AFRL-RI-RS-TR-2021-156

DOMAIN-ADAPTIVE ACTIVE META-LEARNING (DAAML)

SRI INTERNATIONAL

SEPTEMBER 2021

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2021-156 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
PETER A. JEDRYSIK
Work Unit Manager

/ S /
JULIE BRICHACEK
Chief, Information Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
SEPTEMBER 2021	FINAL TECHNICAL REPORT	JUL 2019 – APR 2021

4. TITLE AND SUBTITLE

DOMAIN-ADAPTIVE ACTIVE META-LEARNING (DAAML)

5a. CONTRACT NUMBER
FA8750-19-C-0511

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
61101E

6. AUTHOR(S)

Yi Yao, Meng Ye, Xiao Lin, Giedrius Burachas, Nikoletta Basiou, and Ray Kolczynski

5d. PROJECT NUMBER
LWLL

5e. TASK NUMBER
00

5f. WORK UNIT NUMBER
09

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
SRI International
201 Washington Road
Princeton NJ 08540

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RISB
525 Brooks Road
Rome NY 13441-4505

DARPA/I2O
675 N. Randolph St
Arlington VA 22203-2114

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2021-156

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Our Domain-Adaptive Active Meta-Learning (DAAML) system leverages semi-supervised few-shot learning including unsupervised representation learning for maximized data economy, active sampling (i.e., active learning) to maximize information gain per label query, and adaptive module selection for cross-domain few-shot learning. In this report, we detail our technical approach and experimental results regarding these three major building modules. This effort is funded by the DARPA Learning with Less Labels (LwLL) program.

15. SUBJECT TERMS

Few-shot learning, cross-domain few-shot learning, active learning, image classification, action recognition

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**PETER A. JEDRYSIK**
U	U	U	UU	31	19b. TELEPHONE NUMBER (*Include area code*) **NA**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. SUMMARY

Our Domain-Adaptive Active Meta-Learning (DAAML, Figure 1) system leverages semi-supervised few-shot learning including unsupervised representation learning for maximized data economy, active sampling (i.e., active learning) to maximize information gain per label query, and adaptive module selection for cross-domain few-shot learning. In this report, we detail our technical approach and experimental results regarding these three major building modules. This effort is funded by the DARPA Learning with Less Labels (LwLL) program.

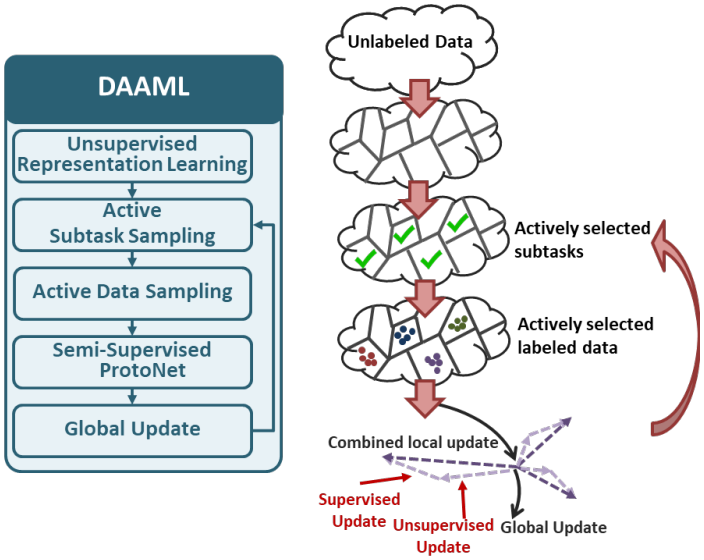

Figure 1. Overview of DAAML.

2. INTRODUCTION

DARPA LwLL aims to significantly reduce the amount of labeled data needed to train deep neural networks for diverse tasks including image classification/detection, action recognition, and machine translation. To achieve LwLL envisioned data reduction, DAAML leverages three distinctive techniques (Figure 1): 1) semi-supervised few-shot learning including unsupervised representation learning for maximized data economy, 2) active sampling (i.e., active learning) to maximize information gain per label query, and 3) adaptive module selection for cross-domain few-shot learning.

More specifically, we introduce hybrid consistency training for semi-supervised few-shot learning to impose interpolation consistency on linear combinations of samples and their transformed/augmented versions. In so doing, we can exploit both cross-sample linearity and local continuity to learn a richer representation on smoother manifolds for improved generalizability under data scarce conditions. We also develop calibrated iterative prototype adaptation for improved few-shot transductive inference. We normalize features to compensate for distribution variations and update prototypes iteratively using unlabeled data.

As for active learning under data scarce conditions, we observe that the traditional information-based (e.g., entropy) active learning suffers from degraded effectiveness due to the decreased accuracy in information gain estimation. To address the emergent issue of information-based active learning, we design active learning based on distances in the learned embedding space, a more robust metric especially in few-shot learning scenarios.

Finally, we propose adaptive module selection for cross-domain few-shot learning to dynamically select the most appropriate modules based on the source/target domains and the amount of available training examples. We standardize candidate techniques into chainable modules. We then employ Bayesian optimization to turn on or off the individual modules and to configure the hyperparameters of each selected module. Our adaptation method can be considered as applying different loss terms according to the chosen modules sequentially to progressively adjust the input model towards the target downstream task.

We have validated the effectiveness of our three main techniques on image classification and action recognition tasks using public datasets and LwLL development datasets. We have achieved state-of-the-art performance on few-shot learning and cross-domain few-shot learning tasks. We have also demonstrated three times and 22 times label reduction on the DomainNet-Clipart and MNIST datasets, respectively.

3. METHODS, ASSUMPTIONS AND PROCEDURES

3.1. Few-shot learning

Despite its successful applications in various computer vision tasks, deep learning remains challenging in low data regimes. Recently, Few-Shot Learning (FSL) has drawn increasing attention in various computer vision tasks, including image classification, object detection and semantic segmentation. In FSL, the training classes (i.e., seen or base classes) and the testing classes (i.e., unseen or novel classes) are disjoint. In order to perform classification on novel classes using only a few labels, certain form of knowledge must be learned and transferred from base to novel classes. Such knowledge can be a metric space [1] [2] [3], a model initialization [4], a learning algorithm [5], or simply an embedding model [6] [7]. While having demonstrated success on few-shot tasks, these approaches still fall short in addressing the following longstanding challenges: 1) large semantic gap between base and novel classes and 2) sparsity of labeled data of novel classes.

To tackle semantic gaps between base and novel classes, learning richer features to reduce overfitting on the base classes via incorporating knowledge learned from the images themselves is a promising direction [7]. For example, self-supervised losses, such as rotation [8] and exemplars [9], are employed in addition to the supervised loss on base classes for improved features [10] [11] [12]. Instead of constructing explicit surrogate tasks, another popular line of works exploits additional regularization such as consistency losses, inspired by semi-supervised learning. For example, interpolation consistency [13] [9] [14] encourages a model's local linearity and data augmentation consistency [15] [16] [17] enforces a model's local continuity.

We developed Hybrid Consistency Training (HCT) which uniquely combines the above two consistencies by directly imposing interpolation linearity on top of weakly and strongly augmented samples across intermediate features, as opposed to commonly used post-hoc combination of two independent losses (Figure 2). Specifically, we construct mixed features at a randomly selected network layer using a weakly and strongly augmented samples from a pair of labeled input images. The loss is measured by the cross entropy between model predictions of such mixed features and the linear combination of the ground truth labels of the original input images. Intuitively, weakly, and strongly augmented samples reside in a smaller (with limited variations) and a larger (with richer variations) neighborhood of the original image, respectively. Applying interpolation consistency on strongly augmented samples enforces local continuity and linearity in a wider range, leading to richer yet more regularized embedding space. Moreover, applying interpolation consistency across intermediate features further smoothens decision boundaries throughout all network layers. Richer yet flattened (i.e., with fewer directions of variance) representations and smoother decision boundaries lead to improved generalization capability despite large semantic gaps.

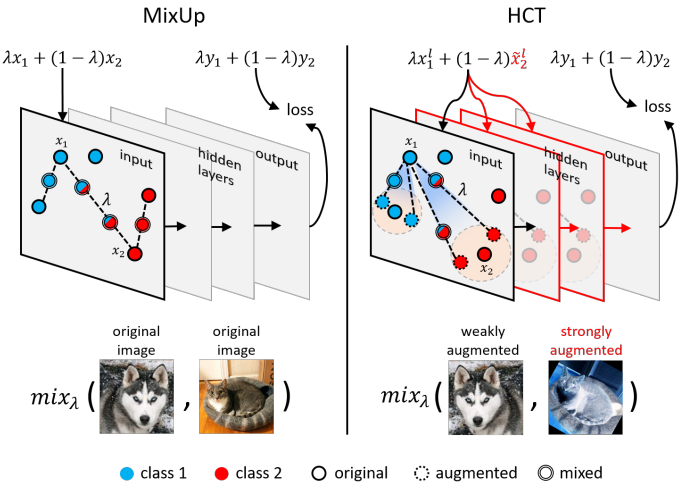

Figure 2. Comparison between Mixup and Hybrid Consistency Training (HCT).

The second challenge stems from the sparsity of labeled samples from novel classes. In this regard, transductive inference is introduced to leverage unlabeled data to fill in the gaps between labeled and query examples [18]. In this work, we advance prototype-based transductive inference by introducing an iterative method to calibrate features and adapt prototypes of novel classes using unlabeled data, referred to as Calibrated Iterative Prototype Adaptation (CIPA). While being simple, feature calibration (e.g., power transformation, centering, normalization) is a critical step that aligns samples from the support and query/unlabeled sets, producing an improved common ground for distance computation. Meanwhile, by estimating pseudo-labels on unlabeled data and updating prototypes iteratively, prototype estimations can be more precise despite the sparse and non-uniformly distributed labeled samples. Compared to another iterative method [19], where Sinkhorn [20] mapping is employed for pseudo labeling unlabeled data, our CIPA uses simple but effective cosine similarity, which requires much less computation. More critically, [19] assumes equal number of examples per class. In contrast, our CIPA does not rely on such assumptions and can work properly even under class imbalance.

3.1.1. Hybrid consistency training

Assume that the embedding function is a composition of multiple layers $f_\phi = f^L \circ \cdots \circ f^1 \circ f^0$. The hidden representation at layer l can be obtained by passing the input image through layer $0, 1, \ldots, l$: $h^l = f^l \circ \cdots \circ f^1 \circ f^0(x)$. Note that f^0 is the input layer and $h^0 = f^0(x) = x$. Given an embedding model, we optimize its weights by minimizing the following loss function $\mathcal{L} = \mathcal{L}_{ce} + \eta\mathcal{L}_{hct}$, where $\mathcal{L}_{hct}$ is the cross entropy loss on the base classes, η is a balancing parameter (we set it to 1 in all our experiments), and $\mathcal{L}_{hct}$is our newly introduced hybrid consistency loss which we explain in details below.

Consistency training has been widely used in semi-supervised learning. In this work, we propose HCT, combining two different consistency training approaches into a unified framework to regularize model training. Given any two images x_1 and x_2, we perform weak augmentation, e.g., horizontal flip, to x_1 so that the augmented image is still close to the original image. We overload the notation x_1 to represent both the original image and its weakly augmented version. For x_2, we apply strong augmentation so that it is heavily distorted and has a higher chance of

being out of the local data distribution. We denote this example as $\tilde{x}_2$. To generate an interpolation between x_1 and $\tilde{x}_2$, we feed them both into the embedding network and then randomly choose a layer *l* to get their hidden representations:

$$x_1^l = f^l \circ \cdots \circ f^1 \circ f^0(x_1)$$
$$\tilde{x}_2^l = f^l \circ \cdots \circ f^1 \circ f^0(\tilde{x}_2)$$

The hidden representations are mixed and passed through the remaining layers to get the final feature representation $\bar{x}$

$$\bar{x}^l = \lambda x_1^l + (1-\lambda)\,\tilde{x}_2^l$$
$$\bar{x} = f^L \circ \cdots \circ f^{l+1}(\bar{x}^l)$$

The corresponding target $\bar{y}$ is the interpolation of the ground truth one-hot label vectors y_1 and y_2 of the original input samples x_1 and x_2

$$\bar{y} = \lambda y_1 + (1-\lambda) y_2$$

The loss function on these interpolated examples is

$$\mathcal{L}_{hct} = \mathbb{E}_{\substack{(x_1,y_1)\in\mathcal{D}_{base} \\ (x_2,y_2)\in\mathcal{D}_{base} \\ \lambda\sim Beta(\alpha,\alpha),l\sim U(0,L)}} \sum_{c=1}^{|\mathcal{C}_{base}|} -\tilde{y}_c \log p_c(\tilde{x})$$

HCT combines interpolation consistency and data augmentation consistency in a unique and tightly integrated way: the generated new data points not only cover linear space between examples, but also expand further to the regions where heavily distorted examples reside. By doing this at a random layer each time, hidden representations at all levels are regularized. This leads to a smoother manifold that generalizes better to novel classes. HCT can also be combined with other representation learning techniques, e.g., self-supervised rotation classification $\mathcal{L}_{rot}$, by simply adding another head and performing multitask learning (denoted as HCT_R), which often results in further improved representations.

3.1.2. Calibrated iterative prototype adaptation

We use the embedding model trained by HCT to infer predictions for novel data. Given a novel task, we first extract features of both the support examples and the query examples. A straightforward way to get class probabilities is to compute class prototypes and then the distances between query examples and each prototype followed by softmax. However, due to the sparsity and sporadicity (i.e., nonuniformly distributed) of the support examples, the quality of prototypes varies substantially from episode to episode. In order to better estimate class prototypes as well as better adapt to specific tasks, we need to make full use of unlabeled query examples for semi-supervised or transductive inference. As described in [21], pseudo-labels can be used to update prototypes in a K-means step.

Another problem of centroid-nearest neighbor method is that, since each time only a few data points are sampled, the data distribution of a single task drifts heavily from the overall data distribution. Thus, certain transformations [22] [19] on the features are needed to calibrate them. To this end, we propose CIPA that: 1) calibrates the features for better distance computation, and 2) iteratively predicts pseudo-labels on unlabeled data and updates the estimation of prototypes progressively.

3.2. Active learning

Introduction. The goal of active learning is to adaptively select a set of unlabeled samples for annotation to maximize the performance gain per label query. The entropy-based methods

select samples the label of which are most uncertain whereas goal-driven-based methods select samples with stronger influence on test samples. However, for few-shot active learning, the accuracy of information gain estimation can degrade substantially especially for weak models (e.g., models trained with few labeled samples). This directly affects the effectiveness of the aforementioned active learning methods.

Methods. To this end, we design active learning based on distances in the embedding space. As we adopt a two-stage training paradigm for few-shot learning: namely pre-training and adaptation, the pre-trained embedding space is well defined and the distance in such an embedding space, therefore, becomes a more robust metric. From distance-based point of view, the counterpart of entropy-based methods selects examples that are in-between two different classes whereas the counterpart of goal-driven-based methods selects samples that can better cover unlabeled test samples.

3.3. Adaptive module selection

In FSL, the task of building models using limited examples, adapting pre-trained representations has demonstrated successful applications in computer vision and natural language processing. Finetuning pre-trained networks and learning classifiers on top of these embeddings leads to highly accurate classifiers even for data scarce conditions. An ideal pre-trained representation to start with is one that is learned in domains relevant to the target task or sufficiently diverse to enable effective transfer. However, in practice, relevant data is scarce and there is often a certain degree of domain shifts between the pretext and downstream tasks, where label ontology, viewpoint, image style, and/or input modality may differ. As such, cross-domain FSL [23] [24] [25] recently brings renewed interest to this classical transfer learning problem focusing on the low data regime.

Existing studies show that depending on the characteristics of the underlying domain shifts, different downstream tasks may favor different adaptation methods, either finetuning-based [26] [25] or metric learning-based approaches [7] [6]. For finetuning-based methods, the degree of finetuning required may also depend on the domain differences [27] [28] [29] and the amount of training data available in the target domain [30]. As a result, developing a one-size-fits-all cross-domain FSL approach has been challenging, if not entirely infeasible. To facilitate the appropriate design of cross-domain few-shot algorithms, a deeper understanding of domains, algorithms and their relationships is critical.

We propose a modular approach for cross domain few-shot adaptation that can be dynamically customized based on domain characteristics as an alternative to commonly used one-size-fits-all solutions. We refer to our method as the Modular Adaptation Pipeline (MAP, Figure 3). To build MAP, we first standardize few-shot adaptation methods into modules, where each module takes as the input a model and the adaptation datasets and produces an adapted model as the output. We chain these modules into a consolidated pipeline, where multiple adaptation methods are applied in sequence. Given a downstream task, which adaptation modules should be applied as well as the hyperparameters of each selected module can be flexibly configured and automatically optimized via Bayesian optimization.

MAP can, thus, be considered as applying different loss terms according to the chosen methods sequentially to progressively adjust the input model towards the target downstream task. MAP enables flexible and customized integration of various state-of-the-art (SOTA) techniques, whereas schemes based on joint optimization with an integrated loss function (e.g., weighted combination of multiple loss terms) may find it difficult to accommodate such diverse methods.

Furthermore, while, in theory, our modular approach sacrifices global optimality due to stepwise optimization, schemes based on joint optimization, in practice, may also find it difficult to arrive at the global optima simply because of the complexity of the loss landscape, which can be further complicated by the involvement of module hyper-parameters. As a result, our modular and sequential approach stands in contrast to methods based on joint optimization as an arguably better design choice for practical problems.

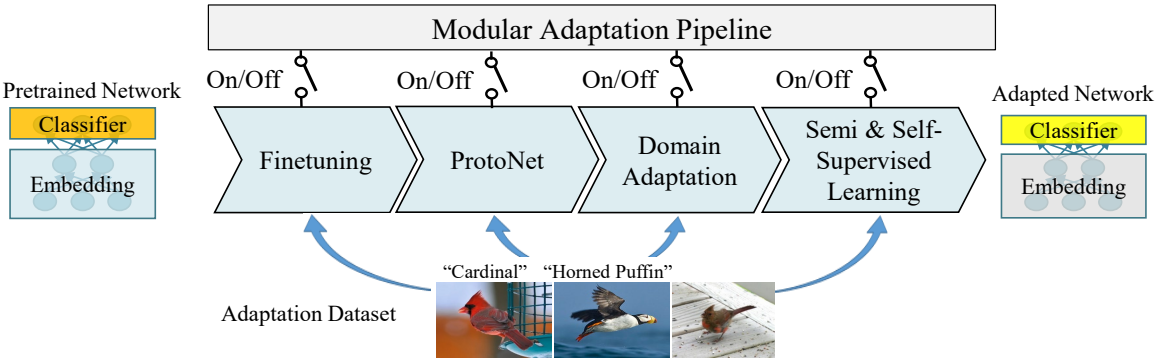

Figure 3. Block diagram of our proposed modular adaptation pipeline (MAP).

3.3.1. Building blocks

We implement seven modules based on adaptation methods with demonstrated successes in literature. The goal is to cover a diverse set of techniques such as FSL, domain adaptation and semi-supervised learning.

Finetuning. Finetune both the encoder (i.e., networks that map the input to an embedding feature space) and the classifier layers. The choice of optimizer (e.g., Adam or SGD), learning rate, momentum, weight decay, data augmentation, batch size, number of epochs and learning rate stepping schedule are set as hyperparameters. We also included the option of re-initializing the classification network with a fully connected layer.

Transductive Prototypical Networks (TransPN). Prototypical Networks with scaled cosine distance [6], embedding power scaling [19], and CIPA prototype propagation for transductive learning. These algorithms have been shown to reach SOTA on FSL datasets. The cosine distance scaling factor, power scaling factor and the toggle of transductive learning are part of the hyperparameters. TransPN reduces the classification layer to simply comparing the embedding of a test sample against class prototypes based on scaled cosine similarity.

Finetuning batchnorm layer (TuneBN). Updating batchnorm statistics with unlabeled data from the target domain helps model adaptation in few-shot settings [31] [25]. Network weights are frozen while batches of unlabeled data are passed through the network to update batchnorm statistics. TuneBN also serves the role of setting the momentum of batchnorm layers of the input network, which is an important factor in adaptation.

Semi-supervised learning with pseudo labels (SSL-PseudoLabel). High-confidence predictions on unlabeled data are used as "pseudo labels" along with ground truth labels during training [32].

Semi-supervised learning with entropy minimization (SSL-Entropy). Entropy on unlabeled examples is used as an additional loss term during training [33].

Semi-supervised learning with student-teacher (SSLMeanTeacher). A semi-supervised learning approach that uses predictions from a running average network as the "mean teacher" to regularize the training of a student network [34].

Semi-supervised learning with FixMatch (SSLFixMatch). Use consistency between strongly and weakly augmented inputs for semi-supervised learning [17].

3.3.2. Pipeline design

Figure 4 depicts an exemplar realization of our MAP. It consists of 11 modules. Finetuning is followed by semi-supervised learning with BatchNorm and ProtoNet modules in-between. Each module can be switched on or off. When switched off, a module is replaced with a skip connection. While our empirical pipeline design does not cover all possible combinations of the building blocks, it supports a rich and compact set of configurations for selecting various methods. It covers standard baselines such as BatchNorm + ProtoNet and BatchNorm + finetuning. It also follows best practices such as finetuning the network before semi-supervised learning.

The search space of hyperparameters consists of switches, $\{\text{on; off}\}^n$, and the hyperparameters for each module, 126 hyperparameters in total. By keeping the number of modules in our pipeline fixed, standard Bayesian hyperparameter search techniques can be used to optimize MAP given a downstream task. Neural architecture search [35] and hyperparameter search transfer learning [36] may enable further expansion of the search space and potentially result in better pipelines. In this work, we focus on studying the feasibility and impacts of MAP and leave the aforementioned further improvements to future work.

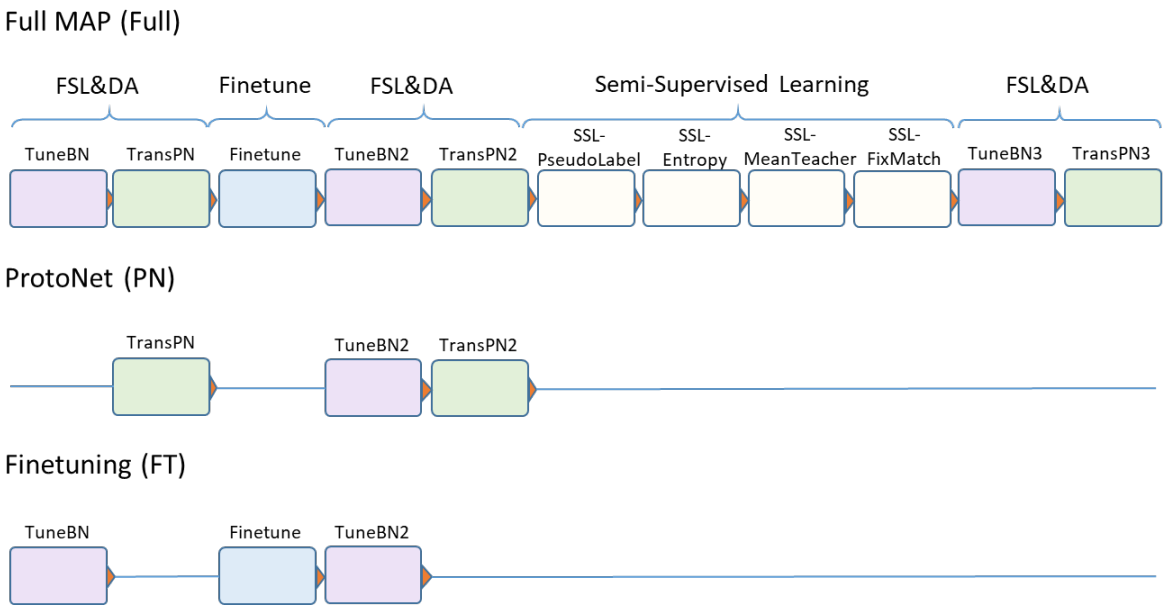

Figure 4. An exemplar realization of MAP (top) with ProtoNet (middle) and finetuning (bottom) as special cases.

3.3.3. Search strategy

From our experiments, we find that MAP optimization from scratch typically requires >300 iterations to converge, which often takes days on a single GPU. As reference, finetuning and ProtoNet, on the other hand, often converge within 30 iterations. To speed up MAP optimization, we propose to transfer MAP hyperparameters across domains. Intuitively, a MAP optimized for one adaptation problem may also perform well on similar problems with similar requirements. It, therefore, could serve as a better starting point of hyperparameter search than from scratch. A set of MAPs optimized for a diverse set of domains could readily cover a large variety of adaptation problems so that satisfactory performance could be reached by simply selecting the best performing MAP out of the set for a given new downstream task.

Specifically, we first collect an initial set of 20 to 50 high-performing MAPs on N-way K-shot adaptation tasks sampled from a diverse collection of datasets (excluding the target dataset), by running MAP hyperparameter search from scratch for 500 iterations. Building the initial collection of MAPs may take weeks across multiple GPUs. However, during adaptation, only MAPs in the collection are evaluated on the cross-validation splits to select the most suitable MAP for the given downstream task. We show in experiments that transferring with a set of MAPs reaches similar accuracy as searching from scratch most of the times but converges more than $10\times$ faster.

Our search protocol is related to hyperparameter transfer learning [36] and BiT-Hyperrule [30], an empirical rule for determining finetuning hyperparameters. Different from [36] which learns a transferable embedding space of hyperparameters, our approach simply finds hyperparameters that addresses a diverse set of tasks. MAP creates a more extended hyperparameter space than finetuning, which may open opportunities for deriving more sophisticated domain-dependent hyperparameter rules.

4. RESULTS AND DISCUSSIONS

4.1. Few-shot learning

4.1.1. Experimental settings

Datasets. We conducted experiments on five FSL datasets:

1) mini-ImageNet [2] is derived from the ILSVRC2012 [37] dataset. It contains 100 randomly sampled classes and is split into 64, 16 and 20 classes for train, validation, and test, respectively. Each class has 600 images.

2) tiered-ImageNet [21] is also a subset of ILSVRC2012. It contains in total 34 super categories and is split into 20, 6 and 8 for train, validation, and test, respectively. The corresponding class numbers are 351, 97, and 160. On average, each class has around 1280 images.

3) CIFAR FS [38] is a few-shot learning dataset that contains all 100 classes from CIFAR100 [39]. The dataset is randomly split into 64, 16 and 20 classes for train, validation, and test. Each class has 600 images.

4) FC 100 [38] is also derived from CIFAR100 [39]. But it is instead split into 60, 20 and 20 classes that are from 12, 4 and 4 super categories. It is like the "tiered" version of CIFAR FS. Similarly, it has 600 images for each class.

5) CUB [40] is a dataset of 200 fine-grained bird species. We follow [41] to split the dataset into 100, 50 and 50 for train, validation and test. This dataset only has around 59 images for each class. For all these five datasets, we resize the images into 84×84 if they are not so already.

Training Settings. In all our experiments, we use ResNet-12 [6] as our backbone network. To train the network, we use the Adam optimizer with a learning rate of 0.001 and train for 300 epochs (60 on tiered-ImageNet). During the first $1/3$ of total epochs we use $\mathcal{L}_{ce} + \mathcal{L}_{rot}$, for the remaining $2/3$ of the epochs we add the $\mathcal{L}_{hct}$ loss term. To interpolate examples, by default we use $\alpha = 2$ to sample $\lambda \sim Beta(\alpha, \alpha)$ unless stated otherwise. For the weak augmentation, we use random crop and random flip at 50% chance. For the strong augmentation, we follow FixMatch [17] and use RandAugment [42]. Each time, 2 out of 14 augmentations are randomly selected and applied to the image, after which a random square region in the image is cut out [43]. We use the same settings for all datasets to obtain our main results. Performance on validation data is monitored during training for model selection.

Evaluation Settings. In the test phase, we fix the trained backbone network and use it as a feature extractor. The extracted features of the support and query samples are used by CIPA to predict their classes. We use $\beta = 0.5,\ \sigma = 0.2, \tau = 10$ and $N_{iter} = 20$ for all experiments. In each experiment, a number of novel episodes (600 or 10,000) are sampled. Each episode contains N classes, and each class has K support and 15 query examples. Note that we do not use any auxiliary unlabeled examples as did in semi-supervised FSL [44] [45] and thus these methods are not comparable to ours. We report the average accuracy and 95% confidence interval as performance measurements.

4.1.2. Experimental Results

Standard FSL. We separate comparison methods into the inductive and transductive groups. In both groups, we list SOTA results with a similar backbone (e.g., ResNet12, ResNet18) as well as those with heavier backbones (e.g., WRN and DenseNet). Note that performance

achieved with deeper backbones are not directly comparable to our results, those are listed just for reference.

We summarize the results on mini-ImageNet and tiered-ImageNet in Table 1. We can see that our method, HCT_R+CIPA, has achieved the best performance among comparison methods on the mini-ImageNet dataset. Comparing to LaplacianShot [46], the best performing method reported using a ResNet18 backbone, we achieve more than 4% and nearly 3% improvements on 1-shot and 5-shot, respectively. As for tiered-ImageNet, HCT_R+CIPA yields the best performance on 1-shot while being on par with EPNet [47] and LR+ICI [48] on 5-shot.

The results on CIFAR FS and FC100 are summarized in Table 2. Similarly, our method achieves the best performance across all settings. Note that some of the methods in the inductive group, such as CC+rot and $S2M2_R$, use a larger network (e.g., WRN-28-10). Our method still outperforms them, showing that our training method combined with transductive inference can compensate for the disadvantages of using a lighter network.

CUB is a different dataset from the previous ones in that it contains fine-grained bird species as classes rather than generic objects. We summarize results on CUB in Table 3. Our approach has achieved the best performance on both 1-shot and 5-shot with an improvement of ~5% and ~2%, respectively, over LR+ICI [48], the best reported method in literature using a ResNet12 backbone. Notably, even comparing to transductive methods with a larger backbone of WRN-28-10 (e.g., PT+MAP [19]), our method still remains the best. The results on CUB strongly suggest that regularizing learned embedding in a wider extent and across network layers can help to learn rich and robust representations to significantly benefit FSL on fine-grained classes.

Table 1. Results on mini-ImageNet and tiered-ImageNet.

	Method	Backbone	5-way mini-ImageNet		5-way tiered-ImageNet	
			1-shot	5-shot	1-shot	5-shot
Inductive	TADAM [49]	ResNet12	58.50±0.30	76.70±0.30	-	-
	ProtoNet [1]	ResNet12	59.25±0.64	75.60±0.48	61.74±0.77	80.00±0.55
	MetaOptNet [50]	ResNet12	62.64±0.61	78.63±0.46	65.99±0.72	81.56±0.53
	SNAIL [51]	ResNet15	55.71±0.99	68.88±0.92	-	-
	SimpleShot [22]	ResNet18	62.85±0.20	80.02±0.14	69.09±0.22	84.58±0.16
	DeepEMD [52]	RestNet12	65.91±0.82	82.41±0.56	71.16±0.87	86.03±0.58
	LEO [53]	WRN-28-10	61.76±0.08	77.59±0.12	66.33±0.05	81.44±0.09
	CC+rot [8]	WRN-28-10	62.93±0.45	79.87±0.33	70.53±0.51	84.98±0.36
	$S2M2_R$ [54]	WRN-28-10	64.93±0.18	83.18±0.11	73.71±0.22	88.59±0.14
Transductive	TPN [18]	ResNet12	59.46	75.65	58.68	74.26
	Fine-tuning [55]	ResNet12	62.35±0.66	74.53+0.54	68.41±0.73	84.41±0.52
	TEAM [56]	ResNet18	60.07	75.90	-	-
	LR+ICI [16]	ResNet12	66.80	79.26	80.79	87.92
	DSN-MR [57]	ResNet12	64.60±0.72	79.51±0.50	67.39±0.82	82.85±0.56
	EPNet [47]	ResNet12	66.50±0.89	81.06±0.60	76.53±0.87	87.32±0.64
	FEAT [58]	ResNet18	66.78±0.20	82.05±0.14	70.80±0.23	84.79±0.16
	LaplacianShot [46]	ResNet18	72.11±0.19	82.31±0.14	78.98±0.21	86.39±0.16
	HCT_R+CIPA (ours)	**ResNet12**	**76.94±0.24**	**85.10±0.14**	**81.07±0.25**	**87.91±0.15**
	ICA+MSP [59]	DenseNet	77.06±0.26	84.99±0.14	84.29±0.25	89.31±0.15
	PT+MAP [19]	WRN#/DenseNet*	82.92±0.26#	88.82±0.13#	85.67±0.26*	90.45±0.14*

Table 2. Results on CIFAR FS and FC100.

	Method	Backbone	5-way CIFAR_FS		5-way FC100	
			1-shot	5-shot	1-shot	5-shot
Inductive	TADAM [49]	ResNet12	-	-	40.1±0.4	56.1±0.4
	ProtoNet [1]	ResNet12	72.2±0.7	83.5±0.5	37.5±0.6	52.5±0.6
	MetaOptNet [50]	ResNet12	72.0±0.7	84.2±0.5	41.1±0.6	55.5±0.6
	SimpleShot [22]	ResNet18	-	-	40.13±0.18	53.63±0.18
	DeepEMD [52]	RestNet12	-	-	46.47±0.78	63.22±0.71
	CC+rot [8]	WRN-28-10	76.09±0.30	87.83±0.21	-	-
	$S2M2_R$ [54]	WRN-28-10	74.81±0.19	87.47±0.13	-	-
Transductive	TPN [18]	ResNet12	65.89	79.38	-	-
	Fine-tuning [55]	ResNet12	70.76±0.74	81.56±0.53	41.89±0.59	54.96±0.55
	TEAM [56]	ResNet18	70.43	81.25	-	-
	LR+ICI [16]	ResNet12	73.97	84.13	-	-
	DSN-MR [57]	ResNet12	75.6±0.9	86.2±0.6	-	-
	HCT_R+CIPA (ours)	**ResNet12**	**85.72±0.21**	**89.69±0.14**	**53.30±0.25**	**64.90±0.20**
	PT+MAP [19]	WRN-28-10	87.69±0.23	90.68±0.15	-	-

Cross-domain FSL. To study the robustness of representations learned via our HCT across datasets with certain amounts of covariate shift, we evaluate its performance under cross-domain scenarios as an outreaching test. We train models on mini-ImageNet and test them on CUB. From Table 4, our method achieves the best performance on both 1-shot and 5-shot tasks, with an improvement of 7% and 8% over LaplacianShot, respectively. The 1-shot accuracy is on par with PT+MAP despite that HCT uses a shallower network. This manifests that our method not only works under in-domain settings, but also can generalize well under the more challenging cross-domain settings.

Table 3. Results on CUB - ours are averaged over 10k episodes.

	Method	Backbone	5-way CUB	
			1-shot	5-shot
Inductive	DeepEMD [52]	RestNet12	75.65±0.83	88.69±0.50
	$S2M2_R$ [54]	WRN-28-10	80.68±0.81	90.85±0.44
Transductive	TEAM [56]	ResNet18	80.16	87.17
	LaplacianShot [46]	ResNet18	80.96	88.68
	LR+ICI [16]	ResNet12	88.06	92.53
	HCT_R+CIPA (ours)	ResNet12	93.03±0.15	94.09±0.08
	BD-CSPN [60]	WRN-28-10	87.45	91.74
	PT+MAP [19]	WRN-28-10	91.55±0.19	93.99±0.10

Table 4. Results for cross-domain FSL.

	Method	Backbone	mini-ImageNet à CUB	
			1-shot	5-shot
Inductive	MatchingNet [2]	RestNet10	36.61±0.53	55.23±0.83
	ProtoNet [1]	RestNet10	44.07±0.77	59.46±0.71
	$S2M2_R$ [54]	WRN-28-10	48.24±0.84	70.44±0.75
Transductive	GNN [61]	ResNet10	47.47±0.75	66.98±0.68
	LaplacianShot [46]	ResNet18	55.46	66.33
	HCT_R+CIPA (ours)	**ResNet12**	**62.15±1.08**	**74.51±0.77**
	PT+MAP [19]	WRN-28-10	62.49±0.32	76.51±0.18

4.2. Active learning

We conducted experiments under the cross-domain few-shot learning setting where we pre-trained our embedding on the ImageNet-1K dataset and test on CIFAR-100, CUB, DomainNet-Real, and DomainNet-Clipart datasets. We start from 1 seed label per class, query for 20% more data per query until label budget is reached, and report classification accuracy over 10 active learning runs.

Figure 5 shows classification accuracy on DomainNet-Real using different labeling strategy including random query, entropy-based, and our distance-based methods. For cases with less than 50 labels/class, entropy-based method can be unreliable, resulting in inferior classification accuracy. In contrast, our distance-based methods can improve classification accuracy in low-shot regime (4.5% accuracy improvement at 6.2 labels/class or equivalently 2.5× data reduction at 10 labels/class).

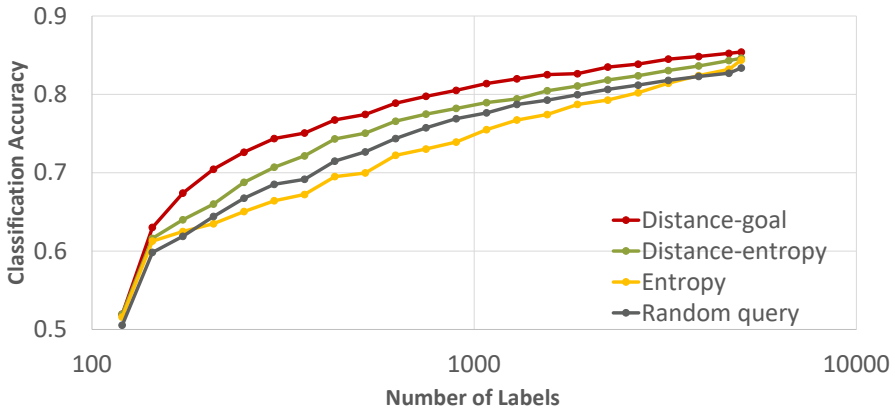

Figure 5. Active learning on DomainNet-Real.

Figure 6 to Figure 8 show the results on DomainNet-Clipart, CIFAR-100, and CUB datasets, respectively. We achieved 5.1%, 4.9%, and 3.5% accuracy improvement at 7.5 labels/class, 7.5 labels/class, and 4.3 labels/class for the DomainNet-Clipart, CIFAR-100, and CUB datasets,

respectively. The corresponding data reduction is about 1.7×, 1.5×, and 1.2× at 10 labels/class.

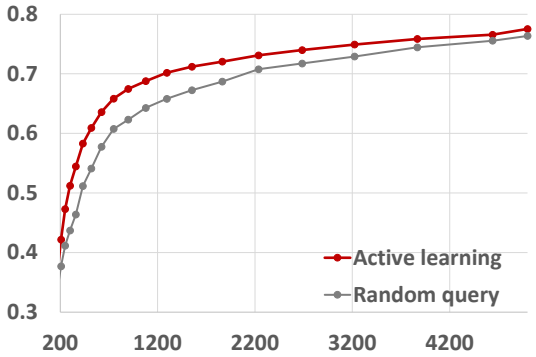

Figure 6. Active learning on DomainNet-Clipart.

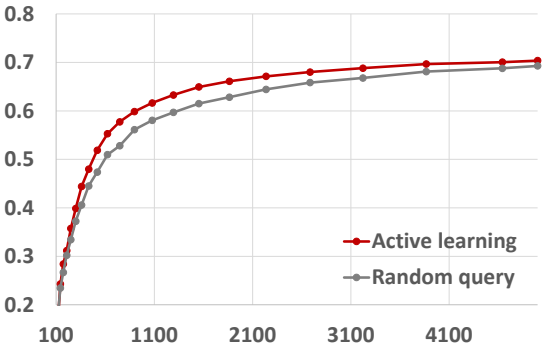

Figure 7. Active learning on CIFAR-100.

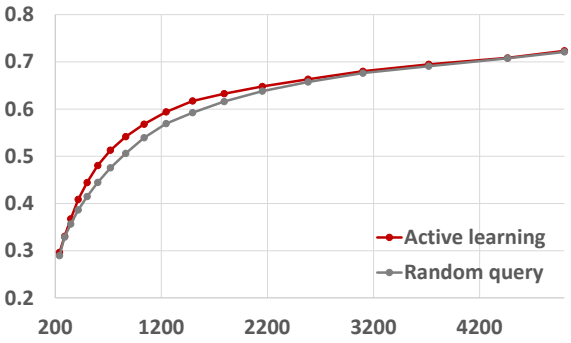

Figure 8. Active learning on CUB.

4.3. Adaptive module selection

4.3.1. Experimental settings

To fully benefit from representation learning, we introduce a new large-scale 100-way 2-to 20-shot ImageNet → 10 datasets benchmark for cross-domain few-shot learning. These 10 adaptation datasets include DomainNet-real, clipart, sketch, quickdraw, infograph, painting [62], CIFAR-100, DTD-textures [63], FGVC-aircraft [64] and GTSRB-traffic signs [65]. The choice of datasets is inspired by the Taskonomy [66] and Meta-Dataset [23] benchmarks but with focus on fixed-way fixed-shot to study domain-dependent adaptation strategies. All classes in the respective datasets are used except for DomainNet, where only the top 100 most frequent classes are used as there are insufficient testing examples for categories towards the tail of the class distribution. We randomly sample images to create $K \in \{2, 5, 10, 20\}$-shot adaptation tasks with 20 test examples per class. The process is repeated over 5 random seeds to create 5 splits for each of the N-way K-shot problem. Accuracy is reported per dataset per shot, averaged over the 5 random splits. Following existing FSL conventions, unlabeled test examples are available for semi-supervised and/or transductive learning [21] [18].

We also benchmark our approach using existing cross-domain few-shot datasets including the VL3 challenge [25] (5-way 5- to 20-shot miniImageNet → CropDisease [67], EuroSAT [68], ISIC [69] and ChestX [70]) and the four datasets from the LFT work [24] (5-way 1-&5- shot mini-ImageNet → CUB [40], Cars [71], Places [72] and Plantae [73]).

4.3.2. Experimental results

Table 5 compares the performance of MAP vs. baselines, i.e., Prototypical Networks (PN), finetuning (FT), on the ImageNet→10 datasets cross-domain task. Between PN and FT, we observe that PN, a FSL method, and FT, a transfer learning method, performs better on lower-shots (i.e., 2-shot) with similar domain and higher-shots (i.e., 5-, 10-, 20-shots) with disjoint domain, respectively. This agrees with the underlying designs of these methods and the general observations in literature. Thanks to its modularized design that supports online pipeline reconfiguration, MAP on average outperforms both PN and FT for 2-, 5-, 10-shots by 3.96%, 3.13% and 1.04%, respectively. For 20-shot, MAP achieves comparable performance as FT. As the number of shots increases, the performance gain achieved by MAP degrades. This is expected since in general FT outperforms few-shot methods given sufficient labeled data.

Table 5. Comparing performance of MAP against PN and FT.

		QDraw	Infgrph	Sketch	Clipart	Pnting	Real	CIFAR	Text.	Aircraft	Signs	**Overall**
2-shot	PN	21.37	8.35	21.59	35.60	36.89	66.68	32.33	40.62	11.85	27.30	30.26
	FT	31.88	8.61	22.68	35.54	34.22	57.02	28.87	30.40	16.61	58.37	32.42
	MAP	30.10	9.91	27.38	39.25	39.66	69.25	35.63	41.19	14.01	57.42	**36.38**
5-shot	PN	30.51	12.72	31.82	48.09	45.72	72.39	42.85	52.00	18.28	39.21	39.36
	FT	47.29	14.26	39.02	54.05	50.42	71.26	47.25	46.47	35.94	87.56	49.35
	MAP	45.73	16.49	43.23	59.21	55.21	74.94	55.97	52.15	36.13	85.74	**52.48**
10-shot	PN	35.23	16.75	40.85	54.99	51.44	75.50	49.25	56.04	22.86	44.74	44.77
	FT	58.14	20.00	48.88	65.98	58.22	77.05	59.54	56.22	51.51	92.79	58.83
	MAP	55.33	23.64	49.74	68.45	59.70	77.66	61.50	54.36	53.34	94.06	**59.78**
20-shot	PN	39.62	21.36	44.60	60.77	56.48	77.91	53.94	56.30	26.79	53.49	49.13
	FT	66.04	26.89	56.58	72.96	64.97	80.61	68.14	62.77	72.25	96.44	66.77
	MAP	62.89	30.28	55.65	75.23	65.25	79.93	68.42	61.12	75.05	98.37	**67.22**

We compare the performance of our MAP against existing approaches using the same backbone architectures on the LFT (Table 6) and VL3 (Table 7) cross-domain FSL benchmarks. We achieve competitive performance against SOTA on the small-scale 5-way datasets, especially for the LFT benchmark. Notably, on ISIC and ChestX in the VL3 benchmark, which are datasets of medical imagery, MAP performance is still relatively lower than the baselines. This is because our exemplar MAP does not include strong domain adaptation modules (the only domain adaptation comes from BatchNorm). By incorporating additional domain adaptation modules, we expect MAP performance to be further improved on these two datasets.

Table 6. Performance on 5-way 5-shot LFT benchmark.

Method	CUB	Cars	Places	Plantae
RelationNet	57.77±0.69	37.33±0.68	63.32±0.76	44.00±0.60
RelationNet+LFT	59.46±0.71	39.91±0.69	66.28±0.72	45.08±0.59
GNN	62.25±0.65	44.28±0.63	70.84±0.65	52.53±0.59
GNN+LFT	66.98±0.68	44.90±0.64	73.94±0.67	53.85±0.62
PN	66.48±1.08	51.68±1.16	73.66±1.04	58.98±1.12
FT	67.31±1.03	51.89±1.14	71.68±1.02	60.26±1.09
MAP	67.92±1.10	51.64±1.16	75.94±0.97	58.45±1.15

Table 7. Performance on 5-way 5- and 20-shot VL3 benchmark.

	CropDisease		EuroSAT		ISIC		ChestX	
	5-shot	20-shot	5-shot	20-shot	5-shot	20-shot	5-shot	20-shot
Trans. FT	90.64±0.54	95.91±0.72	81.76±0.48	87.97±0.42	49.68±0.36	61.09±0.44	26.09±0.96	31.01±0.59
FT+DA	92.23±0.46	95.95±0.30	82.67±0.50	87.84±0.46	51.76±0.50	60.32±0.59	31.60±0.41	35.91±0.42
PN	90.10±1.11	93.95±0.84	80.49±1.27	86.08±1.00	44.73±1.31	56.86±1.18	24.31±0.83	29.48±0.83
FT	84.52±1.37	96.07±0.75	79.93±1.31	89.93±1.29	46.86±1.29	62.86±1.62	24.29±0.85	30.60±1.09
MAP	90.29±1.56	95.22±1.13	82.76±2.00	88.11±1.78	47.85±1.95	60.16±2.70	24.79±1.22	30.21±1.78

Table 8 compares averaged performance of PN, FT, MAP across the six domains in DomainNet under the from-scratch, transfer, and oracle settings for 2-, 5-, 10-shot. Our experiments under the oracle setting show that the improvement margin between MAP and PN/FT on DomainNet is approximately 4.04%, 2.72%, and 0.85%, for 2-, 5-, and 10-shots, respectively. MAP under both from-scratch and transfer settings has achieved such improvement, suggesting that despite the large number of additional parameters introduced by dynamic pipeline configuration, MAP seems not yet suffer from over-fitting (i.e., similar improvement is achieved on the holdout fold of cross-validation). This allows for a larger search space by adding more modules into the MAP pipeline (e.g., self-supervised learning).

Table 8. Comparing the performance of from-scratch and transfer hyperparameter selection strategies against the oracle on 100-way K-shot ImageNet → DomainNet datasets.

Shots	From-Scratch				Transfer				Oracle			
	PN	FT	MAP	Δ	PN	FT	MAP	Δ	PN	FT	MAP	Δ
2	31.81	31.36	36.15	+4.34	31.75	31.66	35.93	+4.18	32.30	30.74	36.34	+4.04
5	40.77	44.14	47.70	+3.56	40.21	46.05	49.14	+3.09	41.06	46.28	49.00	+2.72
10	46.61	54.14	55.84	+1.70	45.79	54.71	55.75	+1.04	46.61	55.07	55.92	+0.85

Comparing between from-scratch and transfer, transfer achieves similar performance to from-scratch. The fact that the performance difference is marginal also verifies our conjecture that initializing from a set of pre-selected pipelines is a practical and effective alternative to searching the whole parameter space for reduced online computation. Figure 9 shows the convergence rate of transfer against vanilla Bayesian hyperparameter search on 5-shot DomainNet-Clipart, where transfer yields an approximately 20 × reduction in online computation.

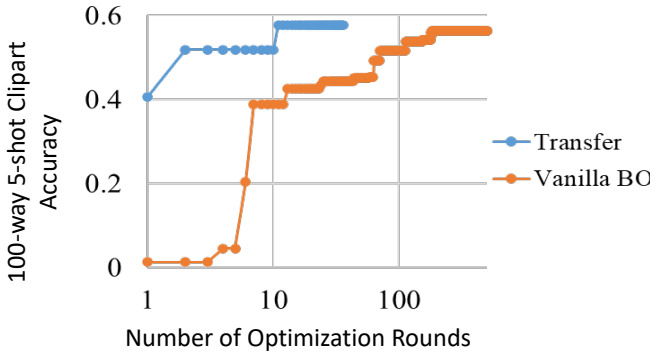

Figure 9. MAP convergence rate on a DomainNet-Clipart 5-shot task.

4.4. LwLL development datasets

We integrated our few-shot learning, active learning, and adaptive module selection algorithms and applied the LwLL development datasets. Below, we include experimental results on the development datasets to illustrate the achieved label reduction from each component of our approach in the figure below. Such step-by-step improvements may not be obvious in the Year 1 Evaluation. Specifically, on the DomainNet-Clipart and MNIST datasets, we achieved 3.0 and 22.3 times label reduction, respectively, against the Eval. team's baseline (i.e., ResNet50 finetuned with online hyperparameter search). Each algorithmic component makes its contribution towards data reduction, which validates its necessity.

DomainNet-Clipart MNIST

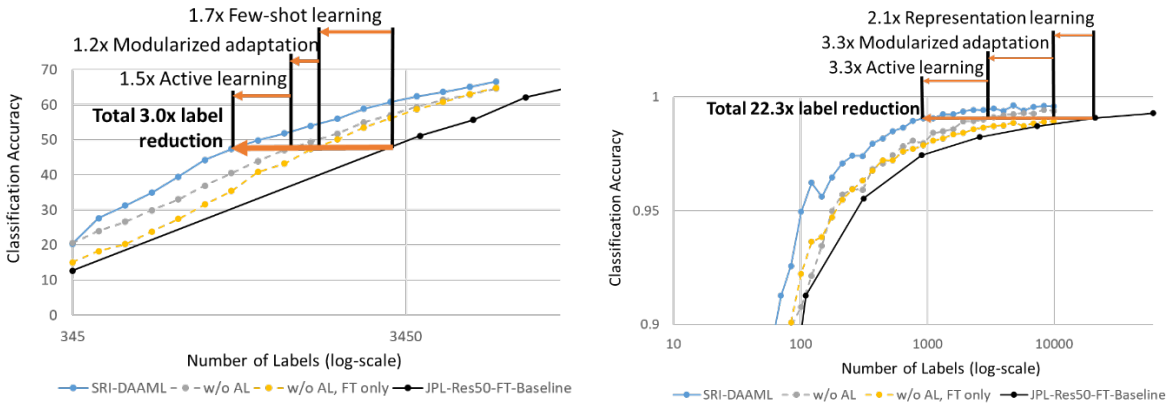

Figure 10. Classification performance on LwLL development datasets.

4.5. Few-shot action recognition

We first validate our prototype-based few-shot learning methods for action recognition. As our first baseline, we employ ResNet50 pre-trained on ImageNet to extract frame-level features. We average these frame-level features over T frames as the clip-level features. We then compute class prototypes by averaging clip-level features of samples in the support set. Given a testing sample, its category is determined by selecting the closest class prototype based on Euclidean distance. We evaluate the performance of our baseline on HMDB51 and UCF101 datasets under 5-way 1- and 5-shot settings. The table below shows that the performance of our baseline is comparable to some of the high-performing methods such as ARN [74] on HMDB51 and FAN [75] on UCF101.

Table 9. Few-shot action recognition on HMDB51 and UCF 101 datasets.

Method	Architecture	HMDB51		UCF101	
		1-shot	5-shot	1-shot	5-shot
ProtoGAN [76]	C3D	34.7±9.2	54.0±3.9	57.8±3.0	80.2±1.3
TAV [77]	I3D	36.0	53.1	68.3	88.3
ARN [74]	C3D	44.6±0.9	59.1±0.8	62.1±1.0	84.8±0.8
FAN [75]	DenseNet121	50.2±0.2	67.6±0.1	71.8±0.1	86.5±0.2
PAL [78]	ResNet-50	60.9	75.8	85.3	95.2
ITA [79]	C3D+LSTM	63.4±0.3	79.7±0.2	88.7±0.2	96.8±0.1
Ours	ResNet50-8	43.8±1.3	66.5±1.3	72.7±1.2	90.5±0.8
	ResNet50-4	43.6±1.3	65.8±1.3	72.3±1.2	90.0±0.8
	ResNet50-1	42.2±1.2	62.6±1.3	68.6±1.3	87.8±0.9

To improve the performance of few-shot action recognition, we explore two directions: 1) better architecture designed specifically for video processing and 2) better temporal alignment. For better architecture, we tap into two 3D convolutional networks pretrained on Kinetics: namely SlowFast [80] and X3D [81]. The table below lists the few-shot performance on HMDB51 and UCF101. With better pre-trained models, our prototype-based method readily achieves the state-of-the-art performance on HMDB51 and UCF101.

Table 10. Few-shot action recognition performance on HMDB51 and UCF 101 datasets using pre-trained 3D convolutional networks.

Method	Architecture	HMDB51		UCF101	
		1-shot	**5-shot**	**1-shot**	**5-shot**
ProtoGAN [76]	C3D	34.7±9.2	54.0±3.9	57.8±3.0	80.2±1.3
TAV [77]	I3D+SLDG(2,1)	36.0	53.1	68.3	88.3
ARN [74]	C3D	44.6±0.9	59.1±0.8	62.1±1.0	84.8±0.8
FAN [75]	DenseNet121	50.2±0.2	67.6±0.1	71.8±0.1	86.5±0.2
PAL [78]	ResNet-50	60.9	75.8	85.3	95.2
ITA [79]	C3D+LSTM	63.4±0.3	79.7±0.2	88.7±0.2	96.8±0.1
Ours	SlowFast-16	55.4±1.3	75.1±1.2	84.8±1.0	94.6±0.6
	SlowFast-8	46.4±1.3	65.9±1.3	78.6±1.1	92.3±0.7
	X3D-16	**62.4±1.3**	**82.8±1.0**	**90.4±0.8**	**97.8±0.4**
	X3D-8	58.2±1.2	80.8±1.1	87.1±0.9	97.1±0.5

5. CONCLUSIONS AND FUTURE RESEARCH

We tackled two longstanding difficulties in FSL. 1) To generalize from base to novel classes, we developed hybrid consistency training, a combination of interpolation consistency and data augmentation consistency to regularize the learning of representations. 2) To bridge the gap between sparse support and query examples, we developed a transductive inference algorithm, CIPA, to calibrate features and adapt prototypes iteratively. Through extensive experiments, we have shown that our method can achieve SOTA performance on all five FSL datasets. Ablation studies also justified the necessity and quantified the effectiveness of each component in HCT and CIPA. We also extended our prototype-based few-shot learning methods to action recognition. Our initial results on HMDB51 and UCF101 showed state-of-the-art accuracy in 1- and 5-shot classification. Incorporating prototype-based FSL and transformer-based temporal alignment, a critical step in handling temporal data for action recognition, can be a promising direction for future research.

As for active learning, we empirically demonstrated the deficiency of information-based approach under data scarce conditions due to the inaccurate probability estimation. We, then, studied distance-based approach and showed improved performance under few-shot (<10 examples per class) scenarios. Theoretical analysis of the performance bounds of information- versus distance-based approaches is an interesting direction for future research that can shed insightful light on active learning in few-shot scenarios.

Finally, we developed adaptive module selection as an alternative to one-size-fits-all solutions for cross-domain FSL. We showed 1) that various adaptation methods can be unified under a common framework to achieve greater adaptation performance; and 2) that given a specific downstream task with limited training examples, which adaptation methods to apply can be configured and optimized dynamically and efficiently through hyperparameter transfer. The effectiveness of adaptive module selection was verified via an average 3.1% improvement over finetuning on 5-shot ImageNet → 10 datasets benchmark and competitive performance against SOTA on LFT and VL3. Encouraged by these demonstrated advantages, promising directions for future research include speeding up online hyperparameter adaptation, improving hyperparameter generalization across datasets, and architectures and empirical rules for predicting hyperparameters.

6. REFERENCES

[1] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *NeurIPS*, 2017.

[2] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, and others, "Matching networks for one shot learning," in *NeurIPS*, 2016.

[3] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning Workshop*, 2015.

[4] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017.

[5] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *ICLR*, 2017.

[6] Y. Chen, X. Wang, Z. Liu, H. Xu, and T. Darrell, "A new meta-baseline for few-shot learning," *arXiv preprint arXiv:2003.04390*, 2020.

[7] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?," *ECCV*, 2020.

[8] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, and M. Cord, "Boosting few-shot visual learning with self-supervision," in *ICCV*, 2019.

[9] V. Verma *et al.*, "Manifold mixup: Better representations by interpolating hidden states," in *ICML*, 2019.

[10] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *ICCV*, 2015.

[11] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *ICLR*, 2018.

[12] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *ECCV*, 2016.

[13] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation Consistency Training for Semi-supervised Learning," in *IJCAI*, 2019.

[14] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[15] D. Berthelot *et al.*, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," in *ICLR*, 2020.

[16] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," *arXiv preprint arXiv:1904.12848*, 2019.

[17] K. Sohn *et al.*, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *NeurIPS*, 2020.

[18] Y. Liu *et al.*, "Learning to propagate labels: Transductive propagation network for few-shot learning," in *ICLR*, 2019.

[19] Y. Hu, V. Gripon, and S. Pateux, "Leveraging the Feature Distribution in Transfer-based Few-Shot Learning," *arXiv preprint arXiv:2006.03806*, 2020.

[20] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *NeurIPS*, 2013.

[21] M. Ren *et al.*, "Meta-learning for semi-supervised few-shot classification," in *ICLR*, 2018.

[22] Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. van der Maaten, "Simpleshot: Revisiting nearest-neighbor classification for few-shot learning," *arXiv preprint arXiv:1911.04623*, 2019.

[23] E. Triantafillou *et al.*, "Meta-dataset: A dataset of datasets for learning to learn from few examples," *ICLR*, 2020.
[24] H.-Y. Tseng, H.-Y. Lee, J.-B. Huang, and M.-H. Yang, "Cross-Domain Few-Shot Classification via Learned Feature-Wise Transformation," in *ICLR*, 2020.
[25] Y. Guo *et al.*, "A broader study of cross-domain few-shot learning," in *ECCV*, 2020, pp. 124–141.
[26] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A Closer Look at Few-shot Classification," in *ICLR*, 2019.
[27] J. Cai and S. M. Shen, "Cross-domain few-shot learning with meta fine-tuning," *arXiv preprint arXiv:2005.10544*, 2020.
[28] H. Li *et al.*, "Rethinking the hyperparameters for fine-tuning," *ICLR*, 2020.
[29] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, and R. Feris, "Spottune: transfer learning through adaptive fine-tuning," in *CVPR*, 2019.
[30] A. Kolesnikov *et al.*, "Big transfer (bit): General visual representation learning," *ECCV*, 2020.
[31] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.
[32] D.-H. Lee and others, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, 2013, vol. 3, no. 2.
[33] Y. Grandvalet, Y. Bengio, and others, "Semi-supervised learning by entropy minimization.," in *CAP*, 2005, pp. 281–296.
[34] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *NeurIPS*, 2017.
[35] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *ICLR*, 2017.
[36] V. Perrone, R. Jenatton, M. Seeger, and C. Archambeau, "Scalable hyperparameter transfer learning," in *NeurIPS*, 2018, pp. 6846–6856.
[37] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.
[38] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *ICLR*, 2019.
[39] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Technical Report TR-2009, 2009.
[40] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, CNS-TR-2011-001, 2011.
[41] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *ICLR*, 2019.
[42] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *CVPR Workshops*, 2020.
[43] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.
[44] Z. Yu, L. Chen, Z. Cheng, and J. Luo, "Transmatch: A transfer-learning scheme for semi-supervised few-shot learning," in *CVPR*, 2020.
[45] X. Li *et al.*, "Learning to self-train for semi-supervised few-shot classification," in *NeurIPS*, 2019.

[46] I. M. Ziko, J. Dolz, E. Granger, and I. B. Ayed, "Laplacian Regularized Few-Shot Learning," in *ICML*, 2020.
[47] P. Rodriguez, I. Laradji, A. Drouni, and A. Lacoste, "Embedding propagation: smoother manifold for few-shot classification," in *ECCV*, 2020.
[48] Y. Wang, C. Xu, C. Liu, L. Zhang, and Y. Fu, "Instance Credibility Inference for Few-Shot Learning," in *CVPR*, 2020.
[49] B. Oreshkin, P. R. López, and A. Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," in *NeurIPS*, 2018.
[50] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *CVPR*, 2019.
[51] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *ICLR*, 2018.
[52] C. Zhang, Y. Cai, G. Lin, and C. Shen, "DeepEMD: Few-Shot Image Classification with Differentiable Earth Mover's Distance and Structured Classifiers," in *CVPR*, 2020.
[53] A. A. Rusu *et al.*, "Meta-learning with latent embedding optimization," in *ICLR*, 2019.
[54] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, and V. N. Balasubramanian, "Charting the right manifold: Manifold mixup for few-shot learning," in *WACV*, 2020.
[55] G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto, "A baseline for few-shot image classification," in *ICLR*, 2020.
[56] L. Qiao, Y. Shi, J. Li, Y. Wang, T. Huang, and Y. Tian, "Transductive episodic-wise adaptive metric for few-shot learning," in *ICCV*, 2019.
[57] C. Simon, P. Koniusz, R. Nock, and M. Harandi, "Adaptive subspaces for few-shot learning," in *CVPR*, 2020.
[58] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *CVPR*, 2020.
[59] M. Lichtenstein, P. Sattigeri, R. Feris, R. Giryes, and L. Karlinsky, "Tafssl: Task-adaptive feature sub-space learning for few-shot classification," in *ECCV*, 2020.
[60] J. Liu, L. Song, and Y. Qin, "Prototype rectification for few-shot learning," *arXiv preprint arXiv:1911.10713*, 2019.
[61] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *ICLR*, 2018.
[62] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *ICCV*, 2019, pp. 1406–1415.
[63] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing Textures in the Wild," in *CVPR*, 2014.
[64] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi, "Fine-Grained Visual Classification of Aircraft," in *arXiv:1306.5151*.
[65] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, vol. 32, pp. 323–332, 2012.
[66] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *CVPR*, 2018.
[67] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in plant science*, vol. 7, p. 1419, 2016.
[68] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected*

Topics in Applied Earth Observations and Remote Sensing, vol. 12, no. 7, pp. 2217–2226, 2019.
[69] N. Codella *et al.*, "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)," *arXiv preprint arXiv:1902.03368*, 2019.
[70] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *CVPR*, 2017, pp. 2097–2106.
[71] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D Object Representations for Fine-Grained Categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.
[72] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *NeurIPS*, 2014, pp. 487–495.
[73] G. Van Horn *et al.*, "The inaturalist species classification and detection dataset," in *CVPR*, 2018, pp. 8769–8778.
[74] H. Zhang, L. Zhang, X. Qi, H. Li, P. H. S. Torr, and P. Koniusz, "Few-shot Action Recognition with Permutation-invariant Attention," *ECCV 2020*, Aug. 2020, [Online]. Available: http://arxiv.org/abs/2001.03905.
[75] S. Tan and R. Yang, "Learning Similarity: Feature-Aligning Network for Few-shot Action Recognition," in *2019 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2019, pp. 1–7, doi: 10.1109/IJCNN.2019.8851694.
[76] S. K. Dwivedi, V. Gupta, R. Mitra, S. Ahmed, and A. Jain, "ProtoGAN: Towards Few Shot Learning for Action Recognition," *ICCVw 2019*, [Online]. Available: http://arxiv.org/abs/1909.07945.
[77] Y. Bo, Y. Lu, and W. He, "Few-Shot Learning of Video Action Recognition Only Based on Video Contents," in *WACV*, 2020.
[78] X. Zhu, A. Toisoul, J.-M. Perez-Rua, L. Zhang, B. Martinez, and T. Xiang, "Few-shot Action Recognition with Prototype-centered Attentive Learning," *arXiv:2101.08085*, Mar. 2021, [Online]. Available: http://arxiv.org/abs/2101.08085.
[79] C. Cao, Y. Li, Q. Lv, P. Wang, and Y. Zhang, "Few-shot Action Recognition with Implicit Temporal Alignment and Pair Similarity Optimization," *arXiv:2010.06215 [cs]*, Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.06215.
[80] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast Networks for Video Recognition," *ICCV 2019*, Oct. 2019, [Online]. Available: http://arxiv.org/abs/1812.03982.
[81] C. Feichtenhofer, "X3D: Expanding Architectures for Efficient Video Recognition," *CVPR 2020*, Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.04730.

LIST OF ACRONYMS

ARN	Action relation network
BD-CSPN	Cosine similarity based prototypical network with a bias diminishing module
BN	Batchnorm
BO	Bayesian optimization
CC	Cosine classifier
CIPA	Calibrated iterative prototype adaptation
DA	Domain adaptation
DAAML	Domain-adaptive active meta-learning
DSN-MR	Deep subspace networks with mean refinement
EMD	Earth mover distance
EPNet	Embedding propagation networks
FAN	Feature aligning network
FEAT	Few-shot embedding adaptation with transformer
FSL	Few-shot learning
FT	Finetuning
GAN	Generative adversarial networks
GNN	Graph neural networks
GPU	Graphics processing unit
HCT	Hybrid consistency training
HCT_R	Hybrid consistency training with rotation-based self-supervision
ICA	Independent component analysis
ICI	Instance credibility inference
ITA	Implicit temporal alignment
LEO	Latent embedding optimization
LFT	Learned feature-wise transformation
LR	Logistic regression
MAP	Modular adaptation pipeline
MSP	Mean-shift propagation
PAL	Prototype-centered attentive learning
PN/ProtoNet	Prototypical networks
PT	Power transform
S2M2	Self-supervised manifold mixup
$S2M2_R$	Self-supervised manifold mixup with rotation-based self-supervision
SGD	Statistic gradient descent
SNAIL	Simple neural attentive learner
SOTA	State-of-the-art
SSL	Semi-supervised learning
TADAM	Task dependent adaptive metric
TAV	Temporal attention vectors
TEAM	Transductive episodic-wise adaptive metric
TPN	Transductive propagation network
WRN	Wide residual networks