SelectNAdapt: Support Set Selection for Few-Shot Domain Adaptation

Youssef Dawoud[1], Gustavo Carneiro[2], and Vasileios Belagiannis[1]

[1]Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany , {first.last}@fau.de
[2]University of Surrey, United Kingdom , g.carneiro@surrey.ac.uk

Abstract

Generalisation of deep neural networks becomes vulnerable when distribution shifts are encountered between train (source) and test (target) domain data. Few-shot domain adaptation mitigates this issue by adapting deep neural networks pre-trained on the source domain to the target domain using a randomly selected and annotated support set from the target domain. This paper argues that randomly selecting the support set can be further improved for effectively adapting the pre-trained source models to the target domain. Alternatively, we propose SelectNAdapt, an algorithm to curate the selection of the target domain samples, which are then annotated and included in the support set. In particular, for the K-shot adaptation problem, we first leverage self-supervision to learn features of the target domain data. Then, we propose a per-class clustering scheme of the learned target domain features and select K representative target samples using a distance-based scoring function. Finally, we bring our selection setup towards a practical ground by relying on pseudo-labels for clustering semantically similar target domain samples. Our experiments show promising results on three few-shot domain adaptation benchmarks for image recognition compared to related approaches and the standard random selection.

1. Introduction

Domain shifts between source and target domain data are considered harmful to the generalisation performance of deep neural networks (DNNs). The adaptation of DNNs to the target domain is, thus, essential to preserve their performance on the task in place. Among the family of adaptation methods, few-shot adaptation is a well-known approach that adapts DNNs to the target domain using a few annotated target domain samples. However, few-shot adaptation relies on the random selection of target domain samples to be annotated, which is likely a sub-optimal sample selection procedure.

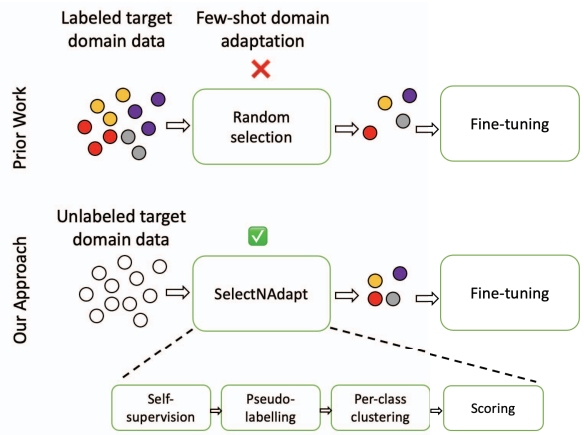

Figure 1. Few-shot domain adaptation is a powerful technique that should be exploited carefully. We propose a more effective support set selection for few-shot domain adaptation by replacing a random selection strategy by an algorithm to select representative target domain samples in an unsupervised way. Our pipeline leverages self-supervision, pseudo-labelling, clustering and selection via a distance metric score.

Adaptation of DNNs can be carried out with varying settings regarding source and target domain data availability. Vanilla (unsupervised) domain adaptation assumes adaptation of DNNs using jointly source and unlabelled target domain data [12]. On the other hand, recent studies argue that access to source domain data at test time is often impractical due to several reasons, including privacy and computation efficiency [39]. As a result, test-time domain adaptation emerged as a more interesting alternative setting that disregards source domain data at test-time and assumes only access to the pre-trained source model and the target domain. Well-known test-time adaptation strategies perform on-the-fly adaptation by updating the batch-normalisation (BN) statistics of the source models using unsupervised losses, e.g., entropy minimisation [39]. Nevertheless, it has been shown by [46] that proper adaptation of the BN statis-

tics can not be achieved without supervision from the target domain, as it can not be guaranteed that unsupervised adaptation can correct the domain shifts. Furthermore, test-time adaptation approaches require large mini-batches from the target domain for a good approximation of the BN statistics. Therefore, supervision from the target domain is necessary and can be provided in the form of a small number of randomly selected and annotated target domain samples known as the support set [8]. This process is referred to as source-free few-shot domain adaptation.

Similarly to few-shot adaptation, few-shot classification tasks assume a class-balanced support set. However, this would require access to the ground-truth of the target domain to select a set of samples per class which is also impractical in real-world situations. Indeed few-shot adaptation has brought a significant improvement compared to the state-of-the-art unsupervised test-time adaptation approaches, yet, it remains unclear whether adaptation of the source model using randomly selected samples from the target domain is sufficient for a good performance on the target domain. Optimising for data sample selection has been thoroughly studied in active learning where data samples are selected and annotated sequentially using unsupervised losses like Shannon's entropy [36] or MC-dropout [11]. Nevertheless, it has not been addressed before for few-shot adaptation, where a *support* set is selected in one step only to adapt a pre-trained source model.

In this paper, we focus on few-shot adaptation for image recognition. We empirically argue that proper adaptation of the pre-trained source model requires selecting *representative* target domain data to be included in the support set. Therefore, we propose a simple yet effective selection approach that boosts the few-shot adaptation performance by improving the selection of target samples to be included in the support set. In particular, we propose to perform per-class clustering of the target domain features where the number of clusters is equivalent to the number of K-shot adaptation task at hand. The target samples with features close to the cluster centres are included in the support set. However, using the source backbone for extracting target domain's features may negatively impact the clustering and selection process due to the domain shift between source and target. Thus, we seek to narrow this gap by training the source backbone with self-supervision from the target domain. Unlike the prior work [46], we rely next on pseudo-labels for determining the target samples of the same pseudo-class to avoid using the target domain ground-truth. Finally, we rely on the Euclidean distance as our selection score to determine the distance of target samples' features to their corresponding cluster centres. A percentage of samples with the smallest distance to the cluster centres are annotated and included in the support set i.e. we use the real labels of support set samples at the adaptation stage. Our code is made publicly available [1].

To the best of our knowledge, we are the first to propose a mechanism to select a support set for few-shot domain adaptation. In summary, our contributions are as follows:

- Our algorithm overcomes the target domain shift using state-of-the-art self-supervised tasks to learn target-specific features which aid in a robust support set selection. The learned target features are utilised to select representative samples by per-class clustering of target data features.
- We rely on pseudo-labels for the target domain data to select support set samples, in particular, K-shots per class. Hence, we ought to perform the selection in a more practical and realistic setup without the need to access the target ground-truth, unlike, prior work.
- In our experiments, including several domain adaptation benchmarks, we deliver major improvements compared to random support set selection. Additionally, we show promising results when comparing with other selection approaches, namely, entropy and MC-dropout and few-shot transfer-learning baselines.

2. Related Work

2.1. Few-Shot Domain Adaptation

Early domain adaptation approaches assumed the availability of source data for jointly adapting a pre-trained deep neural network to a new unlabelled target data [12, 31]. Under this assumption, several unsupervised-domain adaptation tasks have been developed over the course of the years, including unsupervised-domain classification and segmentation [35, 25] for natural and medical datasets [16, 13]. Recently, test-time domain adaptation restricted access to the source data and only allowed access to the pre-trained source model along with the target domain data [40, 38, 42]. Test-time adaptation approaches adapt the BN statistics of the source model using unlabelled data from the target domain. For instance, Tent and test-time BN adaptation [27] rely on unsupervised loss functions like entropy minimisation to update BN parameters using mini-batches from the target domain. Nevertheless, it has been shown in [46] that proper adaptation of the BN parameters in neural networks requires supervision from the target domain using a randomly selected support set comprising few-annotated samples. In this paper, we argue that random support selection remains ineffective for good approximation of BN statistics. Therefore, we present a method for optimising the selection of support set that further enhances the BN approximation and in return the overall few-shot adaptation performance.

[1] https://github.com/Yussef93/SelectNAdaptICCVW

Additionally, we pose the selection problem as an unsupervised selection where we rely on pseudo-labels in the per-class clustering stage. In our experiments, we demonstrate state-of-the-art results using our selection mechanism.

2.2. Self-Supervised Learning for Domain Adaptation

Self-supervision is a widely used technique for learning useful representations that enhances the performance on downstream tasks [9, 2, 45, 14]. Over the past few years, self-supervision tasks have been introduced in unsupervised-domain adaptation approaches [28] where the target domain in conjunction with the supervision from the source domain is utilised to reinforce the representations of the shared backbone network. Accordingly, several self-supervised tasks have been put to practice in unsupervised-domain adaptation and have demonstrated promising results [20, 34]. Similarly, contrastive learning has been exploited in test-time adaptation [4] jointly with pseudo-labels to learn classification on the target domain. Also, in our work, we utilise self-supervised learning. In particular, we train the backbone of the source network using self-supervision defined over the target domain data for reducing the gap between the data features of source and target domains similar to [7]. In return, the learned target features of each class are clustered based on the K-shot problem at hand. We show that relying on self-supervision delivers significantly better results than using features of the source backbone.

3. Method

In this section, we start by defining the problem of support set selection. Then, we present our unsupervised support set selection approach for adapting a pre-trained source model to the target domain.

3.1. Problem Definition

Let $h_S = g \circ f$ be a deep neural network trained on the source domain $\mathcal{D}_S$, where f denote the backbone network of the source model that maps an input image to a latent code (feature representation) $f : \mathcal{X} \rightarrow \mathcal{Z}, \mathcal{Z} \subset \mathbb{R}^D$ and g is the task head network that maps features extracted by f to the output space of the learning task at hand $g : \mathcal{Z} \rightarrow \mathcal{Y}$. In this work, we focus on image recognition, hence, g is a classification head that maps the features to the label space $\mathcal{Y} \subset [0, 1]^C$, where C is the total number of classes. Note that both source and target images share the same label space. Given access only to h_S and the target domain $\mathcal{D}_T$ at adaptation time, our objective is to seek for few annotated target samples, namely, the support set $\tilde{\mathcal{D}}_T \subset \mathcal{D}_T$, to adapt h_S to the shifted target domain $\mathcal{D}_T$. In general, a few-shot classification task is framed as C-way, K-shot task which is referred to as the support set, where C is the number of semantic classes and K is the number of samples per class, as a rule of thumb $K \leq 10$. Instead of randomly selecting the target samples, we present a support set selection algorithm that improves the adaptation performance compared to random selection. The size of $\tilde{\mathcal{D}}_T$ is set to KC.

Algorithm 1: Unsupervised Support Set Selection

Input: Source model h_S trained using $\mathcal{D}_S$, unlabelled target domain data $\mathcal{D}_T$, and annotation budget $|\tilde{\mathcal{D}}_T| = KN$.
Adapt f with self-supervision using $\mathcal{D}_T$ and keep f', q.
Get pseudo-labels of $\mathbf{X} \in \mathcal{D}_T$ (1).
for $c = 1, 2, \ldots, C$ **do**
  $\mathbf{x}^c = \{\}$
  for $m = 1, 2, \ldots, |\mathcal{D}_T|$ **do**
    if $\hat{\mathbf{y}}_m = c$ **then**
      $\mathbf{x}^c \cup \{\mathbf{x}_m\}$
    end if
  end for
  Get features $\mathbf{z}^c$ from (2).
  Cluster $\mathbf{z}^c$ using K-means algorithm i.e. calculate cluster centres $\mu^c = [\mu^c_0, \ldots, \mu^c_{K-1}]$.
  for i= 1,2,...,K **do**
    Calculate $d(\mathbf{z}^{c,i}, \mu^c_i)$ from (3).
  end for
  Select $\tilde{\mathcal{D}}_T$ from (4) using $d(\mathbf{z}^c, \mu^c)$.
end for
Update BN parameters of f' using LCCS and $\tilde{\mathcal{D}}_T$ from (5).
Output: Evaluate h_T on $\hat{\mathcal{D}}_T$.

To reach our goal, we learn features of the target domain data by training the backbone network of the source model on state-of-the-art self-supervised tasks, namely, contrastive learning task [15, 3]. Next, we get pseudo-labels of the target domain data using the features of both the source backbone and the backbone trained using self-supervision. Afterwards, we perform per-class clustering using the features learned from the self-supervised task and select the target samples according to our scoring function, which is the minimum Euclidean distance to the cluster centre. In the end, we adapt the model to the target domain using our selected support set and evaluate the adapted model on the target test set. We present our algorithm in detail below.

3.2. SelectNAdapt Algorithm

As previously stated, we assume access to the pre-trained source model h_S and the target domain $\mathcal{D}_T$ of size M containing unlabelled images denoted by $\mathbf{X}$. We summarise the steps of our approach in algorithm 1. Moreover, a visual explination is provided in Fig. 2.

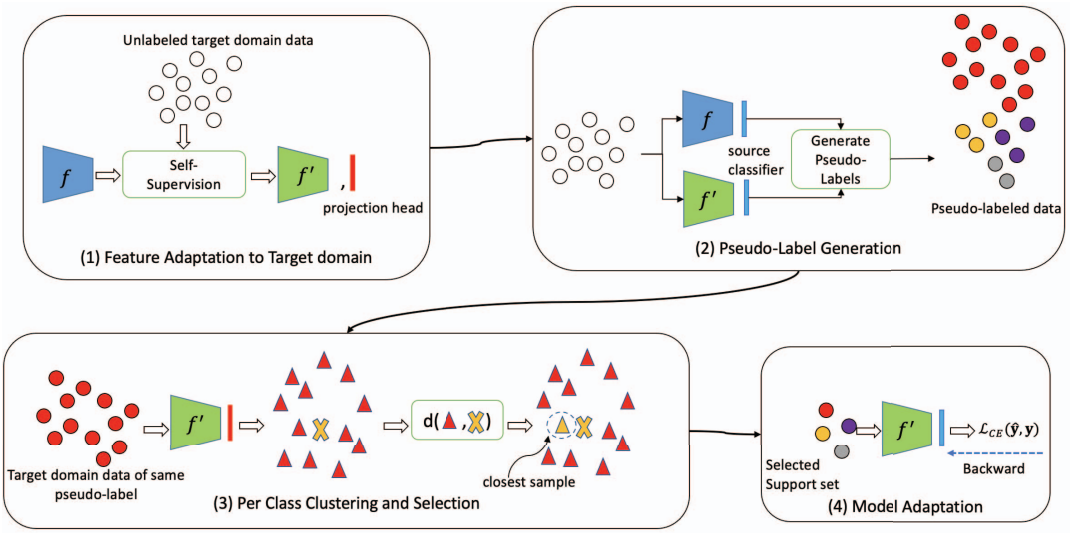

Figure 2. The complete pipeline of our *SelectNAdapt* algorithm at $K = 1$-shot. First, we utilise self-supervision for adapting the source features to the target domain data. Then, we generate pseudo-labels with the source classifier for the target domain using the features from the backbone of the pre-trained source model and the backbone trained using BYOL task. Next, we do per-class clustering and calculate the cluster centre (represented by "X") using K-means algorithm where the support set samples are selected using the Euclidean distance as the scoring metric. Finally the model is adapted using selected the support set.

3.2.1 Feature Adaptation

We argue that source-extracted features of target domain data may prohibit an effective selection of support samples in the per-class clustering step due to the shifted target domain. To alleviate this issue, we train the source backbone $f \rightarrow f'$ using self-supervision to learn target-related features, which provisions a better feature clustering compared to using the source backbone and, thus, better support set selection. Contrastive learning [5, 3, 15] have gained a wide reputation over the past years for their ability to learn useful representation. In this context, the objective of the learning task is to train the backbone of a DNN with a projection head attached to it to learn an embedding space that pulls similar data pairs together while pushing dissimilar ones apart. Afterwards, the backbone is fine-tuned on a particular downstream task. Accordingly, we use contrastive learning tasks to train the source backbone using the target domain data $\mathcal{D}_T$. After training we keep the trained backbone f' and the projection head q, which projects the features extracted by f' onto a d-dimensional feature space where $d < D$.

3.2.2 Pseudo-labels generation

A few-shot classification task is defined as C-way, K-shot i.e. the support set should contain K-shots for every class $c \in C$. To construct the support set from unlabelled target data, we first generate pseudo-labels for $\mathbf{X}$ by using the features of f and f' along with the source classifier. To obtain the pseudo-labels, we follow an ensemble prediction model [33] where we average the output probability distributions of the source classifier g using the features of f and f'. The assigned pseudo-label is based on the maximum output probability over the distribution of classes which we define as follows:

$$\hat{\mathbf{Y}} = \arg \max_{1,2,\ldots,C} \frac{\sigma(g(f(\mathbf{X}))) + \sigma(g(f'(\mathbf{X})))}{2}, \quad (1)$$

where $\hat{\mathbf{Y}} = [0,1]^{M \times C}$ is a matrix that holds one-hot encoding vectors of length C for all the target sample and σ is a softmax activation function [1]. Afterwards, we group the target samples according to their pseudo-labels into C categories, i.e. $\mathbf{X} = [\mathbf{x}^0, \ldots, \mathbf{x}^{C-1}]$, with $\mathbf{x}^c = [\mathbf{x}^c_1, \ldots, \mathbf{x}^c_{|\mathbf{x}^c|}], c \in \hat{\mathbf{Y}}$ and extract their features $\mathbf{z}^c$ using f':

$$\mathbf{z}^c = q(f'(\mathbf{x}^c)), \quad \forall \mathbf{x}^c \in \mathbf{X}, \quad (2)$$

where $\mathbf{x}^c$ is the set of all target domain samples with pseudo-label c, $\mathbf{z}^c$ is a matrix containing d-dimensional feature vectors of $\mathbf{x}^c$, and q is the projection network retained from the self-supervised learning task. It is noteworthy that adding a projection head in contrastive learning frameworks empirically performs better than relying on the raw features of the backbone network [5]. Therefore, we leverage the output features of the projection network in the clustering step.

3.2.3 Per-Class Clustering and Selection

We rely on a scoring function to rank and select target data samples per class. To this end, we cluster $\mathbf{z}^c$ into K-clusters. Note that K is the number of shots per class c. We calculate the cluster centres $\mu^c = [\mu_1^c, \ldots, \mu_K^c]$ based on the K-means algorithm [29] and score the target samples according to the distance of their features to the cluster centres. We make use of the Euclidean distance as our scoring function that measures the distance between the features assigned to cluster i and their cluster centre μ_i^c, where $i \in [1, \ldots, K]$, we define the distance metric:

$$d(\mathbf{z}^{c,i}, \mu_i^c) = ||\mathbf{z}^{c,i} - \mu_i^c||_2. \tag{3}$$

The features of the target samples with the minimum distance to their corresponding cluster centre are included in the support set $\tilde{\mathcal{D}}_{\mathcal{T}}$, hence, our selection metric becomes:

$$\tilde{\mathcal{D}}_T = \arg\min_{\mathcal{D}_T} \sum_{c=0}^{C-1} \sum_{i=0}^{K-1} \sum_{z^c \in \mu_i^c} d(\mathbf{z}^{c,i}, \mu_i^c) \quad s.t. |\tilde{\mathcal{D}}_T| = KN. \tag{4}$$

Next, the selected samples to be included in the support set are associated with their true labels i.e. we do not use their pseudo-labels at adaptation time, hence, $\tilde{\mathcal{D}}_T = \{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{KN}$, where $\mathbf{y}_j$ is the true label of $\mathbf{x}_j$.

3.2.4 Model adaptation

Eventually, we use $\tilde{\mathcal{D}}_T$ to update the BN parameters of the pre-trained source model where its backbone network is replaced with f'. We follow the approach of linear combination coefficients for batch normalisation statistics (LCCS) [46] to update the BN parameters using the cross-entropy loss [1] $\mathcal{L}_{CE}(g, f', \tilde{\mathcal{D}}_T)$, hence the update BN parameters are optimised as:

$$\theta^* = \arg\min_{\theta_{f'}} \mathcal{L}_{CE}(g, \theta_{f'}, \tilde{\mathcal{D}}_T), \tag{5}$$

where $\theta_{f'}$ denote the BN parameters of the backbone trained on BYOL and θ^* are the updated BN parameters. Finally, we evaluate the updated model (h_T) on the target test set $\hat{\mathcal{D}}_T = \mathcal{D}_T \setminus \tilde{\mathcal{D}}_T$.

4. Experiments

Dataset	PACS	VisDA	Office-31
Backbone	ResNet-18	ResNet-101	ResNet-50

Table 1. Backbone architectures used in our experiments.

4.1. Datasets

We conduct our experiments using three domain adaptation benchmarks for image recognition. Namely, we use PACS (4 domains with 7 classes) [22], Office-31 (3 domains with 31 classes) [32], and VisDA datasets (2 domains with 12 classes) [30]. Each of these datasets incurs domain shifts in the form of different image styles (PACS and VisDA) or images captured with different cameras (Office-31). We adopt the same evaluation metrics used in [46], in particular, we adopt accuracy as an evaluation metric for PACS, average-per-class accuracy for Office-31, and average precision for VisDA. The evaluation protocol for PACS and VisDA follows a leave-one-domain-out cross validation [10] where one domain is left out as the target domain and the rest is treated as the source domain(s). On the other hand, the evaluation protocol for Office-31 splits the dataset into 6 pairs, each pair contains one source domain and one target domain.

4.2. Implementation Details

Source Models We use different backbone networks for each dataset as shown in Tab.1. Our source models are trained on the source domain(s) using empirical risk minimisation (ERM) [17]. However, we use the publicly available pre-trained model CSG ResNet-101 [6] on the source domain of VisDA. As for PACS and Office-31 we reproduce the training on the source domains following the implementations of [47, 48].

Feature Adaptation As previously stated, we rely on contrastive learning, namely BYOL, a regressive self-supervised task. BYOL is a well-known self-supervised contrastive learning framework that does not require negative samples and is less sensitive to hyper-parameter changes. In BYOL, two networks with identical backbones, namely, online and target networks interact and learn from each other. In particular, the online network learns to regress the features of the target network under different augmentations of the same image. Hence, it enforces consistent representations. We initialise the backbones of the online and target networks with the parameters of f and learn the BYOL task. For PACS dataset, we train for 100 epochs using a LARS optimiser [44] with initial learning rate of 0.2 and cosine annealing scheduler [26]. As for the remaining datasets, we use an Adam optimiser [21] with learning rate of 0.0001. Moreover, we train for 100 epochs for the target domains of Office-31. However, for VisDA we empirically observed that 10 epochs are sufficient for the training to converge on the BYOL task. For all target domains, we use a mini-batch size of 256. Upon completing the learning task we keep the backbone and the projection head of the online network and use them for extracting

Support Set Selection Method	Office-31						
	A → W	A →D	W →A	W →D	D→ W	D →A	Average
Random [46]	92.8	91.8	75.1	99.9	98.5	75.4	88.9
Entropy	88.4	87.7	72.3	97.7	97.3	71.9	85.9
MC-dropout	87.5	85.8	73.8	100.0	98.2	71.9	85.9
Ours	95.0	97.4	76.4	100.0	99.7	75.9	**90.7**

Table 2. A comparison of adaptation test results using random, entropy, MC-dropout and *SelectNAdapt* algorithm (***Ours***) for Office-31 dataset at K = 5-shot.

Support Set Selection Methods	PACS			VisDA		
	1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
Random [46]	81.6	86.1	87.6	67.8	76.0	**79.2**
Entropy	79.6	86.2	86.1	61.0	73.8	74.2
MC-dropout	80.3	86.1	86.8	68.1	73.1	74.1
Ours	**84.5**	**87.9**	**87.9**	**73.0**	**78.0**	78.0

Table 3. A comparison of averaged few-shot adaptation test results using different selection approaches, namely, random, entropy, MC-dropout and *SelectNAdapt* algorithm (***Ours***) for PACS and VisDA datasets. Note that we average adaptation results over the target domains of PACS.

features of target domain data.

Support Set Selection We compare our selection against random, entropy, and MC-dropout selection approaches. For random selection, following [46], the target domain data are chosen according to a uniform probability distribution. Note that the support set in [46] is class-balanced i.e. there is an equal number of support samples for every class in the target domain. As for entropy, we calculate Shannon's entropy using the softmax output of the source model [36]. For each class in the target data, we select the top-K samples with the highest entropy loss. Similarly, for MC-dropout we average Shannon's entropy over 10 forward passes using h_S with a dropout layer of probability 0.5 inserted at the final layer of the backbone [11]. This setting has been shown to yield the best result. We also compare our selection approach to few-shot transfer learning baselines, which adapts the pre-trained source model in different ways.

Model adaptation The BN parameters are adapted based on LCCS method of [46] using an Adam optimiser for 10 epochs with 0.001 learning rate with mini-batch size of 32. During adaptation on datasets PACS and VisDA we use a nearest-centroid classifier [43] for $K \geq 5$ [37], otherwise, we use the pre-trained source classifier. However, for Office-31, the source classifier is fine-tuned after adapting the BN parameters for 200 epochs using the same Adam optimiser settings for adapting the BN parameters. We average the test results over at least 3 different seeds.

4.3. Support Set Selection Comparison

We present our averaged numerical test results for random, entropy, MC-dropout and our selection approach in Tab. 3 and 2. For PACS datasets, we also average the test results over the target domains for better comparison of different selection approaches. Results per domain could be viewed in the supplementary material. Clearly, our approach mainly dominates all other selection approaches on all the benchmarks which supports our claim that careful selection of support set samples from the target is vital for an effective adaptation performance. Although our approach may result in a class-imbalanced support set due to false pseudo-labels that do not align with the real labels of target samples, we notice that adaptation performance remains robust and still performs better compared to the random selection, which is yet an additional reason that highlights the importance of representative support set samples over a randomly selected and class-balanced support set. Furthermore, we observed in several cases that random selection could have a better performance than entropy and MC-dropout approaches. We attribute this behaviour to the tendency of entropy to select samples that lie close to the decision boundaries of per-class clusters, which result in high prediction uncertainty. These samples are less beneficial for the adaptation performance as they are biased towards a specific region i.e. the boundaries of the decision space. Hence, they are considered poor representative candidates for the adaptation task of BN parameters. On the other hand, our approach that learns target features and then performs per-class clustering to select target domain samples that fall near the cluster centres, i.e. samples that represent each cluster, positively impacts the adaptation performance.

4.4. Few-Shot Learning Comparison

In Tab. 4, we report the results of few-shot transfer learning approaches tailored to fit the setting of few-shot adaptation, which neglects the presence of source domain data

Few-Shot Adaptation Methods	PACS			VisDA		
	1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
Ada-BN [24]	82.9	85.5	85.8	56.5	60.9	61.8
fine-tune BN [24]	79.0	84.3	85.4	59.1	70.9	74.9
fine-tune classifier [24]	82.5	83.7	83.8	67.6	69.7	77.4
fine-tune feat. extractor [24]	83.6	86.0	86.1	67.3	68.4	74.7
L^2 [41]	84.4	85.8	85.6	66.0	66.4	69.6
L^2-SP [41]	84.4	85.8	85.6	66.0	66.4	69.6
DELTA [23]	84.4	85.8	85.6	65.9	66.5	70.1
Late Fusion [19]	83.2	83.6	83.6	67.2	69.8	74.5
FLUTE [37]	73.4	85.8	88.1	48.3	67.1	65.7
LCCS [46]	84.4	87.1	88.8	67.8	76.0	**79.2**
Ours	**88.2**	**89.3**	**89.5**	**73.0**	**78.0**	78.0

Table 4. We report test results averaged over PACS target domains, as well as the test results of VisDA for K = 1-, 5-, 10-shots. In this table, we compare against different few-shot transfer learning approaches tailored to the setting of source-free few-shot domain adaptation. Note that the pre-trained source model f has been trained with MixStyle domain generalisation approach [48] on PACS dataset.

at adaptation time and adapts the pre-trained source model using a randomly selected support set. These approaches include AdaBN [24] for replacing BN statistics using a randomly selected support set from the target domain. Later on either the BN parameters of the source model or the backbone network or classifier layer are fine-tuned. Moreover, other approaches like L^2, L^2-SP [41], DELTA [23] that finetunes the entire source model using an additional regularisation term are shown. Finally, we have FLUTE [37] that adapts BN parameters using nearest-centroid classifier and Late Fusion [19] which averages classification results using source and target classifier. We employ the source model trained using the MixStyle approach [48] for a fair comparison with the random selection baseline, . We observe that our selection method can still show quite significant results compared to the random selection and the few-shot learning baselines, which highlights the significance of proper support set selection compared to using different adaptation techniques. In the following section, we conduct more ablation studies using the PACS dataset to analyse various components of our approach.

4.5. Ablation Study

Class-balanced support set and model adaptation As previously mentioned, the random selection baseline [46] ensures a class-balanced support set following the few-shot classification protocol of C-way, K-shot. To achieve this, prior knowledge of target domain data ground-truth is essential to select K-shots randomly from the set of target samples belonging to a particular class $c \in C$. Accordingly, we carry out an experiment assuming a prior knowledge of target domain ground-truth. In this case, we skip the pseudo-labelling step and use directly the self-supervised model f' to extract target data features and perform per-class clustering to find representative target samples to be included in the support set. The averaged results on the PACS datasets are shown in Tab. 5 where it is noticeable that a class-balanced support set widens the performance gap between our approach and random selection. Hence, we deduce from this experiment that applying our selection for finding representative target samples is more efficient compared to random selection even in the presence of ground-truth data. Additionally, we analyse the impact of our selection approach against random selection using f' as the backbone to be adapted. We notice the overall performance remains better using our selection approach, especially at K = 1-shot, implying a more efficient selection mechanism compared to random selection. On the other hand, the performance of random selection also increases relative to using the pre-trained source backbone since the self-supervised pre-trained backbone yields a better initialisation of network parameters due to the learned representation on the target domain. Finally, we observe the performance gap between the random selection and our approach is reduced as K increases due to the availability of sufficient training data.

Source model for pseudo-label generation, per-class clustering and selection In Tab.6, we document the results of neglecting the self-supervision step. Specifically, we use the source backbone for generating pseudo-labels, per-class clustering, and selecting support set samples from target domain data, and adapting it using the selected support set. From the results, we notice a significant difference between using the backbone of the source model alone and the backbone trained on the BYOL task. This clearly indicates the impact of self-supervision in the selection process as it bridges the domain shift gap between source and target domains by learning useful target features for a robust selection of the support set. Furthermore, in Tab. 8, we

Selection Method	Adapted Model	Bal	PACS		
			1-shot	5-shot	10-shot
Random [46]	f	✓	81.6	86.1	87.6
	f'	✓	84.3	85.9	87.9
Ours	f'	x	84.5	87.9	87.9
	f'	✓	85.6	88.4	88.6

Table 5. A comparison of few-shot adaptation test results for random selection and our approach under different settings of adapting BYOL trained backbone, i.e., f' and the source backbone f, in addition to, class-balanced (Bal) support sets.

SelectNAdapt		PACS		
f	f'	1-shot	5-shot	10-shot
✓	x	79.6	85.5	87.6
✓	✓	84.5	87.9	87.9

Table 6. We compare the few-shot adaptation results of using only the backbone of the pre-trained source model (f) for pseudo-label generation, per-class clustering, and model adaptation against using additionally self-supervision i.e. the backbone trained using BYOL (f').

Self-Supervision	PACS		
	1-shot	5-shot	10-shot
BYOL [15]	84.5	87.9	87.9
SwAV [3]	86.1	87.7	88.2

Table 7. A comparison of averaged few-shot adaptation test results for training the source backbone with BYOL and SwAV in our support set selection pipeline on the target domains of PACS dataset.

Pseudo-labelling		PACS		
f	f'	1-shot	5-shot	10-shot
✓	x	84.6	87.5	87.7
x	✓	84.3	87.4	87.9
✓	✓	84.5	87.9	87.9

Table 8. We show a comparison of averaged few-shot adaptation test results for using different combinations of source f and self-supervised (BYOL) backbones f' to generate pseudo-labels.

study the effect of using different combinations of source and self-supervised (BYOL) backbones i.e. f and f', to generate pseudo-labels. Combining the former with the latter to form an ensemble prediction has a slightly better performance across all K-shot adaptation cases compared to using each backbone individually. Ensemble models are well-known to yield more accurate predictions than individual model predictions [18] since individual models may be prone to bias/variance errors.

Performance with SwAV self-supervised task We conduct an experiment to analyse the performance of our support set selection, however, using a different contrastive learning self-supervised task, namely, SwAV [3]. SwAV is a classification self-supervised task that enforces consistent cluster assignment prediction for each data sample under different augmentations. This experiment aims to demonstrate that our support set selection is agnostic to the selected self-supervision objective. To this end, we conduct an experiment using PACS dataset, where we train the source backbone f using the target domain data on the task of SwAV following the implementation of [3]. Like BYOL, we use the backbone model f' and the projector network q from SwAV in the remaining steps of the selection pipeline. Tab. 7 shows that SwAV yields even improved performance in the few-shot adaptation. The results show improvement at K=1-shot compared to BYOL and on-par performance at 5 and 10-shots. These results imply that our selection mechanism does solely depend on BYOL and can still function well using other self-supervision methods.

4.6. Discussion

Our *SelectNAdapt* algorithm yields effective few-shot adaptation results compared to other selection baseline in the context of image recognition. However, as a part of the future work, support set selection for few-shot adaptation tasks such as image segmentation could be a investigated.

5. Conclusion

We presented a support set selection approach from the target domain data for few-shot domain adaptation. Our approach by leveraging self-supervision, pseudo-labelling, per-class clustering and the Euclidean distance as a scoring metric has effectively boosted the adaptation performance and dominated random selection as well as loss-based selection approaches, namely, entropy and MC-dropout. Furthermore, our selection approach avoids the need to access ground-truth of target data making it more practical compared to prior work. We have also compared to few-shot transfer learning baselines where again, our selection has demonstrated that proper selection of support samples is sufficient to improve the adaptation performance. We observed on three image recognition benchmarks that careful selection of the support set from the target domain data significantly impacts on few-shot domain adaptation.

Acknowledgment

G.C. was supported by Australian Research Council through grant FT190100525.

References

[1] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[2] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[4] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305, 2022.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[6] Wuyang Chen, Zhiding Yu, SD Mello, Sifei Liu, Jose M Alvarez, Zhangyang Wang, and Anima Anandkumar. Contrastive syn-to-real generalization. 2021.

[7] Youssef Dawoud, Arij Bouazizi, Katharina Ernst, Gustavo Carneiro, and Vasileios Belagiannis. Knowing what to label for few shot microscopy image cell segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3568–3577, 2023.

[8] Youssef Dawoud, Julia Hornauer, Gustavo Carneiro, and Vasileios Belagiannis. Few-shot microscopy image cell segmentation. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part V*, pages 139–154. Springer, 2021.

[9] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE international conference on computer vision*, pages 567–575, 2015.

[10] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32, 2019.

[11] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[13] Amir Gholami, Shashank Subramanian, Varun Shenoy, Naveen Himthani, Xiangyu Yue, Sicheng Zhao, Peter Jin, George Biros, and Kurt Keutzer. A novel domain adaptation framework for medical image segmentation. In *International MICCAI Brainlesion Workshop*, pages 289–298. Springer, 2018.

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[16] Hao Guan and Mingxia Liu. Domain adaptation for medical image analysis: a survey. *IEEE Transactions on Biomedical Engineering*, 69(3):1173–1185, 2021.

[17] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[19] Judy Hoffman, Eric Tzeng, Jeff Donahue, Yangqing Jia, Kate Saenko, and Trevor Darrell. One-shot adaptation of supervised deep convolutional models. 2013.

[20] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[23] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, Zeyu Chen, and Jun Huan. Delta: Deep learning transfer using feature map with attention for convolutional networks. *arXiv preprint arXiv:1901.09229*, 2019.

[24] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

[25] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017.

[26] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[27] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

[28] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[29] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[30] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017.

[31] Pedro O Pinheiro. Unsupervised domain adaptation with similarity learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8004–8013, 2018.

[32] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[33] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

[34] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *Advances in neural information processing systems*, 33:16282–16292, 2020.

[35] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[36] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

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

[38] Dequan Wang, Shaoteng Liu, Sayna Ebrahimi, Evan Shelhamer, and Trevor Darrell. On-target adaptation. *arXiv preprint arXiv:2109.01087*, 2021.

[39] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[40] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022.

[41] LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pages 2825–2834. PMLR, 2018.

[42] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8978–8987, October 2021.

[43] Donghyun Yoo, Haoqi Fan, Vishnu Boddeti, and Kris Kitani. Efficient k-shot learning with regularized deep networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[44] Yang You, Igor Gitman, and Boris Ginsburg. Scaling sgd batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888*, 6(12):6, 2017.

[45] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2547–2555, 2019.

[46] Wenyu Zhang, Li Shen, Wanyue Zhang, and Chuan-Sheng Foo. Few-shot adaptation of pre-trained networks for domain shift. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1665–1671. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

[47] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[48] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *IEEE Transactions on Image Processing*, 30:8008–8018, 2021.