SELF-TRAINED CENTROID CLASSIFIERS FOR SEMI-SUPERVISED CROSS-DOMAIN FEW-SHOT LEARNING

Hongyu Wang, Eibe Frank, Bernhard Pfahringer, Geoffrey Holmes
Department of Computer Science
University of Waikato
New Zealand
{hw168@students.,eibe@,bernhard@,geoff@}waikato.ac.nz

ABSTRACT

State-of-the-art cross-domain few-shot learning methods for image classification apply knowledge transfer by fine-tuning deep feature extractors obtained from source domains on the small labelled dataset available for the target domain, generally in conjunction with a simple centroid-based classification head. Semi-supervised learning during the meta-test phase is an obvious approach to incorporating unlabelled data into cross-domain few-shot learning, but semi-supervised methods designed for larger sets of labelled data than those available in few-shot learning appear to easily go astray when applied in this setting. We propose an efficient semi-supervised learning method that applies self-training to the classification head only and show that it can yield very consistent improvements in average performance in the Meta-Dataset benchmark for cross-domain few-shot learning when applied with contemporary methods utilising centroid-based classification.

1 INTRODUCTION

Supervised machine learning methods for cross-domain few-shot learning (CDFSL) are designed to be applicable in target domains for which only small amounts of labelled training data are available. Training a complex machine learning model such as a deep neural network from scratch on such "few-shot" data runs the risk of overfitting. CDFSL methods address this by transferring knowledge learned from other domains, so-called "source domains", into the few-shot target domain. This is challenging because the source domains may differ substantially from the target domain. In particular, this cross-domain setting is more challenging than the setting that is traditionally considered in the few-shot learning literature. Contemporary CDFSL methods (Wang et al., 2022; Li et al., 2022; Triantafillou et al., 2021; Li et al., 2021) generally apply knowledge transfer by fine-tuning pretrained deep feature extractors, used in conjunction with a simple nearest-centroid classifier (Mensink et al., 2013; Snell et al., 2017), on the target dataset. However, they do not attempt to exploit unlabelled data during learning. In scenarios where additional target domain instances are available but lack labels, semi-supervised learning offers the prospect of improved performance. However, common semi-supervised methods are designed for relatively large sets of labelled data (Laine & Aila, 2017; Tarvainen & Valpola, 2017; Chen et al., 2020) and can easily go astray using small labelled sets. There exist a number of semi-supervised few-shot learning methods that apply semi-supervised learning to meta-training (Ren et al., 2018; Bateni et al., 2022; Xu et al., 2022; Islam et al., 2021), but literature is lacking on semi-supervised learning applied to CDFSL at meta-test time, based on any given pretrained feature extractors, whether obtained with meta-training or not.

We propose an efficient semi-supervised learning method applicable to any pretrained feature extractors, that keeps the feature extractor fixed after fine-tuning it on the labelled data and applies a classic semi-supervised learning method known as self-training to the classification head—the nearest-centroid classifier—only. Full self-training is a semi-supervised learning method that leverages unlabelled instances through an iterative process (Rosenberg et al., 2005): 1) train a model using the labelled dataset, 2) pseudo-label unlabelled instances with the trained model, and 3) update the labelled dataset with the unlabelled instances and their pseudo-labels. The labelled dataset consists of only labelled instances during the first iteration of training and additionally includes unlabelled instances with their pseudo-labels in all following iterations. The train-label loop iterates until a stopping criterion is met.

The training step in this full self-training loop involves optimising all trainable parameters in the model and can become time-consuming if the feature extractor is heavily parameterised. More importantly, in few-shot learning, the small labelled dataset may provide insufficient guidance to reliably update such a large number of parameters in self-training. We address both issues by applying self-training to the centroid classifier only, yielding self-trained centroids (STC) for cross-domain few-shot learning. In this approach, feature vectors of labelled and unlabelled instances are

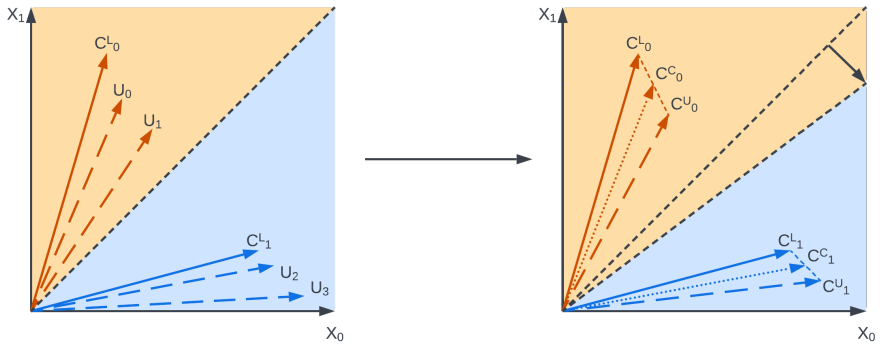

Figure 1: Visualisation of STC. In the left diagram, labelled centroids (solid vectors, C_0^L and C_1^L) are used to soft-label unlabelled feature vectors (dashed vectors, U_0 - U_3) with cosine similarity distance. The black dashed line represents the decision boundary. In the right diagram, "labelled" and "unlabelled" centroids are averaged to produce combined centroids (dotted lines, C_0^C and C_1^C). This results in a shift in the decision boundary.

extracted using the fine-tuned feature extractor, and the labelled feature vectors are used to compute initial centroids. Subsequently, these centroids are used in the nearest-centroid classifier to assign soft labels to the unlabelled feature vectors. Once these soft labels have been obtained, another set of centroids can be computed from the pseudo-labelled feature vectors. The two sets of centroids are averaged on a per-class basis to produce combined centroids, as shown in Figure 1. To form an iterative process, the new centroids can be used to soft-label the unlabelled instances again, for "labelled" and "unlabelled" centroids to again be averaged. To predict test instances, the combined centroids are used in the nearest-centroid classifier.

We apply STC to contemporary CDFSL methods, including URL (Li et al., 2021), FLUTE (Triantafillou et al., 2021), TSA (Li et al., 2022), as well as their counterparts in the recently proposed ConFES framework (Wang et al., 2022) that applies an ensemble of feature extractors. We evaluate them using an extended form of the Meta-Dataset benchmark (Triantafillou et al., 2020; Requeima et al., 2019; Wang et al., 2022) and show that STC used with 1,000 unlabelled instances improves average performance very consistently. We also demonstrate that STC is more efficient than full self-training based on updating all model parameters and performs better in cross-domain scenarios.

2 Semi-supervised CDFSL with STC

A semi-supervised few-shot learning episode contains, as its training data, a labelled set L and an unlabelled set U. L is of size N, containing instances $X \in \mathbb{R}^{I \times N}$ with I input dimensions and their labels $Y \in [0, C)^N$ belonging to classes C. $U \in \mathbb{R}^{I \times M}$ contains M instances also belonging to C but their labels are unknown. The learning task is to fit a model, pretrained on source domains, using L and U, and evaluate it using a separate test/query set Q. The goal of a semi-supervised CDFSL method is to learn from L and U and outperform its supervised counterpart learning from only L. We first formulate the STC learning algorithm for cases where it is applied with a single feature extractor and then explain how it can be used in ConFES ensembles.

Given a feature extractor Φ, we fine-tune it on L by using a nearest-centroid classifier as the classification head. In CDFSL, common pretraining methods for obtaining the feature extractor include vanilla pretraining on a single source domain (Wang et al., 2022), knowledge distillation (Li et al., 2021), and universal template training (Triantafillou et al., 2021). Common fine-tuning methods include feature projection (Li et al., 2021), batch normalisation fine-tuning (Triantafillou et al., 2021), and task-specific adaptors (Li et al., 2022). All these methods can be used in conjunction with STC. To this end, we use the fine-tuned Φ' to extract feature vectors $X^F \in \mathbb{R}^{J \times N}$ with J feature dimensions from X, and $U^F \in \mathbb{R}^{J \times M}$ from U. We compute class centroids $X^C \in \mathbb{R}^{J \times C}$ for the labelled set using X^F and Y, with S_j representing the set of indices of labelled instances belonging to a particular class j:

$$X_j^C = \frac{1}{|S_j|} \sum_{i \in S_j} X_i^F, S_j = \{k : Y_k = j\}, j = 1, ..., C. \tag{1}$$

Once we have the set of centroid vectors, X^C, we can use them to assign soft labels to U^F by applying a similarity measure s and the softmax function. For each unlabelled feature vector U_i^F, its soft labels P_i^U are computed as:

$$P_i^U(Y_i^U = j|U_i^F) = \frac{e^{s(U_i^F, X_j^C)}}{\sum_{i=1}^{C} e^{s(U_i^F, X_i^C)}}. \tag{2}$$

Here, we use the same s as the one applied in the nearest-centroid classifier during fine-tuning, i.e., the exact formulation depends on the fine-tuning method that is applied. Li et al. (2021) and Li et al. (2022) use cosine similarity scaled by a factor of 10, while Triantafillou et al. (2021) use cosine similarity without scaling.

Once soft labels have been obtained, we can compute class centroids $U^C \in \mathbb{R}^{J \times C}$ for the unlabelled set based on U^F and the corresponding soft labels P^U by using these soft labels to calculate weighted averages:

$$U_j^C = \frac{\sum_{i=1}^{|U^F|} P_i^U(Y_i^U = j|U_i^F) \cdot U_i^F}{\sum_{i=1}^{|U^F|} P_i^U(Y_i^U = j|U_i^F)}. \tag{3}$$

The final set of centroids $C^C \in \mathbb{R}^{J \times C}$ is subsequently obtained as a simple arithmetic average of X^C and U^C, giving equal weight to the centroids from the data with ground-truth labels and the centroids from the pseudo-labelled data:

$$C_j^C = \frac{X_j^C + U_j^C}{2}. \tag{4}$$

An iterative self-training process can be formed to further refine the soft labels by then using C^C instead of X^C to soft-label U^F, i.e., replacing X^C with C^C in Equation 2, and repeating Equations 2 - 4. However, interestingly, in line with the findings of Ren et al. (2018), whose method we discuss in the next section, we found that simply performing Equations 1 - 4 once is often sufficient: iterating the process leads to relatively little benefit overall and may even harm in some cases, despite it being common procedure in full self-training with larger datasets (Rosenberg et al., 2005).

After training, predictions for a query instance Q_i are made with its feature vector Q_i^F and the final centroids C^C:

$$P_i^Q(Y_i^Q = j|Q_i^F) = \frac{e^{s(Q_i^F, C_j^C)}}{\sum_{i=1}^{C} e^{s(Q_i^F, C_i^C)}}. \tag{5}$$

The above description of STC is based on a single feature extractor. However, recent work has shown that the classic stacking approach to ensemble learning, in the form of feature extractor stacking (Wang et al., 2022), can be used to obtain state-of-the-art accuracy on CDFSL problems: a meta-classifier, e.g., a very simple convolutional neural network, is trained using cross-validated predictions obtained from a set of source domain backbones (i.e., "base models") during fine-tuning and learns to appropriately combine predictions from all the snapshots available. Fortunately, it is straightforward to apply STC in conjunction with this stacking approach. Given a meta-classifier that takes fine-tuned base model logits as input, in order to re-purpose it with STC for semi-supervised learning, we simply apply STC to each fine-tuned base model and replace the logits normally obtained from the centroid classifier based on supervised training with logits obtained from STC.

3 Related work

There appears to be comparatively little work on semi-supervised cross-domain few-shot learning. Ren et al. (2018) do not consider cross-domain learning but do investigate semi-supervised learning with prototypical networks (Snell et al., 2017) by soft-labelling unlabelled instances and adjusting the prototypes/centroids with them. As a prototypical network is pretrained using few-shot episodes sampled from source domains—a process also called "meta-training" in this context—these episodes are modified to contain unlabelled instances , but evaluation (also referred to as the "meta-test" phase) is performed "in-domain", i.e., the source and target domains are different class partitions of the same domain. Note that soft labels are essential to facilitate backpropagation in a prototypical network's training.

In contrast to the method proposed in Ren et al. (2018), STC is myopic to, and separated from, a feature extractor's pretraining and fine-tuning, and thus compatible with any feature extractor that applies nearest-centroid classification.

The semi-supervised learning step in STC means that it can be applied with soft or hard labels—but soft labels are generally preferable in CDFSL given high uncertainty in some unlabelled instances. Lastly, STC is designed for cross-domain tasks, which is arguably more relevant for practical applications: Guo et al. (2020) showed that several meta-learning approaches at the time, including prototypical networks, underperform in CDFSL settings.[1]

A number of semi-supervised few-shot learning methods pretrain (or "meta-train") their feature extractors using semi-supervised learning like in Ren et al. (2018): Transductive CNAPS (Bateni et al., 2022) uses query instances for feature adaptation, GCT (Xu et al., 2022) converts instances into graph nodes aided by unlabelled instances, Dynamic Distillation (Islam et al., 2021) uses augmented unlabelled instances to fit a teacher-student network pair, and Li & Zhang (2021) create additional meta-training tasks for meta-learners with large language models using vocabulary tokenisation and self-supervision. In contrast, in this paper, we evaluate all semi-supervised learning methods using feature extractors pretrained in a supervised manner, thus anticipating a scenario that is likely to occur in practical applications, and focus on the effect of semi-supervised learning in the meta-test phase. We compare STC to several generic semi-supervised methods applied in this setting, including full self-training (Rosenberg et al., 2005), which fits all trainable parameters to soft-labelled instances, Pi-Model (Laine & Aila, 2017), which optimises consistency between logits of different copies of augmented unlabelled instances, Temporal Ensembling (Laine & Aila, 2017), which optimises consistency between logits of augmented unlabelled instances and their exponential moving average, Mean Teacher (Tarvainen & Valpola, 2017), which uses augmented unlabelled instances to optimise consistency between a student model and a teacher model given by an exponential moving average of previous students, and SimCLR (Chen et al., 2020), which optimises agreement between projections of feature vectors of different copies of augmented unlabelled instances. Among the methods that implement semi-supervised meta-training for the feature extractor, Transductive CNAPS accommodates multiple source domains, which makes it applicable to our evaluation scenario, so is included in our comparison.

We now briefly review the supervised CDFSL methods that we consider in our experiments with STC, namely URL (Li et al., 2021), FLUTE (Triantafillou et al., 2021), TSA (Li et al., 2022), and FES (Wang et al., 2022):

- URL distils a universal feature extractor from a base feature extractor collection by matching the universal model's feature and logit outputs on each source domain's instances to those of a base feature extractor pretrained on the same source domain. After distillation, the universal feature extractor is used in a CDFSL episode by fitting a linear projection in conjunction with a nearest-centroid classifier to labelled feature vectors. The universal feature extractor remains fixed.
- TSA builds on URL but fits the universal feature extractor to an episode by attaching channel-wise projections as adaptors to its convolutional layers, as well as a linear projection to its feature output. The channel-wise and feature projections are fitted to the labelled set in conjunction with a nearest-centroid classifier.
- FLUTE meta-trains a universal template model with a universal set of convolutional parameters and multiple sets of batch normalisation parameters, one for each source domain, by fitting each set of batch normalisation parameters to its respective source domain, while the universal convolutional parameters are fitted to all source domains. A separate encoder network is trained to predict a training set's likeness to each source domain. Given a CDFSL episode, the encoder produces a linear combination of source domains based on the labelled set, and this combination is used to aggregate the sets of batch normalisation parameters into a single set as a weighted average, which is then fitted to the labelled set in conjunction with a nearest-centroid classifier with the universal convolutional parameters fixed.
- FES fits a pretrained feature extractor collection to a CDFSL episode by training a meta-classifier using stacking with cross-validation on the episode's labelled training set. The labelled set is split into two partitions using stratified cross-validation. Each feature extractor in the collection is fitted to one partition using a user-specified fine-tuning method, with snapshots saved at different iterations, and these snapshots are used to extract logits on the other partition. The process is performed twice with the partitions switching roles to obtain logits from both partitions, which are used with their true labels to train a meta-classifier that weighs the snapshots and produces meta-logits as a weighted average of base logits. For classification, the feature extractor collection is fine-tuned on the full labelled set, their snapshots saved, and these snapshots are used to extract logits from query instances. The logits are aggregated by the trained meta-classifier to produce its predictions. Convolutional FES (ConFES) is a variant of FES that replaces the meta-classifier's flat weight kernel with a multi-level kernel hierarchy that is 1D-convolutional in the dimension of snapshot iterations. The ConFES kernels are connected directly without non-linear activations, which allows them to be expanded

[1]Note that Ren et al. (2018) considered an "inference only" baseline in their experiments, where the feature extractor received supervised pretraining, and unlabelled instances were only used to adjust centroids, which is comparable to STC. We consider STC to be a generalisation of this baseline and show that it can be applied during meta-test to various centroid-based CDFSL methods.

Table 1: Count and average size of unlabelled sets with fewer than 1000 instances.

dataset	sub-1000 unlabelled count	sub-1000 unlabelled average size
ilsvrc_2012	0	-
omniglot	600	96.7
aircraft	554	523.3
cu_birds	553	487.6
dtd	600	369.1
quickdraw	0	-
fungi	258	462.6
vgg_flower	564	508.9
traffic_sign	0	-
mscoco	0	-
mnist	0	-
cifar10	0	-
cifar100	145	531.3
CropDisease	0	-
EuroSAT	0	-
ISIC	0	-
ChestX	0	-
Food101	9	878.7

back into an equivalent flat FES kernel while maintaining fewer parameters than FES. ConFES is the strongest variant of FES evaluated in the experiments in Wang et al. (2022), which is why we use it in this paper.

We apply STC to these methods and evaluate them on the Meta-Dataset benchmark (Triantafillou et al., 2020). Meta-Dataset originally contained eight source domains: ilsvrc_2012, omniglot, aircraft, cu_birds, dtd, quickdraw, fungi, and vgg_flower, and two target domains: traffic_sign and mscoco. Requeima et al. (2019) added three additional target domains: mnist, cifar10, and cifar100, and Wang et al. (2022) added a further five target domains: CropDisease, EuroSAT, ISIC, ChestX, and Food101.

4 Experimental setup

Meta-Dataset produces a supervised few-shot episode for training and evaluating a few-shot learner by first sampling several classes from the test split of a dataset, and then sampling labelled training and test instances from these classes (Triantafillou et al., 2020). We follow the official specifications and sample episodes each containing 5 to 50 classes, with up to 500 labelled (potentially class-imbalanced) instances in total, as well as 10 query instances per class. To enable semi-supervised learning, we pool the remaining instances in the sampled classes that have not been selected as training or test instances, and randomly select 1000 instances from the pool as the unlabelled set of the episode based on the assumption that obtaining 1000 unlabelled instances per task is generally achievable in practical CDFSL scenarios. The 1000 unlabelled instances can potentially be class-imbalanced. In some cases where the classes are small, there may be fewer than 1000 instances in the pool, and the entire pool is used as the unlabelled set. Like Wang et al. (2022), we cache sampled semi-supervised CDFSL episodes, and use the same cached episodes to evaluate all methods, which avoids variance between sampling runs and facilitates paired t-tests. Paired testing increases statistical power compared to unpaired methods by considering the difference in accuracy on a per-episode basis. Table 1 shows the number of episodes, out of 600 sampled per dataset, that failed to obtain 1000 unlabelled instances, and the average size of those particular episodes. Triantafillou et al. (2021) pointed out that Meta-Dataset instances need to be shuffled during sampling in case a dataset has a particular ordering, e.g., consecutive images may be from the same video, and implemented this as a shuffling window of size 1000 for instance streams. We noticed that this window is not big enough for datasets like ChestX, leading to more frequent leaks of same-patient data between training and test sets than true random sampling, which makes an algorithm's performance approximately 3% better on ChestX than with true random sampling. It also causes a 1% accuracy difference in mscoco. Hence, we use true random sampling for our experiments.

We first evaluate four well-known semi-supervised learning methods for large labelled datasets from the literature: Pi-Model, Temporal Ensembling, Mean Teacher, and SimCLR, and compare them to full self-training, by applying all of them to URL (Li et al., 2021), i.e., a linear projection fitted to feature vectors extracted by a fixed universal

feature extractor, used in conjunction with a nearest-centroid classifier. The universal ResNet18 feature extractor is downloaded from the official URL repository. The linear projection in URL is treated as the optimisable parameter set. We used the first 20 episodes of each source domain to tune the hyperparameters of these methods, which led to a multiplier of 100 for consistency loss in Pi-Model, Temporal Ensembling, and Mean Teacher, an α of 0.5 for the exponential moving average in Temporal Ensembling and Mean Teacher, and a multiplier of 1 for agreement loss in SimCLR. For the consistency loss, we found that taking the mean instead of the sum of the squared differences of the logits leads to more stable performance, presumably due to the varying number of classes in different episodes. We also evaluate Transductive CNAPS with our cached episodes. We use the hyperparameters from (Bateni et al., 2022), and the ResNet18 checkpoint downloaded from the official repository, reporting results using the query set as unlabelled data for transduction. We found that including the unlabelled set as additional data degraded performance.

Following this, we thoroughly evaluate the STC method by applying it to URL (Li et al., 2021), FLUTE (Triantafillou et al., 2021), and TSA (Li et al., 2022), as well as their counterparts in a two-level ConFES ensemble (Wang et al., 2022). All fine-tuning processes and hyperparameters are kept consistent with the original papers, all feature extractors are ResNet18 models downloaded from the official sources, and STC is simply applied to the feature extractors post-fine-tuning. A single design choice was made for STC: a plain arithmetic average is used to aggregate labelled and unlabelled centroids instead of an average weighted by the number of instances. This was based on the intuition that a weighted average can be overwhelmed by a large number of noisy unlabelled instances and lead to instability, and a few source domain episodes were sufficient to confirm this. Non-iterative results, obtained after the first iteration of self-training in STC, are presented in tables and compared to non-iterative full self-training for URL, FLUTE, and TSA, but not their ConFES counterparts, because performing non-iterative full self-training with ConFES is prohibitively expensive computation-wise. We also show plots visualising the accuracy of STC across 20 iterations.

5 Results

We first show that semi-supervised algorithms for large labelled datasets may not be well-suited for CDFSL, and justify this claim by showing better performance of simple non-iterative full self-training when using URL as the case study. We then present results obtained by applying non-iterative STC to a range of state-of-the-art CDFSL methods, and show that it achieves improved performance over supervised learning and full self-training. Lastly, we present accuracy-over-iteration plots for iterative STC.

5.1 Common semi-supervised algorithm in CDFSL

Table 2 shows common semi-supervised methods applied to URL, and compares them to the supervised approach. For each dataset, mean accuracy of 600 few-shot episodes is reported, along with the 95% confidence interval. Results are averaged for source and target domains separately, as only target domain tasks are truly cross-domain, and their accuracy represents "strong generalisation" (SG) performance; while source domains represent "weak generalisation" (WG). The best results for each dataset are marked **bold**.

Only non-iterative full self-training achieves greater average accuracy than supervised URL in terms of SG performance, while Pi-Model, Temporal Ensembling, Mean Teacher, SimCLR, and iterative full self-training all perform worse. Transductive CNAPS achieves top accuracy in several target domains but its average SG performance is not as strong as that of the URL-based methods. Non-iterative full self-training always yielding higher accuracy than its iterative counterpart in SG indicates that full self-training, which is commonly iterative in the literature, may exhibit instability in CDFSL as more iterations are performed. Overall, the positive results for full self-training provide the motivation for investigating the more efficient and, as it turns out, more robust STC algorithm, which only applies self-training to the centroids, in the following sections.

5.2 STC CDFSL evaluations

Tables 3, 4, and 5 show non-iterative STC performance—for URL, FLUTE, and TSA respectively—compared to that of supervised learning and non-iterative full self-training. In each of the three tables, the five columns show results for 1) the supervised base algorithm ("base"), 2) full self-training applied to the base algorithm ("base-FST"), 3) the supervised ConFES ensemble of the base algorithm ("ConFES"), 4) STC applied to the base algorithm ("base-STC"), and 5) STC applied to ConFES ("ConFES-STC"). Like before, mean accuracy is reported with the 95% confidence interval. A Wilcoxon-Holm test (Demsar, 2006) is performed to compute mean WG and SG ranks using individual episode accuracy values. Paired t-tests, which are generally more sensitive than 95% confidence intervals, are performed to compute the p value between the methods using their accuracy values in all individual episodes. A p smaller than 0.05 is deemed to indicate a statistically significant difference. Among the columns, "base" and "base-

Table 2: Pi-Model, Temporal ensembling, Mean Teacher, SimCLR, non-iterative (1 iteration) and iterative (20 iterations) full self-training applied to URL with 1000 unlabelled instances, compared to supervised URL. Transductive CNAPS results are provided in the rightmost column.

URL	Sup	Pi	TE	MT	SCLR	ST-1	STC-20	T-CNAPS
ilsvrc_2012	**56.6±1.1**	56.5±1.1	**56.6±1.1**	**56.6±1.1**	56.4±1.1	**56.6±1.1**	**56.6±1.1**	55.8±1.1
omniglot	94.5±0.4	94.5±0.4	94.5±0.4	94.5±0.4	94.4±0.4	**95.1±0.4**	**95.1±0.3**	93.4±0.5
aircraft	87.7±0.5	87.5±0.5	87.5±0.5	87.6±0.5	87.3±0.5	87.8±0.5	**88.1±0.4**	82.1±0.7
cu_birds	80.7±0.7	80.9±0.7	80.9±0.7	80.8±0.7	80.6±0.7	81.2±0.7	**81.3±0.6**	77.7±0.8
dtd	**76.1±0.6**	75.8±0.7	75.8±0.6	75.9±0.6	75.6±0.6	75.8±0.6	76.0±0.6	68.3±0.7
quickdraw	82.0±0.6	81.9±0.6	82.0±0.6	82.0±0.6	81.9±0.6	82.4±0.6	**82.7±0.6**	78.2±0.7
fungi	69.5±1.1	69.2±1.0	69.4±1.0	69.3±1.0	69.5±1.1	70.7±1.0	**71.2±1.0**	50.0±1.3
vgg_flower	91.4±0.5	91.4±0.5	91.4±0.5	91.4±0.5	91.3±0.5	91.8±0.5	**92.1±0.4**	91.3±0.5
mean WG acc	79.8	79.7	79.8	79.8	79.6	80.2	**80.4**	74.6
traffic_sign	**62.6±1.2**	**62.6±1.2**	62.0±1.1	61.3±1.1	62.0±1.2	62.3±1.2	61.5±1.2	57.3±1.1
mscoco	52.7±1.0	52.3±1.0	52.8±1.0	52.7±1.0	53.0±1.0	**53.9±1.0**	53.5±0.9	48.5±1.0
mnist	94.6±0.4	94.5±0.5	94.0±0.4	93.8±0.5	94.2±0.4	94.7±0.4	92.2±1.0	**95.3±0.3**
cifar10	71.4±0.8	71.0±0.8	71.3±0.8	71.1±0.8	71.6±0.8	71.7±0.8	71.3±0.8	**72.1±0.7**
cifar100	62.6±1.1	62.4±1.1	62.6±1.1	62.4±1.1	62.5±1.1	**63.0±1.1**	62.6±1.1	61.3±1.1
CropDisease	80.5±0.8	80.8±0.8	80.9±0.8	78.8±0.8	80.0±0.8	**81.0±0.8**	80.5±0.8	79.4±0.8
EuroSAT	86.5±0.5	**86.7±0.5**	85.3±0.5	85.8±0.5	86.4±0.5	86.6±0.5	85.8±0.6	78.9±0.6
ISIC	45.5±0.8	45.3±0.8	44.1±0.8	44.0±0.8	45.6±0.8	**46.9±0.9**	46.8±0.9	44.1±0.8
ChestX	26.6±0.6	26.5±0.6	26.4±0.5	26.6±0.6	26.5±0.6	26.8±0.6	26.8±0.6	**27.1±0.6**
Food101	51.9±1.1	51.4±1.0	52.0±1.1	51.6±1.1	**52.2±1.1**	52.1±1.1	51.8±1.1	51.2±1.1
mean SG acc	63.5	63.4	63.1	62.8	63.4	**63.9**	63.3	61.5

Table 3: Comparison of STC, supervised learning, and non-iterative full self-training using URL.

URL	base	base-FST	ConFES	base-STC	ConFES-STC	
ilsvrc_2012	56.6±1.1	56.6±1.1	56.0±1.2	**56.7±1.1**	55.9±1.2	−
omniglot	94.5±0.4 •	**95.1±0.4** •	93.9±0.6 •	**95.1±0.4**	94.6±0.5	−
aircraft	87.7±0.5 •	87.8±0.5	87.4±0.7 •	**87.9±0.5**	87.7±0.6	
cu_birds	80.7±0.7 •	81.2±0.7 •	79.0±0.8	**81.3±0.7**	79.2±0.8	−
dtd	**76.1±0.6** ○	75.8±0.6	74.7±0.8 ○	75.8±0.6	74.3±0.8	−
quickdraw	82.0±0.6 •	82.4±0.6 •	83.1±0.6 •	82.6±0.6	**83.5±0.6**	+
fungi	69.5±1.1 •	70.7±1.0 •	69.9±1.1 •	71.0±1.0	**71.2±1.1**	
vgg_flower	91.4±0.5 •	91.8±0.5 •	90.6±0.7	**91.9±0.4**	90.7±0.7	−
mean WG acc	79.8	80.2	79.3	**80.3**	79.6	
mean WG rank	3.19	2.91	3.09	**2.85**	2.95	
traffic_sign	62.6±1.2	62.3±1.2 •	66.1±1.2 •	62.6±1.2	**66.4±1.2**	+
mscoco	52.7±1.0 •	**53.9±1.0**	52.7±1.0 •	53.8±1.0	53.7±1.0	
mnist	94.6±0.4 •	94.7±0.4 •	96.5±0.5 •	95.1±0.4	**96.8±0.5**	+
cifar10	71.4±0.8 •	71.7±0.8 •	71.6±0.9 •	72.0±0.7	**72.1±0.9**	
cifar100	62.6±1.1 •	**63.0±1.1**	62.9±1.1 •	**63.0±1.1**	**63.0±1.1**	
CropDisease	80.5±0.8 •	81.0±0.8 •	87.2±0.7 •	81.4±0.8	**87.7±0.7**	+
EuroSAT	86.6±0.5 •	86.6±0.5 •	86.0±0.6 •	**86.9±0.5**	86.3±0.6	−
ISIC	45.5±0.8 •	46.9±0.9	48.2±0.9 •	46.7±0.9	**49.4±1.0**	+
ChestX	26.5±0.6	**26.8±0.6**	26.7±0.6	**26.8±0.6**	**26.8±0.6**	
Food101	51.9±1.1 •	52.1±1.1 •	54.0±1.1 •	52.3±1.1	**54.2±1.1**	+
mean SG acc	63.5	63.9	65.2	64.1	**65.6**	
mean SG rank	3.52	3.26	2.69	3.12	**2.41**	

FST" are compared to "base-STC", while "ConFES" is compared to "ConFES-STC". If $p < 0.05$, • indicates an algorithm's STC counterpart has better performance, and ○ indicates the algorithm performs statistically significantly better than its STC counterpart. In addition, "base-STC" is compared to "ConFES-STC". If $p < 0.05$, + indicates

Table 4: Comparison of STC, supervised learning, and non-iterative full self-training using FLUTE.

FLUTE	base	base-FST	ConFES	base-STC	ConFES-STC	
ilsvrc_2012	50.2±1.1 ○	50.8±1.1 ○	**54.1±1.2** ○	49.6±1.1	53.9±1.1	+
omniglot	93.9±0.5 ●	93.7±0.5 ●	94.9±0.5 ●	95.0±0.4	**95.9±0.4**	+
aircraft	86.8±0.6 ●	86.0±0.6 ●	87.0±0.9 ●	87.1±0.5	**87.5±0.6**	+
cu_birds	79.3±0.8 ●	78.5±0.8 ●	78.5±0.9	**79.8±0.7**	78.5±0.9	−
dtd	68.8±0.8 ○	68.0±0.7 ●	**74.3±0.9**	68.5±0.7	74.1±0.8	+
quickdraw	79.1±0.7	78.5±0.7 ●	**82.8±0.6**	79.0±0.7	**82.8±0.6**	+
fungi	59.4±1.2 ●	60.5±1.2 ●	69.2±1.1 ●	61.8±1.2	**70.6±1.0**	+
vgg_flower	91.0±0.6 ●	90.9±0.5 ●	92.5±0.6 ●	91.2±0.5	**92.7±0.6**	+
mean WG acc	76.1	75.9	79.2	76.5	**79.5**	
mean WG rank	3.45	3.75	2.26	3.33	**2.22**	
traffic_sign	57.9±1.1 ○	54.1±1.1 ●	**71.8±1.1** ○	55.9±1.1	71.6±1.1	+
mscoco	48.2±1.0	48.6±1.0 ○	51.9±1.1 ●	48.3±1.0	**52.8±1.0**	+
mnist	95.7±0.4 ●	95.0±0.4 ●	97.6±0.4 ●	96.2±0.3	**97.9±0.3**	+
cifar10	78.6±0.7 ●	**79.0±0.7**	75.2±0.9 ●	**79.0±0.7**	75.4±0.9	−
cifar100	67.5±1.0 ●	67.4±1.0 ●	66.9±1.1 ○	**67.7±1.0**	66.6±1.0	−
CropDisease	78.0±0.8 ●	78.6±0.8 ○	86.2±0.7 ●	78.2±0.8	**86.7±0.6**	+
EuroSAT	81.6±0.6 ○	79.9±0.6 ●	**88.1±0.6**	80.8±0.6	**88.1±0.6**	+
ISIC	46.1±1.0 ●	48.7±0.9 ●	48.7±1.0 ●	49.0±0.9	**51.3±0.9**	+
ChestX	26.3±0.5	26.4±0.5 ○	27.3±0.6 ●	26.1±0.5	**27.8±0.6**	+
Food101	45.7±1.1 ○	46.7±1.1 ○	**51.9±1.1**	45.5±1.1	51.8±1.1	+
mean SG acc	62.6	62.4	66.6	62.7	**67.0**	
mean SG rank	3.46	3.55	2.33	3.44	**2.22**	

Table 5: Comparison of STC, supervised learning, and non-iterative full self-training using TSA.

TSA	base	base-FST	ConFES	base-STC	ConFES-STC	
ilsvrc_2012	56.8±1.1 ●	56.8±1.1 ●	56.3±1.2 ○	**57.2±1.1**	56.0±1.2	−
omniglot	95.0±0.4 ●	95.3±0.4 ●	93.4±0.7 ●	**95.7±0.3**	94.5±0.6	−
aircraft	88.4±0.5 ●	88.6±0.5 ●	87.8±0.8 ●	**88.8±0.5**	88.3±0.6	−
cu_birds	81.5±0.7 ●	81.8±0.7 ●	79.8±0.9	**82.2±0.7**	79.9±0.8	−
dtd	**77.1±0.7**	76.8±0.7	76.3±0.8 ○	77.0±0.6	75.9±0.8	−
quickdraw	82.0±0.6 ●	82.4±0.6 ●	83.4±0.6 ●	82.7±0.6	**83.8±0.6**	+
fungi	68.3±1.1 ●	69.0±1.0 ●	69.8±1.1 ●	70.0±1.0	**70.7±1.1**	+
vgg_flower	92.1±0.5 ●	92.3±0.5 ●	91.9±0.7 ●	**92.8±0.5**	92.2±0.6	−
mean WG acc	80.2	80.4	79.8	**80.8**	80.2	
mean WG rank	3.29	3.03	3.01	**2.74**	2.93	
traffic_sign	82.8±0.9 ●	84.0±0.9 ○	85.7±1.0 ●	83.8±0.9	**86.6±1.0**	+
mscoco	53.8±1.1 ●	53.9±1.1 ●	54.5±1.0 ●	54.7±1.0	**55.8±1.0**	+
mnist	96.6±0.4 ●	96.6±0.4 ●	97.1±0.5 ●	97.0±0.3	**97.4±0.5**	+
cifar10	79.9±0.8 ●	80.2±0.8 ●	78.3±0.9 ●	**80.4±0.7**	78.9±0.9	−
cifar100	70.3±1.0 ●	70.4±1.0 ●	70.7±1.1 ●	70.9±1.0	**71.2±1.0**	+
CropDisease	84.4±0.8 ●	85.0±0.8 ●	88.2±0.7 ●	85.6±0.7	**89.0±0.7**	+
EuroSAT	89.6±0.5 ●	**90.0±0.5**	89.2±0.6 ●	89.9±0.5	89.4±0.6	−
ISIC	48.4±0.9 ●	48.0±0.9 ●	48.9±1.0 ●	49.5±0.9	**50.6±1.0**	+
ChestX	27.2±0.6 ●	27.6±0.6	27.1±0.6 ●	27.6±0.6	**28.2±0.7**	+
Food101	53.4±1.2 ●	53.3±1.2 ●	55.2±1.1 ●	53.8±1.2	**55.5±1.1**	+
mean SG acc	68.6	68.9	69.5	69.3	**70.3**	
mean SG rank	3.55	3.25	2.86	2.97	**2.37**	

that the semi-supervised ConFES ensemble has better performance, while − indicates better performance for the semi-supervised base algorithm.

The results show that for URL and TSA, STC consistently exhibits greater estimated accuracy than supervised learning and full self-training in SG. For FLUTE, STC has better average SG performance but its relative performance varies

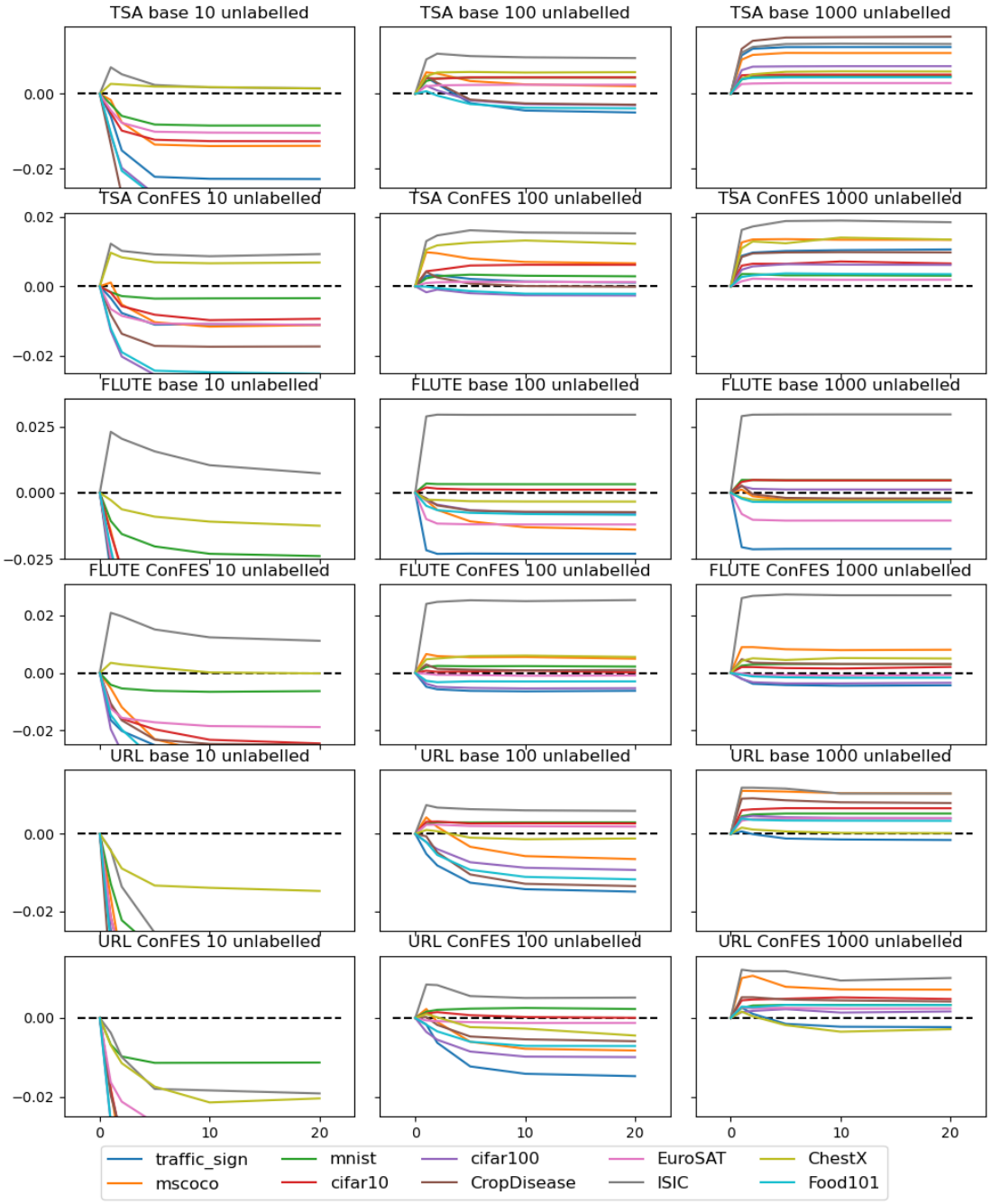

Figure 2: Iterative STC accuracy (relative to supervised) in 20 iterations given 10, 100, or 1000 unlabelled instances.

among datasets. ConFES with STC also consistently exhibits greater estimated SG accuracy than supervised ConFES for URL and TSA, as well as greater average SG accuracy for FLUTE. Finally, considering SG and analogously to the results for purely supervised learning in Wang et al. (2022), STC in a ConFES ensemble exhibits greater estimated accuracy than STC in the base algorithm counterpart. It is worth re-iterating that STC achieves greater estimated accuracy at minimal added computational cost: it only requires unlabelled feature vectors and their soft labels from forward propagation, while full self-training requires backpropagation for re-fitting after soft-labelling.

5.3 Iterative STC

Figure 2 shows how iterative STC SG performance changes over 20 iterations. Values depicted are differences between STC and supervised learning. Values greater than 0 indicate that STC achieves greater estimated accuracy than supervised learning, and values less than 0 indicate the opposite. Each row of figures represents a CDFSL method (a base algorithm or its ConFES counterpart), and each column represents a different unlabelled set size (10, 100, or 1000). The increased accuracy of non-iterative STC with 1000 unlabelled instances vs. supervised learning, discussed above and reported in Tables 3, 4, and 5, is reflected in the sharp change from iteration 0 (supervised accuracy) to iteration 1 (non-iterative STC accuracy) in the rightmost column of the figure.

With 1000 unlabelled instances, STC accuracy generally does not change significantly from iteration 1 to 20, which indicates that 1 iteration (non-iterative STC) is sufficient to achieve optimal performance, although there are small improvements for some target domains when applying STC iteratively using TSA. Clearly, STC is more stable than full self-training in CDFSL, as Table 2 shows that more iterations lead to worse SG performance for full self-training with the same 1000 unlabelled instances. Comparing the three columns in Figure 2, STC performs better with more unlabelled instances, as 1000 unlabelled instances lead to better performance in general, and especially consistently with TSA and its ConFES variant, while 10 unlabelled instances mostly lead to lower accuracy, with the worst drop being -0.06. (We cut off the display for better visualisation of a more densely-populated range.) The ISIC dataset appears to benefit most from STC in most settings, whereas traffic_sign reacts negatively to STC in multiple cases. STC generally shows a stronger tendency to decay over iterations when applied on fewer unlabelled instances. For datasets like mnist and traffic_sign, decay can be observed using FLUTE or URL even with 1000 unlabelled instances.

In general, STC performance tends to either remain stable or decay after the first iteration, so non-iterative STC is the safer option over iterative STC. However, iterative STC is stable with TSA and 1000 unlabelled instances, and small performance gains can be observed over the iterations on some target domains. As the results in Tables 3, 4, and 5 show that TSA achieves greater estimated accuracy than FLUTE and URL as a CDFSL base algorithm, whether used with ConFES or not, iterative STC and TSA (with ConFES) should be used for the best possible CDFSL results in this setting. Even when using 100 unlabelled images per episode, most datasets still exhibit improved performance over supervised learning when applying ConFES-TSA.

6 Future work

This paper focuses on showing the benefits obtained with simple centroid-based self-training. It may be possible to modify STC in certain ways to make better use of an iterative process, in order to achieve consistent performance gains over multiple iterations and ultimately better semi-supervised CDFSL performance. Another potential modification would be to weigh the unlabelled centroids less when averaging with the labelled centroids if the unlabelled set is small, which may reduce STC performance loss with very small unlabelled sets. Additionally, one may investigate whether STC can exploit information in unlabelled instances belonging to classes other than those in the labelled training set: in practice, out-of-class instances may be present in the unlabelled set as either noise or additional data.

7 Conclusion

We show semi-supervised learning algorithms for large labelled datasets may be unsuitable for CDFSL as they frequently exhibit lower estimated accuracy than purely supervised learning. We propose STC, an efficient semi-supervised learning method that is more robust against data scarcity and domain shift and is compatible with a range of state-of-the-art CDFSL methods utilising nearest-centroid classification, including URL, FLUTE, TSA, and ConFES. We evaluate STC extensively and show that it generally improves these CDFSL methods' average performance on the Meta-Dataset benchmark when applied with a moderate number of 1,000 unlabelled instances. STC requires no additional backpropagation beyond applying supervised learning, which means it is computationally efficient.

REFERENCES

Peyman Bateni, Jarred Barber, Jan-Willem van de Meent, and Frank Wood. Enhancing few-shot image classification with unlabelled examples. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022*, pp. 1597–1606. IEEE, 2022. doi: 10.1109/WACV51458.2022.00166. URL https://doi.org/10.1109/WACV51458.2022.00166.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 2020. URL http://proceedings.mlr.press/v119/chen20j.html.

Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Yunhui Guo, Noel Codella, Leonid Karlinsky, James V. Codella, John R. Smith, Kate Saenko, Tajana Rosing, and Rogério Feris. A broader study of cross-domain few-shot learning. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK*, volume 12372 of *Lecture Notes in Computer Science*, pp. 124–141. Springer, 2020.

Ashraful Islam, Chun-Fu (Richard) Chen, Rameswar Panda, Leonid Karlinsky, Rogério Feris, and Richard J. Radke. Dynamic distillation network for cross-domain few-shot recognition with unlabeled data. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 3584–3595, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/1d6408264d31d453d556c60fe7d0459e-Abstract.html.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=BJ6oOfqge.

Wei-Hong Li, Xialei Liu, and Hakan Bilen. Universal representation learning from multiple domains for few-shot classification. In *2021 IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada*, pp. 9506–9515. IEEE, 2021.

Wei-Hong Li, Xialei Liu, and Hakan Bilen. Cross-domain few-shot learning with task-specific adapters. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA*, pp. 7151–7160. IEEE, 2022.

Yue Li and Jiong Zhang. Semi-supervised meta-learning for cross-domain few-shot intent classification. In *Proceedings of the 1st Workshop on Meta Learning and Its Applications to Natural Language Processing*, pp. 67–75, 2021.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11): 2624–2637, 2013.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=HJcSzz-CZ.

James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E. Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *Advances in Neural Information Processing Systems 32, Vancouver, BC, Canada*, pp. 7957–7968, 2019.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *7th IEEE Workshop on Applications of Computer Vision / IEEE Workshop on Motion and Video Computing (WACV/MOTION 2005), 5-7 January 2005, Breckenridge, CO, USA*, pp. 29–36. IEEE Computer Society, 2005. doi: 10.1109/ACVMOT.2005.107. URL https://doi.org/10.1109/ACVMOT.2005.107.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30, Long Beach, CA, USA*, pp. 4077–4087, 2017.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1195–1204, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/68053af2923e00204c3ca7c6a3150cf7-Abstract.html.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *8th International Conference on Learning Representations, Addis Ababa, Ethiopia*. OpenReview.net, 2020.

Eleni Triantafillou, Hugo Larochelle, Richard S. Zemel, and Vincent Dumoulin. Learning a universal template for few-shot dataset generalization. In *Proceedings of the 38th International Conference on Machine Learning, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10424–10433. PMLR, 2021.

Hongyu Wang, Eibe Frank, Bernhard Pfahringer, Michael Mayo, and Geoffrey Holmes. Feature extractor stacking for cross-domain few-shot meta-learning. 2022. doi: 10.48550/ARXIV.2205.05831. URL https://arxiv.org/abs/2205.05831.

Rui Xu, Lei Xing, Shuai Shao, Lifei Zhao, Baodi Liu, Weifeng Liu, and Yicong Zhou. GCT: graph co-training for semi-supervised few-shot learning. *IEEE Trans. Circuits Syst. Video Technol.*, 32(12):8674–8687, 2022. doi: 10.1109/TCSVT.2022.3196550. URL https://doi.org/10.1109/TCSVT.2022.3196550.