdated images in the pool every a period of epochs to ensure the pool is up-to-date, which is the guarantee of representing the current clustering strategy.

4.3 TASK SAMPLING FROM POOL

Given the pool, which represents the currently learned clustering, we can sample tasks from the pool for conflict loss calculation. In order to efficiently enhance pair-wise cluster difference and balance computational resources, we randomly select m clusters in the pool for each meta-training iteration ($\mathcal{U}^1$,$\mathcal{U}^3$ as in Figure 3).

Pure tasks T^us are used to generate θ^us that represent the corresponding clusters $\mathcal{U}^u$. Specifically from a selected cluster $\mathcal{U}^u$, we randomly sample $N \times K$ images with N different labels (K images for each label) to construct task T^u with only the support set. The labels of all selected images are changed to a relative manner $\{y_s\}_{s=1}^{NK}$. Examples of T^1, T^3 are shown in Figure 3.

Mixed tasks T^{uv}s are used to generate θ^{uv}s, that mimic the cluster-specific initializations for OOD tasks. Firstly, we generate two new tasks $\tilde{T}^u, \tilde{T}^v$ with support sets $\{(\tilde{x}^u_s, y_s)\}_{s=1}^{NK}, \{(\tilde{x}^v_s, y_s)\}_{s=1}^{NK}$ sampled from two selected clusters $\mathcal{U}^u$,$\mathcal{U}^v$, respectively. Next, we perform CutMix (Yun et al., 2019) augmentation on the corresponding images from both support sets, while remaining the labels to be still relative. In details, $\mathcal{T}^{uv}$ consists of $\{(\tilde{x}_s, y_s)\}_{s=1}^{NK}$. The mixed image $\tilde{x}_s = M\tilde{x}^u_s + (\mathbb{1} - M)\tilde{x}^v_s$, where M is a binary mask with fixed bounding box width and height indicating foreground or background rectangular regions and $\mathbb{1}$ is an ones matrix with the same shape as images. A random variable $\lambda \sim Beta(a, b)$ determines whether M is the foreground ($\lambda \geq 0.5$) or background ($\lambda < 0.5$) for each image. Examples of T^{13}_1, T^{13}_2 are shown in Figure 3.

Objective tasks T^u_os are used to evaluate pure and mixed tasks. For T^u_os, we use the same strategy as generating pure tasks, but also sample images for query sets to obtain query losses. Examples of T^1_o, T^3_o are shown in Figure 3.

4.4 CONFLICT LOSS CALCULATION

Given sampled $\{T^p, T^{uv}\} = \{T_i\}_{i=1}^{N_i}$ and $\{T^u_o\}_{u=1}^{N_u}$s, we obtain **empirical** multi-objective query loss matrix and further calculate hypervolume loss. Note that in Figure 3c, $N_i = 4, N_u = 2$. The clustering network does clustering on T_is and modulates global initialization θ to cluster-specific initialization $\Theta = \{\theta_i\}_{i=1}^{N_i}$, respectively. During this process, we detach θ from the computational graph for gradient, that the conflict loss is only used to update the clustering network but not θ. For each θ_i, the empirical multi-objective query losses $\mathcal{L}^{(mo)}_o(T_i) \in \mathbb{R}^{N_u}$ evaluated on T^u_os by the base learner are as follows:

$$\mathcal{L}^{(mo)}_o(T_i) = [\mathcal{L}(\phi_1; \mathcal{D}^q_{T^1_o}), \ldots, \mathcal{L}(\phi_{N_u}; \mathcal{D}^q_{T^{N_u}_o})]^\top, \tag{3}$$

where ϕ_i is the corresponding adapted parameters for $\mathcal{D}^s_{T^u_o}$ after the inner loop. The corresponding empirical multi-objective query loss matrix is $\mathcal{L}^{(mo)}_o(\Theta) = [\mathcal{L}^{(mo)}_o(T_1), \ldots, \mathcal{L}^{(mo)}_o(T_{N_i})]^\top \in \mathbb{R}^{N_i \times N_u}$.

On the one hand, a good diversity of Θ in the multi-objective query loss space shows good clustering. On the other hand, the losses are expected to be minimized, indicating the demand for convergence. In order to measure the quality of the current clustering, we propose a conflict indicator, named hypervolume loss.

Hypervolume loss $\mathcal{L}_{HV}$. Inspired by (Deist et al., 2021), $\mathcal{L}_{HV}$ is the negative hypervolume (Zitzler & Thiele, 1999) value between $\mathcal{L}_o^{(mo)}(\Theta)$ and a reference point $\mathcal{Z} \in \mathbb{R}^m$. Specifically, $\mathcal{L}_{HV}(\mathcal{L}_o^{(mo)}(\Theta), \mathcal{Z}) = -\Lambda\left(\bigcup_{p\in\mathcal{L}_o^{(mo)}(\Theta), p\leq\mathcal{Z}}\{q \in \mathbb{R}^m | p \leq q \leq \mathcal{Z}\}\right)$, where $\Lambda(\cdot)$ denotes the Lebesgue measure. Minimizing $\mathcal{L}_{HV}$ will encourage both convergence (i.e., smaller losses in all axes) and diversity (i.e., a better distribution on pure and mixed tasks in the multi-objective query loss space). We derive the total training loss for the whole framework as $\mathcal{L}_{total} = \mathcal{L}_{tr} + \alpha\mathcal{L}_{HV}$, where $\mathcal{L}_{tr}$ is the training loss, consisting of the query losses on batched training tasks and the reconstruction loss of the task embedding, and α is the weight of $\mathcal{L}_{HV}$ to balance the importance of two items.

5 EXPERIMENTS

In this section, we evaluate our proposed method by comprehensive computational experiments on OOD datasets. we will answer the following questions: (1) Can our method outperform other state-of-the-art approaches, especially on achieve consistently higher OOD meta-testing accuracy on a wide range of OOD datasets? (2) Can our method learn more diverse clusters from meta-training tasks? (3) To what extent does meta-training benefit from multi-objective query loss / conflict loss?

5.1 EXPERIMENT SETTINGS

Datasets. We follow the same few-shot classification datasets as in HSML (Yao et al., 2019) for meta-training. They are *Birds* (Wah et al., 2011), *Textures* (Cimpoi et al., 2014), *Aircraft* (Maji et al., 2013), *Fungi* (Kaggle, 2018), which are now parts of Meta-Dataset (Triantafillou et al., 2019). As for meta-testing, we investigate a comprehensive range of OOD datasets including other datasets in **Meta-Dataset** (except Quickdraw (Jongejan et al., 2016), the one also in DomainNet (Peng et al., 2019)): *Mini* (Vinyals et al., 2016) (mini-ImageNet, as a substitution of ILSVRC (Russakovsky et al., 2015)), *Omniglot* (Lake et al., 2015), *VGG Flower* (Nilsback & Zisserman, 2008), *Traffic Signs* (Houben et al., 2013), *MSCOCO* (Lin et al., 2014); **DomainNet** (Peng et al., 2019): *Clipart*, *Infograph*, *Painting*, *Quickdraw*, *Real*, *Sketch*; *CIFAR-100* (Krizhevsky et al., 2009); Stanford *Cars* (Krause et al., 2013); Oxford-IIIT *Pets* (Parkhi et al., 2012); Stanford *Dogs* (Khosla et al., 2011). See short descriptions of all datasets in Appendix. For the sake of uniformity, all images are resized to the same shape (i.e., 84×84 resolution with RGB channels). All tasks are organized in the commonly used 5-way 1-shot episodic fashion with relative labels (i.e., from "1" to "5"), unless stated otherwise.

Implementation Details. We follow the same network structure as described in (Yao et al., 2019) but increase the number of nodes in hierarchical task clustering structure to $4, 4, 1$, to accommodate larger clustering capacity on OOD meta-testing tasks. The pool consists of 16 clusters, each with a maximum of 20 classes. For each class, we store 16 images (1 support sample and 15 query samples) with the closest distances to the corresponding k-means center. The numbers of pure, mixed, and objective tasks are all set to 2. Random variables λs for generating two mixed tasks are from $Beta(5, 2)$ and $Beta(2, 5)$, respectively. We normalize multi-objective query loss matrix $\mathcal{L}_o(\Theta)$ to $[0, 1]^{4\times2}$, then set the reference point $\mathcal{Z}$ to $[1.5, 1.5]^\top$ The weight of hypervolume loss α is set to 0.1. See detailed hyper-parameter settings in Appendix.

Compared Algorithms. We compare our method with some state-of-the-art algorithms which do not utilize prior task clusters, specifically in the following three baselines: 1) Globally shared models: **MAML** (Finn et al., 2017); 2) Task-specific conditioned models: **Bayesian-TAML** (Lee et al., 2020); 3) clustering-based models: **HSML** (Yao et al., 2019), **ARML** (Yao et al., 2020), and **Simple-CNAPs** (Bateni et al., 2020).

5.2 COMPARISON RESULTS AND ANALYSIS

The meta-training of a model consists of 20 epochs with 3000 iterations in each epoch. We evaluate the performance of the model on meta-validation sets after the end of each epoch. Specifically, we

Table 1: 5-way 1-shot meta-testing accuracy (%) comparison to SOTA algorithms meta-trained with Aircraft, Birds, Textures, and Fungi. Ours-Aug is a variant ablating the hypervolume loss. Accuracy (standard deviation) over 10 independent runs are reported.

Test Dataset	MAML	Bayesian-TAML	ARML	Simple-CNAPs	HSML	Ours-Aug	Ours
Aircraft	52.40(0.32)	45.86(0.63)	56.77(0.71)	49.60(0.60)	**58.26(0.24)**	57.34(0.31)	57.12(0.20)
Birds	55.66(0.40)	55.80(0.69)	58.90(0.88)	54.80(0.70)	63.20(0.41)	**63.30(0.36)**	62.94(0.39)
Textures	31.40(0.27)	30.55(0.49)	33.14(0.52)	31.70(0.50)	**35.29(0.13)**	34.61(0.30)	34.20(0.26)
Fungi	41.27(0.30)	39.77(0.65)	43.70(0.89)	40.60(0.70)	**46.01(0.29)**	45.48(0.31)	44.50(0.33)
ID Average	45.18	43.00	48.13	43.98	**50.69**	50.18	49.69
Mini	34.59(0.39)	35.31(0.55)	34.49(0.42)	31.90(0.50)	36.62(0.27)	36.50(0.30)	**38.92(0.32)**
Omniglot	67.54(0.27)	74.10(0.68)	68.26(0.56)	58.70(0.70)	74.03(0.32)	71.48(0.38)	**80.12(0.28)**
VGG Flower	61.72(0.42)	66.12(0.67)	60.15(0.50)	58.70(0.70)	67.14(0.38)	67.53(0.48)	**69.74(0.33)**
Traffic Signs	47.09(0.35)	**50.87(0.70)**	45.04(0.71)	44.30(0.70)	47.53(0.26)	45.06(0.36)	48.52(0.33)
MSCOCO	32.16(0.35)	32.99(0.58)	31.23(0.97)	29.50(0.40)	33.37(0.30)	33.34(0.36)	**34.72(0.40)**
Clipart	34.62(0.25)	37.27(0.60)	33.98(0.85)	31.70(0.50)	36.38(0.27)	36.93(0.21)	**39.18(0.34)**
Infograph	24.64(0.17)	25.98(0.39)	24.56(0.43)	23.70(0.30)	25.39(0.18)	25.19(0.23)	**26.85(0.25)**
Painting	30.80(0.18)	32.69(0.55)	30.26(0.70)	28.60(0.40)	32.46(0.28)	32.36(0.24)	**34.81(0.33)**
Quickdraw	44.42(0.33)	47.08(0.66)	49.24(0.96)	42.70(0.60)	51.81(0.33)	51.57(0.32)	**53.32(0.21)**
Real	38.94(0.11)	41.37(0.64)	38.44(0.70)	34.70(0.50)	41.28(0.33)	41.50(0.30)	**44.51(0.37)**
Sketch	28.73(0.25)	29.34(0.45)	29.13(0.60)	27.50(0.40)	30.51(0.23)	29.91(0.27)	**31.44(0.30)**
CIFAR-100	38.04(0.30)	39.62(0.61)	36.70(0.63)	34.20(0.60)	39.18(0.30)	38.73(0.29)	**41.10(0.31)**
Cars	30.97(0.26)	31.56(0.52)	31.58(0.56)	30.30(0.50)	33.44(0.22)	33.22(0.19)	**34.22(0.29)**
Pets	42.47(0.32)	44.15(0.62)	43.52(0.55)	39.30(0.50)	46.70(0.43)	47.31(0.34)	**48.07(0.37)**
Dogs	34.94(0.28)	36.49(0.54)	35.84(0.46)	31.80(0.50)	39.68(0.35)	39.05(0.34)	**40.83(0.27)**
OOD Average	39.06	41.07	38.97	36.01	41.75	41.36	**43.76**

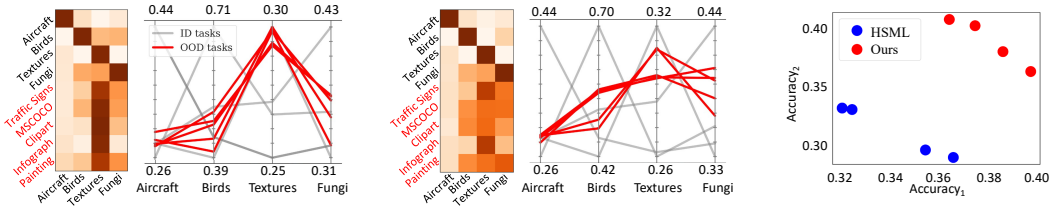
Table 2: 5-way 5-shot OOD meta-testing accuracy (%) comparison with HSML meta-trained with Aircraft, Birds, Textures, and Fungi. Accuracy (standard deviation) are reported.

	MAML	Bayesian-TAML	Simple-CNAPs	HSML	Ours
Mini	49.87(0.51)	51.60(0.55)	46.30(0.50)	52.50(0.55)	**54.03(0.56)**
Omniglot	91.87(0.31)	**91.99(0.31)**	88.50(0.40)	87.56(0.37)	91.70(0.30)
MSCOCO	44.96(0.60)	45.98(0.64)	40.60(0.50)	45.65(0.61)	**46.58(0.61)**
Infograph	32.04(0.42)	33.89(0.42)	31.10(0.40)	33.20(0.45)	**35.00(0.47)**
Quickdraw	66.37(0.57)	68.44(0.60)	**69.80(0.60)**	67.36(0.57)	67.91(0.60)
Real	58.55(0.58)	61.02(0.56)	55.80(0.60)	60.29(0.60)	**61.98(0.60)**
Sketch	40.56(0.48)	42.70(0.47)	39.50(0.50)	44.37(0.49)	**45.43(0.51)**
Cars	44.04(0.55)	46.82(0.59)	44.30(0.60)	46.93(0.58)	**48.37(0.59)**
Pets	64.28(0.52)	66.68(0.49)	62.50(0.50)	66.54(0.54)	**68.48(0.53)**
OOD Average	54.73	56.57	53.16	56.04	**57.72**

construct a pool with the similar strategy described in subsection 4.2, but with image samples in meta-validation sets. Then, we calculate the hypervolume (with the reference point as $[0, 0]^\top$) of the multi-objective validation accuracy as the evaluation metric. The top-3 models on meta-validation sets are used to infer the meta-testing sets.

We report the average meta-testing accuracy over 1000 tasks for each dataset in Table 1. Ours achieves **consistently** outperforming accuracy on a wide range of OOD datasets (except only on Traffic Signs, the case Bayesian-TAML wins and Ours is the second best) compared with other state-of-the-art algorithms, although suffering a small performance drop on ID meta-testing. In the most successful case (i.e., Omniglot), Ours achieves 6.09% better accuracy than HSML. The above observation clearly exhibits the superiority of meta-generalization.

Testing on 5-shot 5-way tasks. To study the case when the few-shot task has more supervision, we increase the number of support samples in one task. The meta-testing accuracy on OOD datasets are summarized on Table 2. Similarly as the case in Table 1, OOD meta-testing shows consistent improvement (i.e., 57.72%), compared with the baselines, except for the cases on Omniglot (Bayesian-TAML wins) and Quickdraw (Simple-CNAPs wins) datasets. Compared with HSML specifically, Omniglot achieves the highest improvement (i.e., an average of 91.70% *vs* 87.56%). Unfortunately, the improvement on OOD average over 5-shot 5-way experiments is less than that over 1-shot 5-way experiments (i.e., 1.68% compared with 2.01% in Table 1). This is because the base learner can adapt better to the specific task when more supervision is provided, thus, the gain from meta-knowledge is less.



a) HSML. b) Ours. c) Pool.

Figure 4: Heatmaps and parallel coordinate plots visualization. a) and b) are multi-objective query accuracy matrices over the averages of 1000 meta-testing tasks for each dataset by HSML and our method, respectively. c) shows averaged empirical multi-objective query accuracy matrices over 50 samples on meta-validation pool for the top model of HSML and our method. **Take-away**: ours indeed can recognize different patterns for OOD tasks, rather than simply regarding them as one of the ID clusters as HSML does.

Ablation study: usefulness of hypervolume loss. In order to verify the effect of the proposed multi-objective query loss matrix and hypervolume loss, we compare Ours with a variant donated as **Ours-Aug**, that directly derives meta-training losses for the pure and mixed tasks rather than evaluating them on the objective tasks as a substitution for hypervolume loss. To this end, we sample not only support sets but also query sets for the pure tasks and mixed tasks. One can imagine Ours-Aug as a task augmentation approach. We compare Ours-Aug with HSML and Ours in Table 1 and show that HSML defeats Ours-Aug (i.e., 41.36% accuracy on average compared with HSML's 41.75% accuracy) on OOD meta-testing, which demonstrates that the improvement on OOD datasets is an outcome of our proposed hypervolume loss, rather than task augmentation. We also compare our method with another variant that uses SpectralNet (Yang et al., 2019) as a substitution for the clustering network to promote disentangled clustering. The computational results are shown in Appendix due to the page limit. Our method outperforms this variant on most of OOD datasets, which hints the advanced clustering learned by our method.

Clustering analysis. We show the averaged multi-objective query accuracy matrix over 50 samples on the meta-validation pool for the top model of HSML and Ours in Figure 4c. The pattern for HSML (blue points) matches Figure 1d (i.e., points w.r.t. two mixed tasks are close to the corresponding points w.r.t. two pure tasks on average), which hints the inferior clustering obtained by HSML. For an OOD task, clustering networks might produce a similar cluster-specific initialization as for one of the ID tasks. Bewildered by this, the base learner therefore can not achieve optimal OOD meta-testing accuracy. By contrast, our method (red points) shows not only good convergence (i.e., better accuracy on both two axes), but most importantly, good diversity (i.e., uniform-distributed points w.r.t. pure and mixed tasks) in the multi-objective query accuracy space. OOD tasks are more likely to have different patterns from ID tasks in the probability score space, thus achieving better meta-generalization for the corresponding $\theta^{\mathcal{T}}$. The comparison of HSML and our method on the multi-objective query accuracy space (as shown in Figure 4a and b) further evidences that our method indeed can recognize different patterns for OOD tasks, rather than simply regarding them as one of the ID clusters (as HSML does in Figure 4a). Furthermore from a probability score point of view, we also show the comparison of UMAP visualization for some tasks in Figure 1a. Our method successfully recognizes different patterns for tasks in each OOD dataset.

6 Conclusion

In this paper, we propose to analyse clustering performance for HSML from a **multi-objective** point of view. Based on our observation, we argue that the clustering learned by HSML is inferior. It is sufficient enough for distinguishing ID tasks but not OOD tasks, which limits the meta-generalization performance. Further, we propose to use an empirical multi-objective query loss matrix, which represents the current learned clustering strategy, and the **hypervolume loss** to regularize the distribution of diverse tasks on multi-objective query loss space, therefore, achieving disentangled clustering.

REFERENCES

Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML. In *International Conference on Learning Representations*, 2019.

Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *Computer Vision and Pattern Recognition*, June 2020.

Kaidi Cao, Maria Brbic, and Jure Leskovec. Concept learners for few-shot learning. In *International Conference on Learning Representations*, 2020.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

Timo M Deist, Monika Grewal, Frank JWM Dankers, Tanja Alderliesten, and Peter AN Bosman. Multi-Objective Learning to Predict Pareto Fronts Using Hypervolume Maximization. *arXiv preprint arXiv:2102.04523*, 2021.

Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In *European Conference on Computer Vision*, pp. 769–786. Springer, 2020.

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

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

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.

Yaochu Jin and Bernhard Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(3):397–415, 2008.

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.

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

Hae Beom Lee, Hayeon Lee, Donghyun Na, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. In *International Conference on Learning Representations*, 2020.

Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pp. 2927–2936. PMLR, 2018.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. META-SGD: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

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.

Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. Pareto multi-task learning. In *Thirty-third Conference on Neural Information Processing Systems (NeurIPS)*, pp. 12037–12047, 2019.

Yanbin Liu, Juho Lee, Linchao Zhu, Ling Chen, Humphrey Shi, and Yi Yang. A multi-mode modulator for multi-domain few-shot classification. In *International Conference on Computer Vision*, pp. 8453–8462, 2021.

Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *International Conference on Machine Learning*, pp. 6597–6607. PMLR, 2020.

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

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

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.

Boris N Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, 2018.

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

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *International Conference on Computer Vision*, pp. 1406–1415, 2019.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

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.

Jacob Russin, Jason Jo, Randall O'Reilly, and Yoshua Bengio. Compositional generalization by factorizing alignment and translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 313–327, 2020.

Jake Russin, Jason Jo, Randall C O'Reilly, and Yoshua Bengio. Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv preprint arXiv:1904.09708*, 2019.

A Rusu, D Rao, J Sygnowski, O Vinyals, R Pascanu, S Osindero, and R Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, 2018.

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.

Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

Xingyou Song, Wenbo Gao, Yuxiang Yang, Krzysztof Choromanski, Aldo Pacchiano, and Yunhao Tang. ES-MAML: simple Hessian-free meta learning. In *International Conference on Learning Representations*, 2020.

Qiuling Suo, Jingyuan Chou, Weida Zhong, and Aidong Zhang. Tadanet: Task-adaptive network for graph-enriched meta-learning. In *International Conference on Knowledge Discovery & Data Mining*, pp. 1789–1799, 2020.

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.

Eleni Triantafillou, Hugo Larochelle, Richard Zemel, and Vincent Dumoulin. Learning a universal template for few-shot dataset generalization. In *International Conference on Machine Learning*, pp. 10424–10433. PMLR, 2021.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

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.

Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Multimodal model-agnostic meta-learning via task-aware modulation. In *Advances in Neural Information Processing Systems*, 2019.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Ruohan Wang, Yiannis Demiris, and Carlo Ciliberto. A structured prediction approach for conditional meta-learning. In *Advances in Neural Information Processing Systems*, 2020.

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.

Huaxiu Yao, Xian Wu, Zhiqiang Tao, Yaliang Li, Bolin Ding, Ruirui Li, and Zhenhui Li. Automated relational meta-learning. In *International Conference on Learning Representations*, 2020.

Feiyang Ye, Baijiong Lin, Zhixiong Yue, Pengxin Guo, Qiao Xiao, and Yu Zhang. Multi-Objective Meta Learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pp. 7343–7353, 2018.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *International Conference on Machine Learning*, pp. 7115–7123. PMLR, 2019.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, pp. 1094–1100, 2020.

Sangdoo 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.

Pan Zhou, Yingtian Zou, Xiao-Tong Yuan, Jiashi Feng, Caiming Xiong, and Steven Hoi. Task similarity aware meta learning: Theory-inspired improvement on maml. In *Uncertainty in Artificial Intelligence*, pp. 23–33. PMLR, 2021.

Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

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 3. 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 5.

- **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 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.

Algorithm 1: Meta-training of the Proposed Framework

Require: termination condition T; outer learning rate β; meta batch size B; shot K; way N
Require: cluster number C; pool update period T_u
Require: hypervolume loss weight α; reference point $\mathcal{Z}$
Initialize **pool** $\mathcal{C} = \{\mathcal{C}_c\}_{c=1}^{C} \leftarrow \{\emptyset\}_{c=1}^{C}$
Randomly initialize the **clustering network** and **base learner** $\theta_{all} = \{\theta_{cn}, \theta_{bl}\}$
for $t = 1$ **to** T **do**
  Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{B}$
  Compute $\mathcal{L}_{train}$ by HSML

  /* Clustering Pool Construction */
  if $\mod(t, T_u) == 0$ **then**
    for $c = 1$ **to** C **do**
      for $\hat{x}$ **in** $\mathcal{C}_c$ **do**
        Update $\boldsymbol{p}_{\hat{\boldsymbol{x}}}$ in Equation (1)
      end for
    end for
  end if
  $P \leftarrow \mathcal{C}$
  for each $\mathcal{T}_i$ **do**
    for x **in** $\mathcal{T}_i$ **do**
      Construct auxiliary task $\mathcal{T}_x$ with $\mathcal{D}_{\mathcal{T}_x}^{(s)} \leftarrow \{(x, y_j)\}_{j=1}^{NK}$
      Calculate $\boldsymbol{p}_{\boldsymbol{x}}$ in Equation (1)
      $P \leftarrow P \cup \boldsymbol{p}_{\boldsymbol{x}}$
    end for
  end for
  Apply k-means on P to have $\{\mathcal{C}_c\}_{c=1}^{C} \leftarrow P$

  /* Task Sampling from Pool */
  Sample $\mathcal{C}_i, \mathcal{C}_j$ from $\mathcal{C}$
  Sample $\mathcal{T}_{p_i}, \mathcal{T}_{p_j}, \mathcal{T}_{m_i}, \mathcal{T}_{m_j}, \mathcal{T}_{o_i}, \mathcal{T}_{o_j}$ from $\mathcal{C}_i, \mathcal{C}_j$ in subsection 4.3

  /* Conflict Loss Computation */
  Compute the multi-objective query loss matrix $\mathcal{L}_{mo}$ in Equation (3)
  Compute $\mathcal{L}_{HV}(\mathcal{L}_{mo}, \mathcal{Z})$

  /* Meta Training */
  Compute $\nabla\mathcal{L}_{total} = \nabla_{\Theta}\mathcal{L}_{train} + \alpha\nabla_{\theta_{cn}}\mathcal{L}_{HV}$
  Update $\Theta \leftarrow \Theta - \beta\nabla\mathcal{L}_{total}$
end for

  - **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.
- **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.

Table 3: 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 $n^{(q)}$	75
	Image shape	(84, 84, 3)

- **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 4, 5 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 1, 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.

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 6. 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.

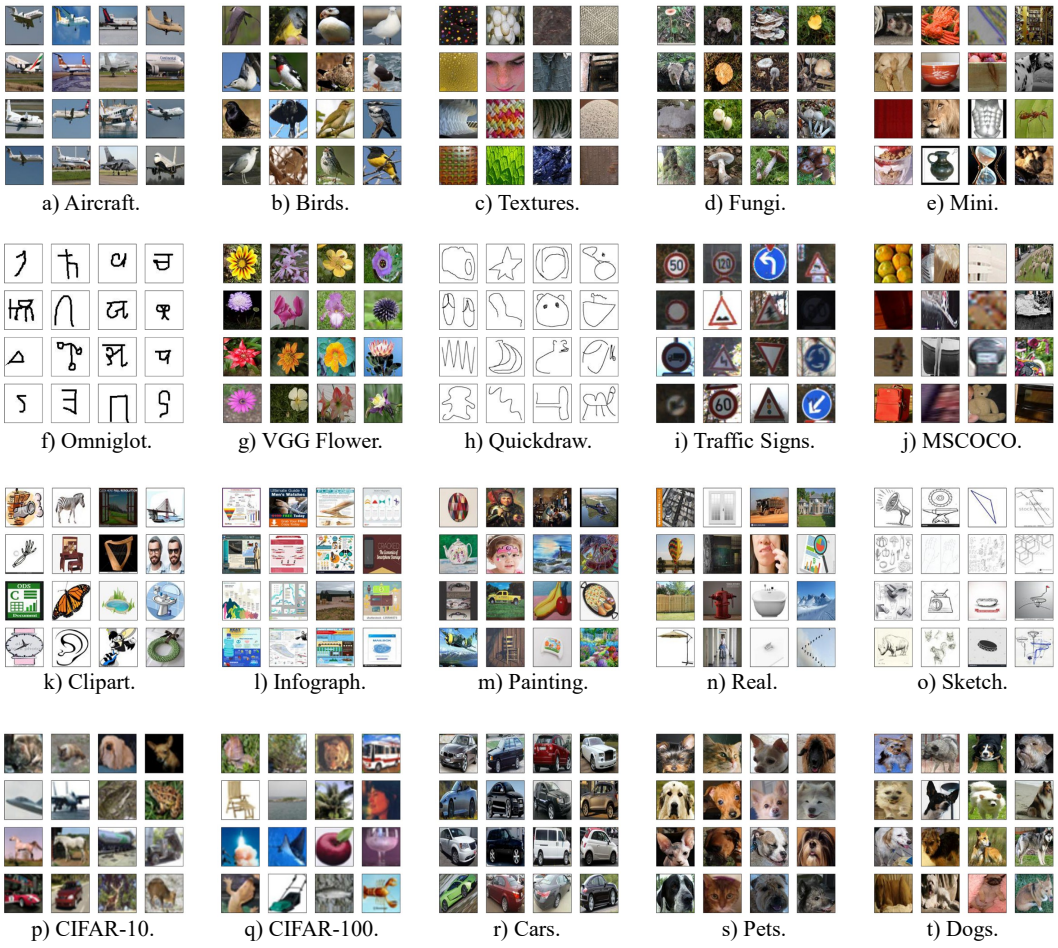


Figure 5: Image examples from all datasets used in the experiments.

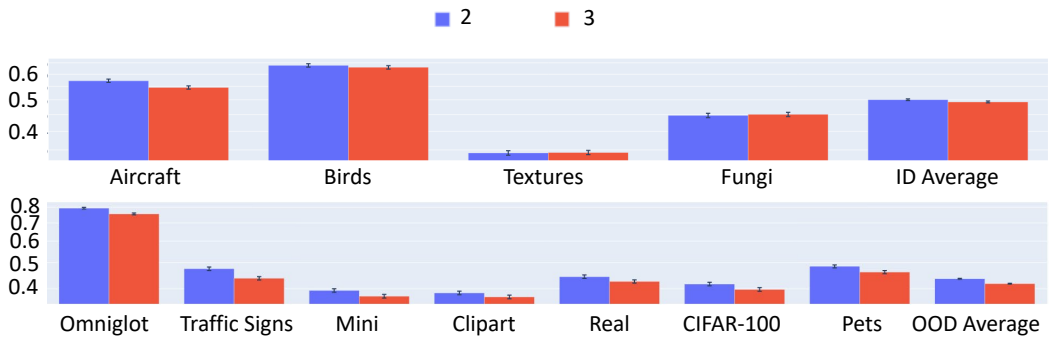


Figure 6: Meta-testing accuracy for varying number of objectives (blue: 2-objective, red: 3-objective) on 1-shot 5-way experiments meta-trained with Aircraft, Birds, Textures, and Fungi datasets. The number of mixed tasks generated in each iteration is set to 2 and 3 for 2-objective and 3-objective cases, respectively.

Table 4: 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 5: 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%**

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 7. 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 6. 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.

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.

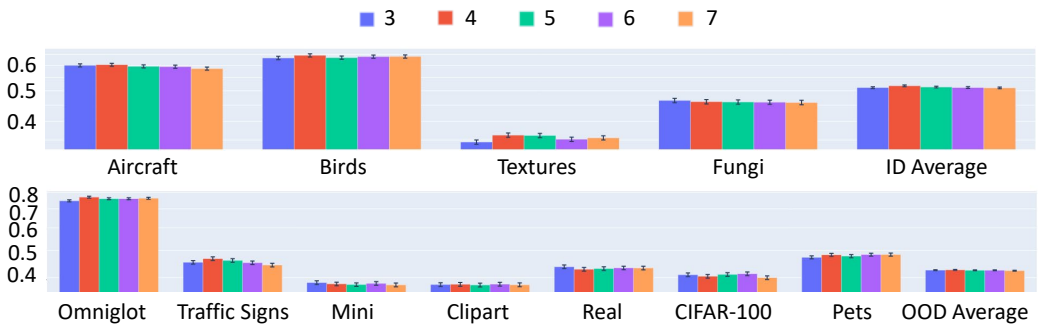


Figure 7: Meta-testing accuracy for varying number of mixed tasks on 1-shot 5-way experiments meta-trained with Aircraft, Birds, Textures, and Fungi datasets. The number of objectives is set to 2.

Table 6: 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%

Table 7: 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%

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 7. 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 8 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).

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 8.

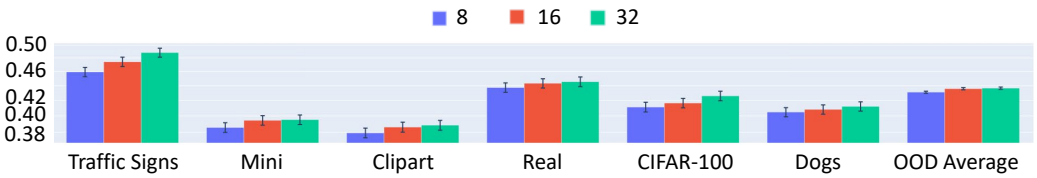
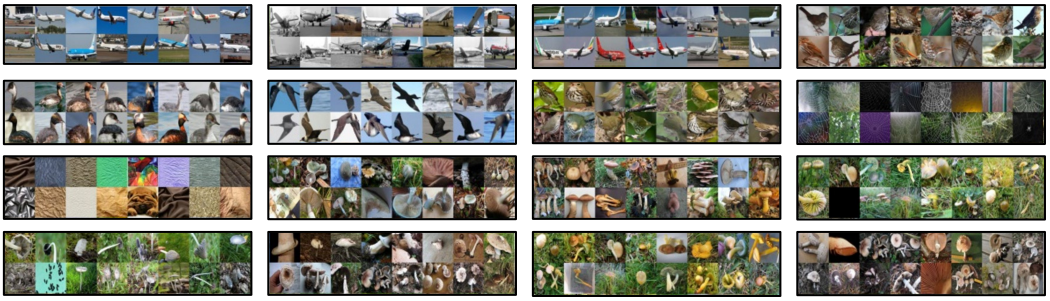


Figure 8: Meta-testing accuracy for different numbers of clusters on 1-shot 5-way experiments meta-trained with Aircraft, Birds, Textures, and Fungi datasets. The clustering network architecture for 8, 16, and 32 are (4, 2, 1), (4, 4, 1), and (8, 4, 1), respectively.

Table 8: 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%**

a) HSML.



b) Ours.

Figure 9: Image examples from learned pool.

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 9. 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.