Chapter 4

Cleaning Noisy Labels by Negative Ensemble Learning for Source-Free Unsupervised Domain Adaptation

4.1 Introduction

Deep Convolutional Neural Networks (CNNs) have shown remarkable achievements in a variety of tasks [107]. However, to perform well, training and testing data are assumed to be drawn from the same distribution. This is unrealistic when the system needs to be deployed in real-world scenarios. Consequently, a model trained on some source domain often fails to generalize well on a related but different target domain, due to the well-known problem called *domain shift* [108, 5]. Since annotating data from every possible domain is expensive and sometimes even impossible, Unsupervised Domain Adaptation (UDA) methods seek to address such a problem by minimizing discrepancy across the domains or trying to learn domain-invariant feature embeddings, without accessing target label information.

Several research efforts have been devoted to developing UDA methods by either enforcing class-level feature distribution alignment [109, 110], matching moments [111, 112], applying domain-specific batch normalization [113], or adopting domain adversarial learning [114, 115]. However, these methods require joint access to both (labeled) source and (unlabeled) target data during training, making them unsuitable for scenarios where source data is inaccessible during the adaptation stage, or when source and target data are not available at the same time. Further, such solutions are also not viable when target data is provided incrementally at different times or if the source/target datasets are very large. Moreover, most UDA methods either focus on the single-source or single-target scenario with specific framework, regardless of the fact that data may belong to multiple source or target distributions, *e.g.,* images taken in different environments or obtained from the web (*e.g.,* sketches, photos).

To get rid of such restrictive assumptions, we propose to cast UDA as a pseudo-label refinement problem in a source data-free scenario. Consequently, unlike most of the former works, our method does not require any (target-style) data to be generated and can cope

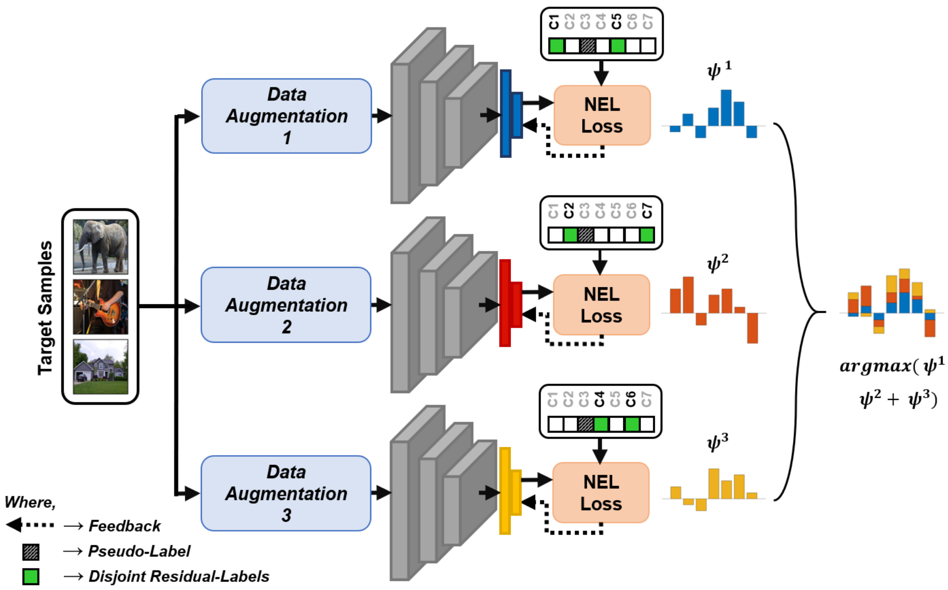


Figure 4.1: Illustration of the proposed method. In each iteration, a batch of target samples with different augmentation is fed to each ensemble member. Next, considering the inferred pseudo-label, different feedback is backpropagated by leveraging *Disjoint Residual Labels* with *Negative Ensemble Learning* (NEL) loss. This allows each member to learn diverse characteristics from data, possibly complementary, leading to a superior noise resilience and a stronger consensus leaning towards the actual class label.

with single-source, multi-source, and multi-target UDA indifferently, making it easy to use and generalizable to the dataset of diverse complexity and challenges. Our approach only assumes the availability of a model pre-trained on the source domain to infer *pseudo-labels* of unlabeled target samples. Obviously, this results in a significant amount of incorrect acquired pseudo-labels, which is a consequence of the *domain shift* [108, 5]. Hence, it appears natural to adapt to the target domain by cleaning the pseudo-labels via a (re-)assignment process, so that a new model can be trained from scratch (or a pre-trained model can be fine-tuned) using the cleaned target labels.

To clean noisy pseudo-labels, we propose Negative Ensemble Learning (NEL) technique, a unified method for adaptive noise filtering and pseudo-label refinement (see Fig. 4.1). Our method takes advantage of several expert ensemble members, each trained using a batch of target samples with different stochastic data augmentation and a novel concept of *disjoint feedback*. The design of disjoint feedback requires two essential components (i) more than one trainable models, and (ii) *different labels* for each member and a supporting *loss function*. The latter is achieved by employing an indirect learning scheme, *i.e.,* instead of using the inferred pseudo-label (out of total C classes) corresponding to target sample x, the remaining $(C-1)$ residual labels (RL) are equally distributed over all ensemble members. Next, separately for each member, the proposed NEL loss function attempts to minimize the confidence of corresponding disjoint residual labels (DRL) as in "x does not belong to either of them". Consequently, the collective form of feedback pushes confidence of inferred pseudo-label to rise.

The intuition behind is that, in case of incorrect pseudo-label, at least $N_e - 1$ members out of N_e (the total number of expert members in the ensemble network) should receive the correct information: stochastically sampling disjoint subsets of residual labels forces each ensemble member to learn different concepts, which is known to be beneficial in ensemble learning to reach a strong, hence more robust, consensus. Thus, such a consensus contributes to reaching higher confidence on clean pseudo-labels, allowing us to introduce a novel *fully adaptive* noise filtering technique to refine labels of the samples with low confidence (*i.e.,* noisy pseudo-labels) via reassignment. Finally, a standard supervised learning procedure is used to train a *single* model on the target domain data using the refined pseudo-labels featuring high confidence only.

The proposed pipeline obtains a significant noise reduction from the inferred pseudo-labels. With extensive experiments on various benchmarks yielding a wide range of shift-noise (PACS: 4% - DomainNet: 82.4%), we show that training on the target domain with refined pseudo-labels outperforms state-of-the-art UDA methods by a considerable margin. To summarise, the contributions of our work can be stated as follows:

- We propose a new, fully-adaptive method that dynamically filters out label noise and assigns cleaner pseudo-labels to noisy target samples. To do so, we introduce Negative Ensemble Learning, a new strategy that enhances diversity among members by different data augmentation and disjoint feedback, leading to improved noise resilience and a stronger consensus.

- Our method can naturally cope with the absence of source data during adaptation. It does not require new (target-style) data to be generated, avoiding the use of GAN-based models that are often difficult to train with stability. Also, it can deal with single/multi-source and multi-target UDA scenarios indifferently.

- We validate our method through detailed ablation analyses and extensive experiments on four well-known benchmarks, demonstrating its superiority over state-of-the-art UDA methods with a significant margin, *e.g.,* up to 21.8$\%$ better accuracy for PACS benchmark.

4.2 Related Work

Unsupervised Domain Adaptation. Most of the existing UDA methods focused on cross-domain feature *alignment* either by employing discriminative class-conditional alignment [110], features and prototype alignment using reliable samples [109], or customized CNN models with domain alignment layers and feature whitening [116]. Other works proposed feature distribution *matching* by approximating joint distributions [117], matching graph [118], or matching moments [111]. However, such methods assume the co-existence of source and target data during training, making them unsuitable for more realistic scenarios where source data is inaccessible, *e.g.,* due to data-privacy issues.

Source-free UDA. A few recent works showed interest in source-free UDA. For example, [119] proposed a feature corruption and marginalization technique using few labeled source samples and [120] adapted the outputs from an off-the-shelf model to minimize distribution shift using some labeled target samples. An instance-level weighting method using negative classes is proposed in [121], which is highly dependent on a procurement stage requiring source data. Another approach leveraged a pre-trained source model to update the target model progressively by generating target-style samples through conditional generative adversarial networks [5], also combined with clustering-based regularization [122]. Similarly, to improve UDA performance in person re-identification task, [123] proposes a pseudo-label cleaning process with on-line refined soft pseudo-labels.

Our proposed approach lies in this category and takes partial inspiration from the methods developed for source-free UDA in [122, 5]. Here, a pre-trained source model is used to infer pseudo-labels of target data, and then a target-style sample generator is employed for adaptation. Like in many previous works, while we start by inferring pseudo-labels using a pre-trained model but, subsequently, we progressively refine such labels exploiting the consensus of an ensemble network, without generating any (target-style), thus data avoiding the use of GAN-based models that require careful hyperparameter balance to reach stability.

Ensemble Learning. Such methods exploit features extracted from multiple models through a diversity of data projections and bring forward the mutual consensus to achieve better performances than those obtained by any individual model [124]. A comprehensive review about ensemble methods is well illustrated in [125]. The importance of learning diverse contributions from data for classifier selection and parameters update is proposed in many works, for instance [126]. Also, multiple choice learning is employed in [127] to improve the accuracy of an ensemble of models.

We also drew inspiration from the general idea proposed in these works, which agree in stressing that *diversity* among members is beneficial for ensemble robustness. We differentiate from them by introducing a new way of inducing diversity in the members, *i.e.,* we back-propagate different feedback to each member by leveraging the novel concept of Disjoint Residual Labels. This allows each member to learn diverse characteristics from data, possibly complementary, leading to a superior noise resilience and a stronger consensus leaning towards the actual class label.

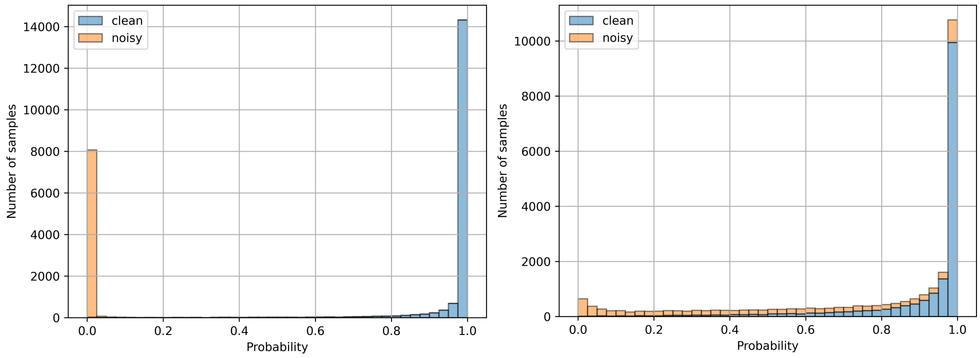
Learning with Noisy Labels. Deep CNNs are capable of memorizing the entire data even when labels are noisy [3]. To overcome such overfitting, existing methods try to select a subset of possibly clean labels for training, *e.g.,* using two networks under a co-teaching framework [128], adopting meta-learning for exemplar weight estimation [129], applying an one-out filtering approach based on the local and global consistency [130], or investigating Negative Learning (NL) as an indirect learning method [3].

In these works, the type of label noise is an important factor to be considered. The above methods typically only consider random noise from selective or uniform distribution, which has a completely different structure from the label noise injected by the domain shift affecting the inferred pseudo-labels (see Sec. 4.3). Thus, [3] fails when the noise is not uniform, and the performance results actually affected by threshold sensitivity, which limits the generalization capability of the method across benchmarks. In contrast to a fixed threshold, our Negative Ensemble Learning method features a fully *adaptive* procedure to progressively filter out the structured noise affecting target pseudo-labels.

4.3 Method

In the context of UDA for a C-class classification task, we use a model pre-trained on source data to infer pseudo-labels of the entire target set $\mathcal{D}_t$ — such set of labels will be noisy due to domain shift [108, 5]. The standard training procedure *i.e.,* training with cross-entropy loss tries to *maximize* the probability of $\boldsymbol{x}$ belonging to the corresponding inferred pseudo-label $\tilde{y}$. But, in case of noisy pseudo-label $\tilde{y} \neq y_t$ (where y_t is the inaccessible actual target label), the model would undeniably be provided the wrong information which results in poor performance.

Instead, Negative Learning (NL) [3], can reduce such probability to $\frac{1}{(C-1)}$. NL refers in fact to an indirect learning method, which instead of using a given label — $\tilde{y}$ in our case — attempts to train the classifier using a complementary label $\bar{y}$ (randomly selected from $\{1, ..., C\} \backslash \{\tilde{y}\}$) as in "data sample $\boldsymbol{x}$ does not belong to $\bar{y}$". Since the chances of selecting a true label as a complementary label are low, NL decreases the risk of providing incorrect information.



(a) Symmetric-noise (b) Shift-noise

Figure 4.2: Histogram showing the noise-filtering performance of [3] on MNIST. In both cases, the amount of noise equals 32.9% (cf. SVHN→MNIST shift-noise in Tab. 4.1).

Nevertheless, the existing NL method with a single network can not tackle *shift-noise* associated with inferred pseudo-labels. To understand better, let our CNN architecture be composed of a feature extractor $\nu_{\boldsymbol{\phi}}(.)$, a classifier $\psi_{\boldsymbol{\theta}}(.)$, and a *softmax* $\sigma(.)$, being $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ the related network parameters. The function $f : \mathcal{X} \to \mathbb{R}^C$, defined as $f(\boldsymbol{x}) = \sigma(\psi_{\boldsymbol{\phi}}(\nu_{\boldsymbol{\theta}}(\boldsymbol{x})))$[1], maps the input $\boldsymbol{x} \in \mathcal{X}$ to the C-dimensional vector of probabilities $\boldsymbol{p} \in \mathbb{R}^C$. Within a standard training procedure, namely Positive Learning (PL), the cross entropy loss function can be defined as:

$$\mathcal{L}_{PL}(\mathcal{D}_t) = -\mathbb{E}_{x_t \sim \mathcal{D}_t} \sum_{c=1}^{C} \mathbb{1}_{[c=\tilde{y}]} \log(\boldsymbol{p}) \tag{4.1}$$

where $\mathbb{1}$ is an indicator function, $\mathcal{D}_t$ represents the unlabeled target domain and $\boldsymbol{p} = f(\boldsymbol{x}_t)$. Clearly, Eq. (4.1) pushes the probability p for the given pseudo-label $\tilde{y}$ towards $\boldsymbol{p}_{\tilde{y}} = 1$. On the contrary, NL aims at encouraging the probabilities of complementary labels $\bar{y}$ to move away from 1, actually pushing them towards $\boldsymbol{p}_{\bar{y}} = 0$. The NL loss function would be so defined as:

$$\mathcal{L}_{NL}(\mathcal{D}_t) = -\mathbb{E}_{x_t \sim \mathcal{D}_t} \sum_{c=1}^{C} \mathbb{1}_{[c=\bar{y}]} \log(1 - \boldsymbol{p}) \tag{4.2}$$

While Eq. (4.2) optimizes the output probability of the complementary label to be close to zero, the probability values of other classes are increased. In such a contest, the samples carrying clean $\tilde{y}$ get higher confidence, whereas noisy ones struggle with scarcely confident scores, meeting the purpose of NL.

Nevertheless, the fundamental limitation of naive NL [3] is that it is suitable for the noise showing a uniform distribution (*symmetric-noise*) only. As reported in Fig. 4.2a, data samples

[1] Networks' parameters θ and ϕ will be omitted for brevity from now on.

with labels are assigned low confidence, resulting in effective noise separation.

Yet, a significant amount of noise is overfitted with high confidence by the network when we consider *shift-noise* — namely noise turned up during inferring pseudo-labels on the target using a model trained with the source (MNIST and SVHN, respectively, in the example of Fig. 4.2b). Note that overfitting is not to be ascribed to the *amount* of noise, which was carefully balanced in the experiment, but rather to its *distribution*. So, NL works nicely when data is affected by symmetric noise but, when complying with shift-noise as in our case, important modifications are required to be able to clean pseudo-labels effectively.

To mitigate such limitations, we propose to employ Ensemble Learning which refers to concurrently training multiple networks of similar configuration. The idea is to create a set of experts trained in different ways, in order to produce predictions with low bias and high variance. Generally, an ensemble produces the final output as a weighted sum of all the experts' logits, *i.e.,* for a data sample $\boldsymbol{x}$, the final prediction can be obtained as:

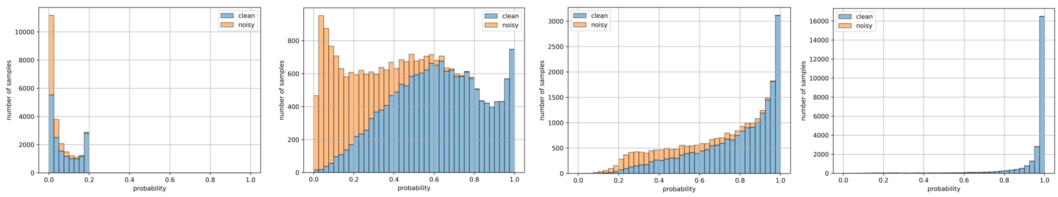
$$\boldsymbol{p}_e = \sigma(\sum_{k=1}^{N_e} \beta^k \psi^k(\nu^k(\boldsymbol{x}))) \tag{4.3}$$

where N_e corresponds to the number of experts in the ensemble network, and β^k is a set of weights modulating the contribution of each expert member.

Precisely, we set $\beta^k = 1, \forall k \in [1, N_e]$ and propose Negative Ensemble Learning loss for learning with noisy labels that, amplifies the diversity of the ensemble members by different stochastic (i) input augmentation, and (ii) feedback using Disjoint Residual Labels. The resulting strong consensus gives rise to the cleaner pseudo-labels, better than those obtained by any stand-alone network. The following sections discuss specific details of the proposed approach.

4.3.1 Adaptive Pseudo-Label Refinement

Problem Setup. The goal of UDA methods is to adapt a model pre-trained on a labelled source domain $\mathcal{D}_s = \{(\boldsymbol{x}_s^i, y_s^i)\}_{i=1}^{N_s}$ in order to generalize well on a different, yet related, unlabeled target domain $\mathcal{D}_t = \{\boldsymbol{x}_t^j\}_{j=1}^{N_t}$, where N_s and N_t denote the number of samples in the source and the target domain, respectively, and the label set $\mathcal{Y}$ is the same for the 2 domains, *i.e.,* $\mathcal{Y}_s = \mathcal{Y}_t$. For the sake of generality, we assume to work with $M_d + 1$ domains: M_d source domains $\mathcal{D}_s$, where $s = \{1, ..., M_d\}$, and a target domain $\mathcal{D}_t$. Differently from many standard UDA methods, our proposed approach does not use any source data for adaptation, nor generate target-style data at any stage. Instead, we simply use only a pre-trained source



(a) epochs = 1 (b) epochs = 10 (c) epochs = 25 (d) epochs = 50

Figure 4.3: Prediction-confidence trend during training and pseudo-label refinement by our proposed NEL method in case of SVHN→MNIST source-free UDA. Almost all samples are predicted with very low confidence at the beginning (a). As the network starts learning, noisy samples are segregated in a low confidence interval. Only if confidence is lower than γ (eq. 4.6), pseudo-labels are reassigned. Noise is thus progressively reduced (c-d).

model to infer pseudo-labels $\mathcal{P} = \{\tilde{y}^j\}_{j=1}^{N_t}$ on the target domain. Being aware that a severe noise (due to domain shift [108]) is affecting such labels resulting in a significant amount of incorrect labels, we propose a way to progressively filter out noisy target samples from the clean ones, and carry out pseudo-label refinement to obtain a cleaner set $\mathcal{P}$.

Pseudo-Label Refinement. The first step for our proposed method refers to inferring pseudo-labels $\mathcal{P}$ of unlabeled data samples of $\mathcal{D}_t$ using f_s, a model pre-trained on the labeled source samples from $\mathcal{D}_s$. In this context, for *single-source* and *multi-target* UDA, the model pre-trained on the chosen source data is used to infer pseudo-labels of target domain(s) being considered. For *multi-source* UDA, we often have a single model pre-trained on aggregated data from all source domains, f_{agg}, which can be used to infer target pseudo-labels $\mathcal{P}$ as:

$$\tilde{y}^j = \text{argmax}(\psi_{agg}(\nu_{agg}(\boldsymbol{x}_t^j))) \quad \forall j \in \{1, ..., N_t\}, \tag{4.4}$$

Subsequently, the proposed label refinement procedure is carried out. In particular, to obtain robust and cleaner pseudo-labels by employing ensemble network, we use moving average of N_a previous ensemble output predictions (i.e., we keep the latest N_a outputs in a buffer and compute the mean). For a certain sample x, this results to:

$$\boldsymbol{p} = \sigma(\frac{1}{N_a \cdot N_e} \sum_{l=1}^{N_a} \sum_{k=1}^{N_e} \psi_e^{k,l}(\nu_e^{k,l}(\boldsymbol{x})) \tag{4.5}$$

where we set $N_a = 10$ for all the experiments in this study.[2] Now, let us define $\tilde{p}$ as the confidence of the pseudo-label $\tilde{y}$, namely the entry of the obtained probability vector $\boldsymbol{p}$ cor-

[2] We anticipate that this is a non sensitive parameter, it has not a relevant effect on the performance as long as $N_a \geq 5$.

responding to $\tilde{y}$.

We categorize the target samples with $\tilde{p} > \alpha$ as High Confidence Samples (HCS) which enable us to define the ratio of HCS over the total number of samples as:

$$\gamma = \frac{\#\ of\ HCS}{N_t} \tag{4.6}$$

Thus, the γ being used as a threshold (derived by parameter α and instantaneous generalization capability of the ensemble network *i.e.,* the ability of the network to discriminate clean data samples as HCS) is the key behind adaptive nature of the noise filtering ability of the proposed method. So, the pseudo-label of each sample $\boldsymbol{x}_t^j$ is updated/retained according to the following condition:

$$\tilde{y}^j(n) = \begin{cases} \text{argmax}(\boldsymbol{p}^j), & \text{if } \tilde{p}^j < \gamma \\ \tilde{y}^j_{(n-1)}, & \text{otherwise} \end{cases} \forall j \tag{4.7}$$

where n denotes the epoch number. The intuition behind such reassignment rule can be drawn from the trend of the prediction confidence along training. As shown in Fig. 4.3a and 4.3b, with the growing number of epochs, noisy samples remain towards low confidence regime and clean samples obtain high confidence progressively. In Fig. 4.3b, the ratio of HCS (with $\alpha = 0.9$) gives $\gamma \approx 0.15$ that corresponds to a confidence region $[0, \gamma] = [0, 0.15]$ in which the noisy samples are prevalent, hence they are the best candidates to be subjected to label reassignment. Consequently, pseudo-label refinement is achieved progressively during training in an adaptive manner (see Fig. 4.3c and 4.3d, where total noise is progressively reduced). We ablate to find optimal value of α in Sec. 5.6.3.

4.3.2 Negative Ensemble Learning

The adaptive pseudo-label refinement procedure discussed in Sec. 4.3.1 heavily depends on the diversity exists among ensemble members. Our approach induces such a diversity by different stochastic data augmentation and feedback. The latter is achieved by employing *Residual Labels (RL)* — randomly chosen complementary labels other than the inferred pseudo-label — a key attribute of our proposed Negative Ensemble Learning (NEL) loss that we define as:

$$\mathcal{L}_{ENL}(\mathcal{D}_t) = -\mathbb{E}_{x_t \sim \mathcal{D}_t} \frac{1}{N_{RL}} \sum_{c=1}^{C} \mathbb{1}_{[c \in RL]} log(1 - \boldsymbol{p}_c) \tag{4.8}$$

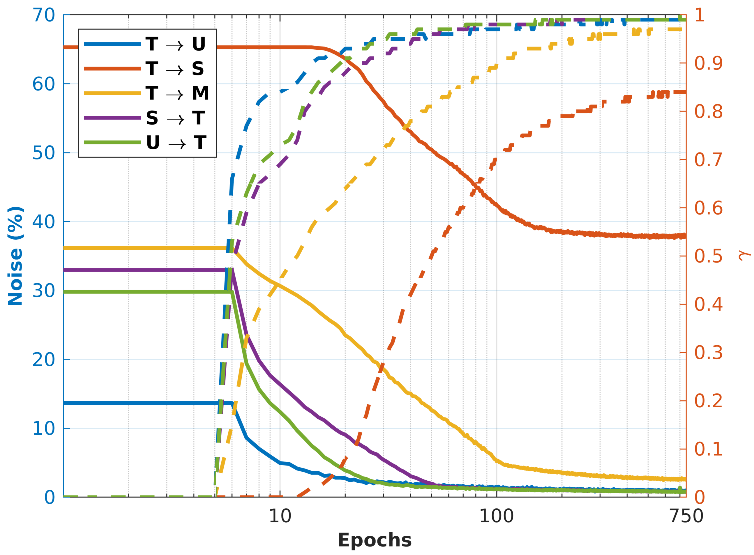


Figure 4.4: Correlation between adaptiveness of γ threshold (right y-axis, dashed lines) and progressive noise reduction (left y-axis, solid lines) achieved by NEL during training for various amount of noise. Legend: ***T****: MNIST,* ***S****: SVHN,* ***U****: USPS, and* ***M****: MNIST-M.*

The NEL loss in Eq. 4.8 is used to train each member independently, where N_{RL} refers to the number residual labels. Thus, for any value $N_{RL} > 1$, the proposed approach can influence the training process in three ways: (a) The likelihood of the actual-label y_t being randomly picked as one of the residual labels increases by factor of $\frac{N_{RL}}{C-1}$, which is bad. (b) In case $y_t \cap RL = \varnothing$, the training is accelerated with the stronger feedback provided by the multiple contributions of RLs, which is good. (c) In case $y_t \cap RL \neq \varnothing$, instead of providing entirely wrong feedback using Eq. 4.2, the impact of wrong feedback is mitigated by a factor of $\frac{N_{RL}-1}{N_{RL}}$, and this is again good. In fact, gradients will follow a mean direction according to Eq. 4.8. Although cases (b) & (c) are essential advantages of using multiple RL, (a) is a downside. To understand the balance among these aspects, we ablate on the different values of N_{RL} in Sec. 5.6.3.

However, we found that best results are obtained using a completely *disjoint* random subset of residual labels (DRL). Not only this allows each member to receive a different feedback (thus enhancing the ensemble's diversity), but also restricts the possibility of receiving wrong feedback to one member only. Thus, in this study, we don't use specific number of RL, rather, we use equally distributed DRL over all ensemble members.

Further, to induce additional diversity in ensemble expert members, we consider several standard stochastic data augmentation strategies including the composition of (i) spatial/geometric transformation via random cropping (with uniform area = 0.08 to 1.0 and aspect-ratio = $\frac{3}{4}$ to $\frac{4}{3}$) followed by resizing to the original size, (ii) affine transformation followed by

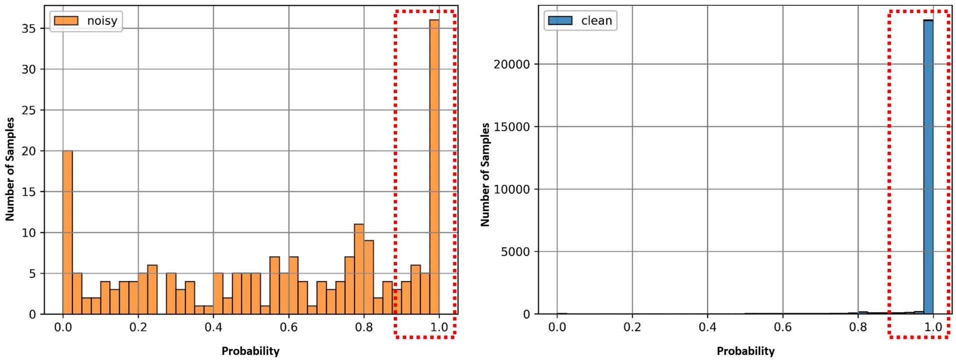


Figure 4.5: Distribution of remaining noise in refined pseudo-labels after SVHN→MNIST UDA. The highlighted bars (in a rectangle) represent the set of samples with confidence greater than $\alpha = 0.9$.

Gaussian blur, and (iii) color distortion. We found that composition of different stochastic data augmentation is crucial to avoid noise overfitting and extend diversity in the ensemble network (See Sec. 5.6.3).

All these ingredients concur in making proposed method capable of filtering out different amounts of noise in an adaptive manner while progressively refining pseudo-labels. In Fig. 4.4, the profiles related to the right y-axis (dashed lines) show the capability of the threshold γ to adapt to different single-source UDA cases, while the profiles related to the left y-axis show the corresponding noise reduction. In particular, it is worth noting that no pseudo-label refinement is applied during the first few epochs until the ensemble gets a mature state of generalization capability. As the ensemble members get more and more confident in the clean examples, label reassignment becomes more rigorous preventing the network to overfit to noisy samples.

After a certain number of training epochs, the pseudo-label refinement process stalls down to an insignificant noise reduction rate (see Fig. 4.4). Therefore, instead of pushing refinement process more for several epochs to achieve further small reduction, we apply standard supervised learning only using high confidence samples determined by α. This is done using only one model (the final target model) for a fair comparison with state-of-the-art methods. As shown in Fig. 4.5, the remaining noise is distributed over the entire probability spectrum, whereas the majority of the clean samples are predicted with high confidence (red box in the right plot). Therefore, only a small fraction of noisy samples affects training.

4.4 Experiments

We consider the image classification task to comprehensively evaluate the proposed method on major UDA benchmarks including Digit5 (MNIST [131], SVHN [132], USPS [133], MNIST-M [134], and Synthetic-Digits [134]), PACS [135], VisDA-C [136], and DomainNet [111]. For all the benchmarks, we use batch-size $= 32$, $\alpha = 0.9$ and Adam as the optimizer with a weight decay of $5e^{-4}$. The base learning rate is set to $1e^{-4}$ and the feature extractors are optimized with a learning rate of $1e^{-5}$. The feature extractor of ensemble members are initialized with a pre-trained source model. For single-source and multi-target UDA, we consider one pre-trained source model to adapt on every target domain. For multi-source UDA, each domain is selected as the target domain while the rest of the domains are treated as aggregated source (according to Eq. 4.4).

Digit5 refers to a set of digit benchmarks. In this study, following [111], we sample a subset of 25000 images from the training and 9000 images from the testing set for MNIST, MINST-M, SVHN, and Synthetic-Digits. Since USPS contain a total of 9298 images, we use standard train-test splits. To keep comparable image resolution, we resize all images to 32×32 and a naive 3-layer CNN is used as ensemble members. For single-source and multi-source UDA, label refinement takes 750 and 300 epochs, respectively, whereas the final target model is trained for 200 epochs.

PACS contains 4 domains, namely *(Art-Painting, Cartoon, Photo, and Sketch)*. There are only 9991 images of 227x227 resolution from 7 object categories that accommodates a large domain shift due to the different image style depictions. We use ResNet-18 as ensemble members. For single-source, multi-target and multi-source UDA, label refinement takes 200, 200 and 100 epochs, respectively, whereas the final target model is trained for 200 epochs.

Visda-C is a challenging large-scale benchmark attempting to bridge the significant synthetic-to-real domain gap across 12 object categories. We follow standard protocol in which the source domain (training split) contains 152K synthetic images and the target domain (testing split) contains 72K real images. We resize all images to 256×256 resolution and use ResNet-101 as ensemble members. Label refinement takes 150 epochs and just 25 epochs were found to be enough for training the final target model.

DomainNet is by far the largest UDA benchmark with 6 domains, 600K images and 345 categories. We resize all images to 256×256 and use ResNet-101 as ensemble members. It was mainly developed for multi-source UDA task for which our proposed label refinement takes 100 epochs for *Infogragh* and *Quickdraw* domain, while 40 epoch were enough for the rest. For training the final target model, 100 epochs were sufficient in all the cases.

4.4.1 Ablation study

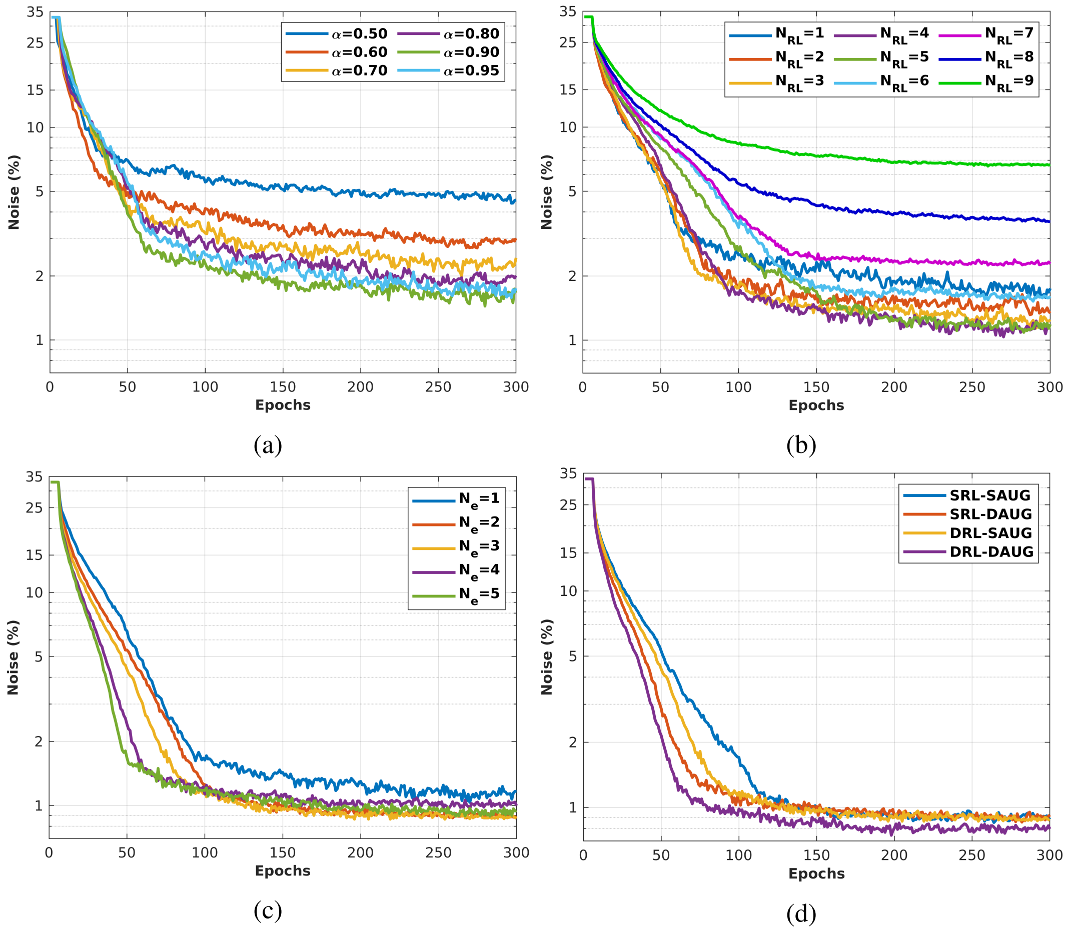


Figure 4.6: Ablation study considering SVHN→MNIST UDA task to determine optimal parameters of our proposed NEL method. (a): Single model is trained with 1 residual-label (RL) to choose the best α required to compute adaptive noise filtering threshold γ. (b): Searching for the right number of RL *i.e.,* N_{RL}. (c): Searching for the optimal number of members in the ensemble network ($N_{RL} = 4$ is used for $N_e = 1$). (d): Investigating the effect of same/disjoint RL (SRL and/or DRL) and same/different data augmentation (SAUG and/or DAUG) in all four possible scenarios.

We ablate the design choices described in Sec. 4.3 on the SVHN→MNIST adaptation task. It is important to note that the parameters estimated here are then used in all subsequent experiments, demonstrating the little sensitivity of the proposed method to such coefficients.

We consider $N_e = 1$ (*i.e.,* no ensemble) to ablate the different values of α. Fig. 4.6a shows that $\alpha = 0.90$ results in the highest noise reduction, whereas $\alpha = 0.50$ is the least effective. These findings make sense, since with $\alpha = 0.50$, the reassignment would start too early (*i.e.,* as soon as some example has confidence greater than 0.5, since γ is always zero beforehand), when the network is not yet "ready" for it. Instead, with $\alpha = 0.90$, reassignment will start only when some samples have very high confidence. From that moment on, γ starts adapting, so that the more the samples with high confidence, the more permissive the threshold γ will be. The reason $\alpha = 0.95$ results slightly less effective is because a relatively higher number of noisy samples overfits before they are reassigned. Keeping $N_e = 1$ and $\alpha = 0.9$, we ablate on the different values of parameter N_{RL} (Fig. 4.6b). Results show that $N_{RL} = 3, 4$, or 5 can be considered as the legitimate choice. Further, the results shown in Fig. 4.6c exhibit the improvement achieved by the RL approach in comparison to $N_e = 1$ (with $N_{RL} = 4$, *i.e.,* the best choice found in Fig. 4.6b). Though faster noise reduction is achieved with higher number of ensemble members, $N_e = 3$ can be regarded as the optimal choice considering performance *vs.* computational cost trade-off.

Also, as shown in Fig. 4.6d, just one type of augmentation for all the ensemble members (ref. SAUG in the figure) is not enough. Also, the results validate the ineffectiveness of using same residual labels (SRL in the figure) for all. The best noise reduction is achieved using Disjoint Residual Labels along with different stochastic augmentations, DRL and DAUG, respectively.

4.4.2 Performances

The reported results in Tab. 4.1-4.4 present the average accuracy of 3 runs. In Tab. 4.1 *(left)*, we compare our method (NEL) with the existing methods which address the challenging *MNIST→SVHN* and *MNIST→MNIST-M* tasks in a multi-target UDA framework. In both cases, the source contains gray-scale images, and the target holds colored (RGB) images, carrying a massive distribution gap across domains for which our method achieves third and second-best performance, respectively. Nevertheless, by outperforming in 3 out of 5 cases, our method achieves state-of-the-art average accuracy. For multi-source UDA task in Tab. 4.1 *(right)*, the performance of proposed method is slightly affected while adapting to *Synthetic-Digits* benchmark. In the remaining 4 out of 5 cases, our method outperforms existing methods and achieves state-of-the-art average accuracy. Especially, the difference is substantial in the case of *MNIST-M*.

In Tab. 4.2 *(left)*, we compare NEL with the existing methods addressing multi-target UDA on PACS. As can be noticed, despite the sub-optimal performance in 2 cases, our method achieves superior average accuracy. For multi-source UDA, we compare recent works in Tab. 4.2 *(right)*. Also in this framework, our method consistently outperforms existing methods, with only in one case getting lower, yet comparable, accuracy. To the best of our knowledge, we are the first to report single-source UDA results on PACS. So, in Tab. 4.5, we consider similar pairs as of Tab. 4.2 *(left)* to evaluate the performance difference. As expected, single-source UDA brings comparatively better performance because of the pairwise

UDA.

Table 4.1: Classification accuracy on Digit5 with a naive 3-layer CNN. Legend: ***T***: *MNIST,* ***S***: *SVHN,* ***U***: *USPS,* ***M***: *MNIST-M, and* ***D***: *Synthetic-Digits.*

	Single-Source UDA						Multi-Source UDA						
Source	***T***	***T***	***T***	***S***	***U***	Avg.		***M,S, D,U***	***T,S, D,U***	***T,M, D,U***	***T,M, S,U***	***T,M, S,D***	Avg.
Target	***U***	***S***	***M***	***T***	***T***			***T***	***M***	***S***	***D***	***U***	
ATT [137]	–	52.8	94.0	85.8	–	–	DCTN [138]	–	70.9	77.5	–	–	–
SBA [139]	97.1	50.9	**98.4**	74.2	87.5	81.6	MM [111]	98.4	72.8	81.3	89.5	96.1	87.6
MALT [140]	97.0	**78.7**	71.4	98.7	20.7	73.3	OML [141]	98.7	71.7	84.8	91.1	97.8	88.8
MTDA [6]	94.2	52.0	85.5	84.6	91.5	81.5	CMSS [142]	99.0	75.3	88.4	**93.7**	97.7	90.8
GPLR [5]	89.3	63.4	94.3	97.3	91.8	87.5							
NEL	**97.4**	61.6	95.4	**99.2**	**99.2**	**90.6**		**99.1**	**95.5**	**89.6**	90.0	**97.8**	**94.4**

Table 4.2: Classification accuracy on PACS with ResNet18. * results are taken from [6]. Legend: ***A***: *Art-Painting,* ***C***: *Cartoon,* ***P***: *Photo, and* ***S***: *Sketch.*

	Multi-Target UDA							Multi-Source UDA					
Source	***P***			***A***			Avg.		***C,P,S***	***A,P,S***	***A,C,S***	***A,C,P***	Avg.
Target	***A***	***C***	***S***	***P***	***C***	***S***			***A***	***C***	***P***	***S***	
1-NN*	15.2	18.1	25.6	22.7	19.7	22.7	20.7	DD [143]	87.5	87.0	96.6	71.6	85.7
ADDA*	24.3	20.1	22.4	32.5	17.6	18.9	22.6	SIB [144]	88.9	89.0	98.3	82.2	89.6
DSN*	28.4	21.1	25.6	29.5	25.8	24.6	25.8	OML [141]	87.4	86.1	97.1	78.2	87.2
ITA*	31.4	23.0	28.2	35.7	27.0	28.9	29.0	RABN [145]	86.8	86.5	98.0	71.5	85.7
KD [146]	24.6	32.2	**33.8**	35.6	46.6	**57.5**	46.6	JiGen [147]	84.8	81.0	97.9	79.0	85.7
								CMSS [142]	88.6	**90.4**	96.9	82.0	89.5
NEL	**80.1**	**76.1**	25.9	**96.0**	**82.8**	49.8	**68.4**		**90.8**	89.5	**98.8**	**85.2**	**91.1**

Table 4.3: Classification accuracy on Visda-C with ResNet101.

Methods	***plane***	***bcycl***	***bus***	***car***	***horse***	***knife***	***mcycl***	***person***	***plant***	***skate***	***train***	***truck***	Avg.
MCD [148]	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
GPDA [149]	83.0	74.3	80.4	66.0	87.6	75.3	83.8	73.1	90.1	57.3	80.2	37.9	73.3
SAFN [150]	93.6	61.3	84.1	70.6	94.1	79.0	**91.8**	79.6	89.9	55.6	89.0	24.4	76.1
DSBN [113]	**94.7**	**86.7**	76.0	72.0	**95.2**	75.1	87.9	81.3	**91.1**	68.9	88.3	45.5	80.2
DADA [115]	92.9	74.2	82.5	65.0	90.9	**93.8**	87.2	74.2	89.9	71.5	86.5	48.7	79.8
NEL	94.5	60.8	**92.3**	**87.3**	87.3	93.2	87.6	**91.1**	56.9	**83.4**	**93.7**	**86.6**	**84.2**

Table 4.4: Classification accuracy on DomainNet with ResNet101. For each target, the rest of the domains are considered as source (multi-source UDA). Legend: ***C***: *Clipart*, ***I***: *Infograph*, ***P***: *Painting*, ***Q***: *Quickdraw*, ***R***: *Real, and* ***S***: *Sketch.*

Target	***C***	***I***	***P***	***Q***	***R***	***S***	Avg.
MM [111]	58.6	26.0	52.3	6.3	62.7	49.5	42.6
OML [141]	62.8	21.3	50.5	15.4	64.5	50.4	44.1
CMSS [142]	64.2	**28.0**	53.6	16.0	63.4	53.8	46.5
NEL	**68.3**	22.1	**54.7**	**22.8**	**67.3**	**57.1**	**48.7**

Table 4.5: Classification accuracy on PACS with ResNet18.

	Single-Source UDA						
Source	***P***	***P***	***P***	***A***	***A***	***A***	Avg.
Target	***A***	***C***	***S***	***P***	***C***	***S***	
NEL	**82.6**	**80.5**	**32.3**	**98.4**	**84.3**	**56.1**	**72.4**

In Tab. 4.3, along with 2 comparable results for Visda-C, the proposed method achieves superior performance in 6 out of 12 categories that give rise to state-of-the-art average accuracy on such a challenging benchmark. Also In Tab. 4.4, except one case, NEL consistently outperforms existing methods despite the large number of classes and discrepancy across domains.

4.4.3 Discussion.

In the pseudo-label refinery framework, the single-source and multi-target UDA scenarios can be considered as the most challenging tasks since the pre-trained source model is optimized for one particular data distribution only. Consequently, inferred pseudo-labels are affected by a relatively higher amount of shift-noise with respect to the multi-source UDA scenario. Thus, in such cases, NEL requires a bit larger amount of epochs for filtering noise and, thus, refining pseudo-labels. On the other hand, starting from a better pre-trained source model in a multi-source UDA scenario, NEL performs better and faster. Moreover, there is no existing method in the literature that targets all three frameworks (*i.e.,* single-source, multi-target, and multi-source UDA) at a time. To sum up, NEL outperforms existing methods in all scenarios, even without using source data, which highlights the general applicability of the proposed method to cope with challenging tasks of different levels of complexity.

4.5 Summary

In this work, we cast UDA as a pseudo-label refinery problem in the challenging source-free scenario. We propose Negative Ensemble Learning technique, which takes advantage of different data augmentation and feedback using Disjoint Residual Labels to diversify the learning of the ensemble members. Thanks to this new training procedure, we were able to obtain an extraordinary cleaning of the target data labels. It requires a minimal tuning of parameters (estimated once and fixed for all the experiments), and can work in single-source, multi-target, and multi-source scenarios indifferently, unlike the existing methods in the literature. Results demonstrate the actual goodness of the proposed approach, outperforming the state-of-the-art average performances in all the challenging public benchmarks.

Chapter 5

Continual Source-Free Unsupervised Domain Adaptation

This study is mainly an extension of our previous work that we described in Chapter 4. Precisely, stage 1 of the newly proposed method (see Fig. 5.2) does not introduce anything different. Thus, reported results related to stage 1 in this chapter (*i.e.,* Pseudo-Label Refinement (PLR) in reported tables) are similar as well. However, the target sample selection technique (after stage 1) and the entire stage 2 are the novel contributions of this study. Therefore, in the following, we briefly recap what we have already discussed in Chapter 4, followed by a detailed description of newly proposed approaches.

5.1 Introduction

Convolutional Neural Networks (CNNs) trained on a labeled *source* domain often fail to generalize well on a related but different *target* domain due to the well-known *domain-shift* problem [108, 5]. Since annotating data from a new domain is expensive and sometimes even impossible, Unsupervised Domain Adaptation (UDA) methods have been developed to address the drop in performance exploiting *unlabelled* target data.

Starting from earliest works, most conventional UDA methods address the adaptation task by somehow bridging source and target distributions by either enforcing discriminative features alignment between domains [109], matching moments [111], applying domain-specific batch normalization [113], or adopting domain adversarial learning strategies [115]. However, these methods require joint access to both *labeled source* and *unlabeled target* domains during adaptation, making them unsuitable for most real-world scenarios, where source data is inaccessible (*e.g.,* due to data privacy or proprietary reasons). Nevertheless, the performance on source often degrades after adaptation to target, even despite the use of source data during training [151].

Recently, UDA methods started addressing the problem under a more realistic source-free assumption, *i.e.,* by using the pre-trained source model and unlabeled data from the target domain only [122, 5, 152]. However, due to absolutely no exposure to source data distribution by any means, these methods naturally undergo catastrophic forgetting, *i.e.,* the

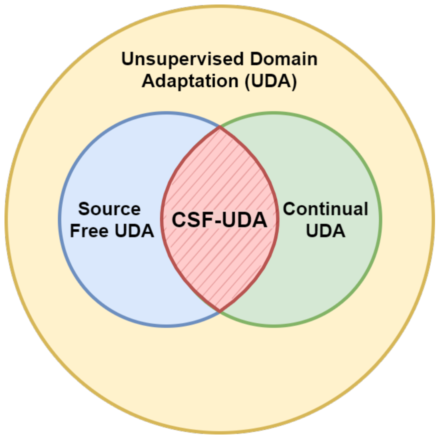


Figure 5.1: Our proposed CSF-UDA method addresses unsupervised domain adaptation in a challenging *continual learning* framework in *source-free* settings, *i.e.,* it aims at mitigating catastrophic forgetting without accessing source data.

model optimized for the target experiences a substantial drop in performance if tested back on the source domain. This is certainly unacceptable in many real-world applications (*e.g.,* autonomous driving/navigation). Moreover, in either case, many UDA methods either focus on the single-source or single-target scenario, regardless of the fact that data may belong to multiple sources or target distributions, *e.g.,* images obtained in different environments or from the web, such as sketches, paintings, or photos for instance.

This brings up an interesting question: can we adapt a source model to a target domain while preserving good performance on the source, yet assuming no access to source data? To address this challenge, we propose a Continual Source-Free Unsupervised Domain Adaptation (CSF-UDA) pipeline comprising two main stages (See Fig. 5.2).

Stage 1 performs Source-Free UDA, assuming the availability of a model pre-trained on the source domain to infer *pseudo-labels* for unlabeled target samples. This results in a significant amount of incorrect pseudo-labels, which is a consequence of the *domain-shift* [108, 5]. So, in absence of source data, it appears natural to adapt to the target domain by refining the inferred pseudo-labels. To this end, we propose an Ensemble Learning method that progressively disentangles samples having noisy pseudo-labesl from the clean ones, by assigning them higher and lower probability, respectively. Low-probability samples iteratively undergo a pseudo-label refinement process via reassignment.

Stage 2 consists in an image synthesis process aimed at promoting continual learning, by preserving performance on source data. We leverage the fact that CNNs are capable of automatically discovering the rich underlying patterns hidden in the data, such as running average statistics stored in the Batch-Norm (BN) layers. The idea here is to exploit such

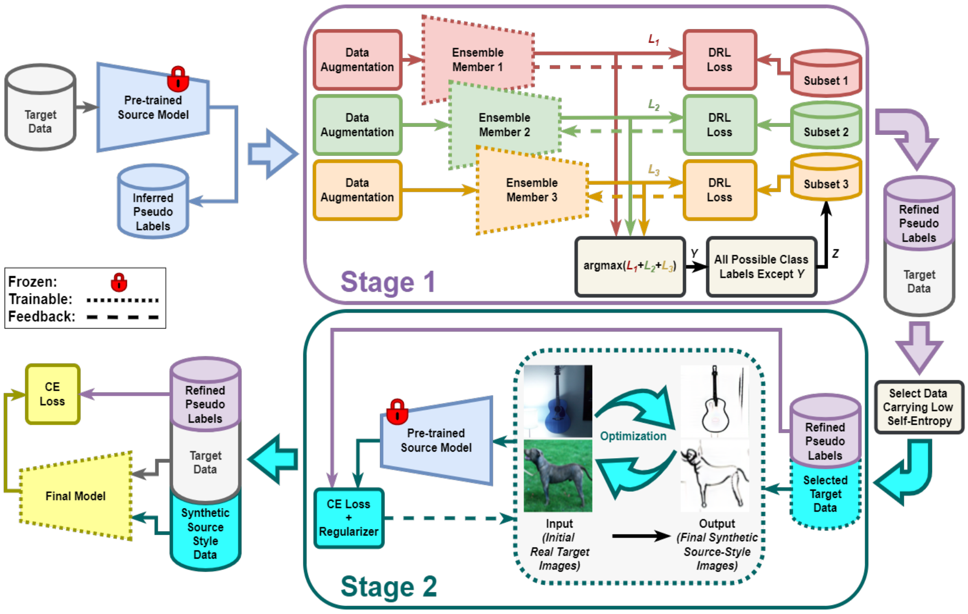


Figure 5.2: Overview of proposed CSF-UDA pipeline. We assume a pre-trained source model to infer pseudo-labels for the target set. *Stage 1* refines wrong pseud-labels attaining source-free UDA, while *Stage 2* synthesizes source-style images to avoid catastrophic forgetting, thus accomplishing continual adaptation. A single final model is trained on refined target plus synthetic-source images.

rich information to generate class-conditioned source-style images, employing feature distribution regularizers, and forcing images to produce sharp classification predictions on the source model. Specifically, we use target images as prior and optimize their statistics and style such that the transformed version of the generated images resembles source domain distribution. Eventually, we mix synthetic source and real target images for training a final model that ensures good performance on both real source and target domains.

With extensive experiments on various benchmarks yielding a wide range of label noise (*e.g.,* PACS: 4% – DomainNet: 82.4%), we show that a model trained with refined pseudo-labels outperforms state-of-the-art UDA methods by a considerable margin. Additionally, we show that the second stage of our proposed pipeline helps alleviating catastrophic forgetting when the model is tested back on the source domain(s). To summarise, the contributions of our work can be stated as follows:

- In order to attain source-free UDA, we propose a new, fully-adaptive method that dynamically filters out label noise and assigns cleaner pseudo-labels to noisy target samples. To do so, we introduce a novel ensemble learning method that employs disjoint feedback.

- In order to fulfil continual UDA, we propose a new method that optimizes target images into high-fidelity class-conditional source images given just a pre-trained source model. Such synthetic images help preserving good performance on source domain(s) without the need of actual source samples.

- We validate our method on four well-known benchmarks, demonstrating its superiority over state-of-the-art UDA methods with a significant margin in single-source, multi-source and multi-target scenarios, most notably, up to 21.8% better accuracy on PACS.

5.2 Related Work

As mentioned in the introduction, our work lies at the intersection of *Source-Free UDA* and *Continual Learning* (CL), whose related literature is discussed below. Also, we briefly review existing literature related to *Learning with Noisy Labels*, which matches our learning scenario and to *Image Synthesis*, as synthesized source images constitute a core stage of the proposed pipeline.

Source-Free UDA. Recent years have seen growing interest in addressing the UDA problem in realistic source-free settings. A common approach is to leverage a pre-trained source model for either transferring the fixed source classifier to the target data employing information maximization and pseudo-labeling [152], adaptive adversarial network with contrastive category-wise matching module [153], or updating the target model progressively by generating target-style samples through conditional generative adversarial networks [5], also by combining clustering-based regularization [122]. Similarly, to improve performance in domain-adaptive person re-identification tasks, [123] proposes a pseudo-label cleaning process with online refined soft pseudo-labels. Our proposed approach lies in this category of works, but we differ from previous methods by not requiring: i) a customized network, ii) generating target-style data using *e.g.,* GAN-based models that require careful hyper-parameter tuning to reach stability, and iii) specific approaches to tackle single-source, multi-source, and multi-target UDA differently.

Continual Learning. CL refers to learning by using many diverse data distributions (tasks) sequentially, while avoiding catastrophic forgetting [154, 155]. Some recent works tackled the continual UDA problem such as, *e.g.,* [156], which introduces a network with domain-specific alignment layers supporting domain relation graphs, or [157], which proposes domain adversarial learning with sample replay. More recently, [158] uses gradient regularized contrastive learning with domain memorization constraint, and [159] employs a meta-learning strategy and domain randomization using heavy image manipulations. Despite the fact that these strategies are effective, they all require the access to source data. Also related, [160] performs source-free adaptation in different settings, focusing on class incremental single target domain adaptation, assuming one-shot labeled target data per class. [161] performs continual adaptation but needs two auxiliary binary embedding layers to be specifically trained on source. These are stored and used during adaptation in order to overcome catastrophic forgetting, and this questions the source-free nature claimed by the method. On the contrary, we only assume any standard CNN networks pre-trained on the source to be available.

Learning with Noisy Labels. Deep models are capable of memorizing the full training data even when labels are noisy [3]. To overcome such overfitting issue, existing methods try to select a subset of possibly clean labels for training, *e.g.,* using two networks subject to a co-teaching strategy [128], adopting meta-learning for exemplar weight estimation [129], applying one-out filtering approach based on local and global consistency [130], or using Negative Learning (NL) as an indirect learning method [3]. These methods typically address ideal artificial noise characterized by uniform distributions (*Symmetric-Noise*), or by mimicking certain structure (*Asymmetric-Noise*) [4]. This is completely different from *shift-noise*, as defined in [5], which is caused by the domain shift in UDA (see Sec. 5.3) and represents a realistic scenario. Our method takes partial inspiration from [3], which indeed fails when noise is not ideal, also being affected by threshold sensitivity that limits its generalization capability across benchmarks. In contrast, our proposed method features a fully *adaptive* procedure to progressively filter out the challenging *shift-noise* affecting target pseudo-labels and refine it through a reassignment process.

Image Synthesis. In the deep learning era, this area refers to the generation of synthetic images, possibly indistinguishable from real ones, and generative adversarial networks (GAN) are the most popular class of approaches adopted. GAN-inversion is an interesting research direction, in which an anchor image is used to guide a GAN to generate realistic images by inverting a pre-trained model [162, 163]. Other works focus on network inversion that enables noise-to-image transformation by back-propagating gradients to the learnable input images like in, *e.g.,* [164], which introduces “dreaming” new visual features onto images, while [165] takes this approach a step further to generate more realistic images. Recently, [166] improved synthesis quality by using a regularizer based on feature matching, and demonstrated its utility for data-free knowledge distillation. In the context of Domain Generalization, [167] optimizes images in the pixels space to produce augmentations, extending the working domain of the classifier to unseen data distributions. On the contrary, we optimize target images in order to produce sharp activation in the source model’s prediction, *i.e.,* we alter them (via back-propagation in the pixel space), until they trigger good predictions in the original source model.

5.3 Method

Our method comprises two main stages (refer to Fig. 5.2). After inferring pseudo-labels for the target samples using the source model, Stage 1 is devoted to refine the so called shift-noise [5] (affecting target samples) resulting in cleaner pseudo-labels. The proposed strategy, detailed in section 5.3.1, is sufficient to obtain state-of-the-art performance on the target set in the source-free setting, yet nothing prevents catastrophic forgetting. For this reason, the target set with refined pseudo-labels is exploited in Stage 2 as a prior to synthesize source-style images which can "anchor" the model to its original performance on the source set, as detailed in section 5.3.2.

Preliminaries. The goal of UDA is to adapt a model pre-trained on a labelled source domain $\mathcal{D}_s = \{(\boldsymbol{x}_s^i, y_s^i)\}_{i=1}^{N_s}$ on a different, yet related, unlabeled target domain $\mathcal{D}_t = \{\boldsymbol{x}_t^j\}_{j=1}^{N_t}$, which share the same label set, *i.e.,* $\mathcal{Y}_s = \mathcal{Y}_t$. We assume $\mathcal{D}_s$ is *never* available in a realistic source-free scenario, while of course a pre-trained source model $f_s(\cdot)$ is at our disposal. This can be used to infer pseudo-labels $\mathcal{P} = \{\tilde{y}^j\}_{j=1}^{N_t}$ for the target domain:

$$\tilde{y}^j = \operatorname{argmax} f_s(\boldsymbol{x}_t^j), \;\; j = 1...N_t \tag{5.1}$$

Clearly, a severe noise affects $\mathcal{P}$: a significant amount of pseudo-labels is wrong due to domain shift. The following section discusses how our method progressively filters such noise and subsequently obtains a cleaner set $\mathcal{P}$, which in turn translates into better accuracy on the target set.

5.3.1 Pseudo-label refinement for Source-Free UDA

Naively training a target model with cross-entropy given $\mathcal{P}$ as supervisory signal (ground-truth) eventually results in the overfitting noisy samples. In our settings, in case of wrong pseudo-labels $\tilde{y}^j \neq y_t^j$ (y_t^j are *inaccessible* target labels), the model would undeniably try to maximize the probability of a sample belonging to the wrong class. Methods based on negative feedback, such as Negative Learning (NL) [3], try to alleviate such overfitting by reducing the probability of providing wrong information from noisy labels. Specifically, for a given sample $\boldsymbol{x}_t^j$ instead of using its given pseudo-label $\tilde{y}^j$, NL employs a complementary label $\bar{y}^j$ (randomly selected from $\{1, \dots, C\} \backslash \{\tilde{y}\}$) as a supervisory signal and trains the model to minimize the probability of $\boldsymbol{x}_t^j$ belonging to $\bar{y}$. The loss function is defined as:

$$\mathcal{L}_{NL}(\mathcal{D}_t) = -\mathbb{E}_{x_t \sim \mathcal{D}_t} \sum_{c=1}^{C} \mathbb{1}_{[c=\bar{y}]} \log(1 - p^c) \tag{5.2}$$

where $\mathbb{1}$ is an indicator function, $\boldsymbol{p} = \{p^1, \ldots, p^c\} = \boldsymbol{\sigma}(f_t(\boldsymbol{x}_t))$, and $f_t(\cdot)$ is the target model being trained. Since in the case $\tilde{y}^j \neq y_t^j$, the chances of selecting exactly y_t^j as a complementary label are only $\frac{1}{(C-1)}$, the loss of Eq. 5.2 reduces the risk of providing incorrect feedback to $f_t(\cdot)$. Consequently, samples carrying clean $\tilde{y}$ attains higher confidence as complementary labels are pushed towards zero. On the contrary, samples carrying noisy $\tilde{y}$ struggles with scarcely confident, providing a criterion for distinguishing them, as shown in Fig. 4.2a.

Nevertheless, naively applying Eq. 5.2 cannot tackle UDA, as we show in Fig. 4.2b: while noisy samples are clearly assigned a low confidence in case of symmetric-noise, they are over-fitted in the case of *shift-noise*. Note that overfitting is not to be ascribed to the *amount* of noise, which was carefully balanced, but rather to its extremely biased *distribution* (more details in supp. mat.).

The first strategy we put forward to mitigate such limitations leverages Ensemble Learning, which denotes the technique of concurrently training multiple networks and averaging their output. We discovered that the loss defined in Eq. 5.2 delivers its best when we further distribute the probability of providing wrong feedback across the ensemble members. This is done by training them independently, with disjoint subsets of complementary labels. In case of wrong pseudo-label $\tilde{y}$, the correct label y_t is wrongly provided as one of the complementary labels to only one member, while the others always learn from clean information. More in detail, we introduce the novel idea of *Disjoint Residual Labels (DRL)* and the loss:

$$\mathcal{L}_{DRL}(\mathcal{D}_t) = -\mathbb{E}_{x_t \sim \mathcal{D}_t} \frac{1}{N_{DRL}} \sum_{c=1}^{C} \mathbb{1}_{[c \in DRL]} log(1 - p^c) \tag{5.3}$$

which is used to train each member *independently*. The predictions of the members are then late-fused via

$$\boldsymbol{p}_e = \boldsymbol{\sigma}\left(\frac{1}{N_e \cdot N_a} \sum_{k=1}^{N_e} \sum_{l=1}^{N_a} f^{k,a}(\boldsymbol{x})\right) \tag{5.4}$$

where f^k are the N_e members and we use a moving average of N_a previous outputs ($N_a = 10$ for all the experiments).

Online Adaptive Pseudo-Labelling. The Ensemble learning strategy using DRL aims at obtaining the ideal situation depicted in Fig. 4.2a, where noisy pseudo-labels can be efficiently separated and re-assigned. We found that if the pseudo-label assignment is not progressively executed during training, many noisy samples attain high confidence, which leads to a sub-optimal performance on target (more details in supp. mat.). This drove us towards devising a more sophisticated procedure.

Let us define $\tilde{p}$ as the confidence of the pseudo-label $\tilde{y}$ (namely the entry of the probability vector $\boldsymbol{p}_e$ corresponding to $\tilde{y}$). We define the target samples carrying $\tilde{p} > \alpha$ as High Confidence Samples (HCS) which enable us to define the ratio of HCS over the total number of samples as:

$$\gamma = \frac{N_{HCS}}{N_t}. \tag{5.5}$$

Note that once α is fixed, γ depends on the current generalization capability of the ensemble network providing us with an adaptive filtering criterion. The pseudo-label of each sample $\boldsymbol{x}_t^j$ is then updated according to the following rule:

$$\tilde{y}_n^j = \begin{cases} \text{argmax}(\boldsymbol{p}_e^j), & \text{if } \tilde{p}^j < \gamma \\ \tilde{y}_{n-1}^j & \text{otherwise} \end{cases} \quad \forall j \tag{5.6}$$

where n denotes the epoch. As shown in Fig. 4.3a and 4.3b, with the growing number of epochs, noisy samples remain towards low confidence regime and clean samples obtain high confidence progressively. In Fig. 4.3b, the ratio of HCS (with $\alpha = 0.9$) gives $\gamma \approx 0.15$ that corresponds to a confidence region $[0, \gamma] = [0, 0.15]$ in which the noisy samples are prevalent, hence they are the best candidates to be subjected to label reassignment. Consequently, pseudo-label refinement is achieved progressively during training in an adaptive manner (see Fig. 4.3c and 4.3d, where the total noise is progressively reduced). Correlation between the adaptiveness of γ threshold and progressive noise reduction achieved during training for various amount of noise is discussed in Appendix.

5.3.2 Image Synthesis for Continual Adaptation

As a result of Stage 1, a refined pseudo-label is associated to each target sample. Note that, at this point, some noise still remains in pseudo-labels, *i.e.,* some of them still do not correspond to the correct ground-truth target label. Since we need to generate class-conditioned source-style images, by leveraging the source model together with $(\boldsymbol{x}_t^j, \tilde{y}_t^j)$ as a prior, we feed Stage 2 with only a subset of the target set which most likely guarantees less noise. More in detail, we compute prediction uncertainty of the target samples quantified by self-entropy [168] as:

$$H(\boldsymbol{x}_t) = -\sum \boldsymbol{p}(\boldsymbol{x}_t) log(\boldsymbol{p}(\boldsymbol{x}_t)), \tag{5.7}$$

where smaller entropy indicates more confident prediction (more details in supp. mat.). Based on this, we sort the target samples in ascending order and select only the first N_h samples from each class. These selected target samples are then optimized by minimizing

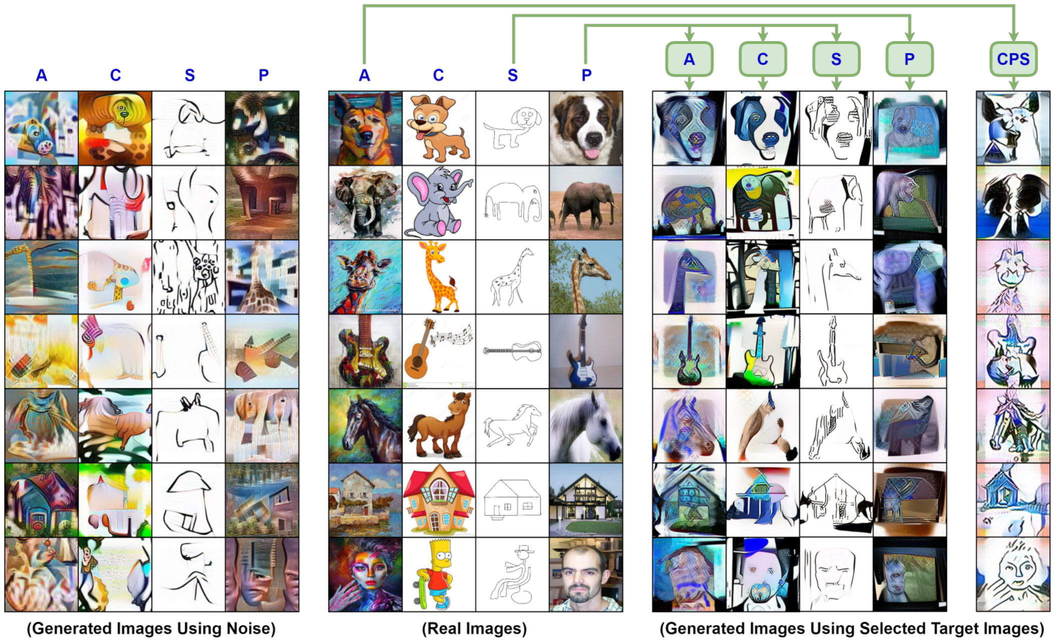


Figure 5.3: Image Synthesis (PACS dataset): Source style images (*right*) are optimized from target images with refined pseudo-labels (*center*). we also provide an example of images synthesized from random noise (*left*). CPS exemplifies a multi-source case. Legend: ***A***: *Art-Painting,* ***C***: *Cartoon,* ***P***: *Photo, and* ***S***: *Sketch.*

the cross-entropy loss for the original source model and two feature distribution regularization terms. Note that here the source model (the only available information we have about the source domain) is kept frozen while we optimize pseudo-labeled target samples $(\boldsymbol{x}_t^j, \tilde{y}_t^j)$ in the pixel space:

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta \nabla_{\boldsymbol{x}} \mathcal{L}(f_s(\boldsymbol{x}), \tilde{y}), \quad \boldsymbol{x}_0 = \boldsymbol{x}_t^j \tag{5.8}$$

$$\begin{aligned} \mathcal{L}(f_s(\boldsymbol{x}), \tilde{y}) = &\ell_{CE}((f_s(\boldsymbol{x})), \tilde{y}) + \\ &\lambda_{TV} \mathcal{R}_{TV}(\boldsymbol{x}) + \lambda_{BN} \mathcal{R}_{BN}(\boldsymbol{x}), \end{aligned} \tag{5.9}$$

$$\mathcal{R}_{TV}(\boldsymbol{x}) = \sum_{u,v} ((\boldsymbol{x}_{u,v+1} - \boldsymbol{x}_{uv})^2 + (\boldsymbol{x}_{u+1,v} - \boldsymbol{x}_{uv})^2)^{\frac{1}{2}},$$

$$\mathcal{R}_{BN} = \sum_{l,j} \parallel \mu_l(\boldsymbol{x}^j) - \mu_l \parallel + \parallel \sigma_l^2(\boldsymbol{x}^j) - \sigma_l^2) \parallel_2,$$

Here $\ell_{CE}(\cdot)$ is the cross-entropy loss, f_s is the frozen source model, λ_{TV} and λ_{BN} are scalar weights.

$\mathcal{R}_{TV}$ is a regularizer that penalizes the *Total Variation norm* approximated as finite pixel difference (u, v are pixel indexes)[169]: it provides more stable convergence and encourages $\boldsymbol{x}$ to consist of piece-wise uniform patches.

$\mathcal{R}_{BN}$ enforces batch-wise (j is the batch index) feature statistics similarities at all layers l, exploiting the BatchNorm running average parameters (μ_l, σ_l) stored in the source models, which implicitly capture the channel-wise means and variances of the original source images [166].

In principle, images can be generated by optimizing random noise [166]. But we verified that starting from the actual target images guarantees higher fidelity, realism, and most of all diversity in generated synthetic images (see Fig. 5.3). We also found that just a handful of synthetic source-style samples generated by our proposed method effectively helps in preserving source performance.

Final Model. We initialize the weights of the final trainable model with the source model. Subsequently, the weights of fully-connected (FC) layers are frozen while feature-extractor (FE) remains trainable. We train the final model with standard cross-entropy loss using both real target and synthetic source-style images together with corresponding refined pseudo-labels. Freezing FC layers not only help FE to learn the representation of synthetic source images most likely to be identical to the real source domain but also forces the target domain to get aligned with the source.

5.4 Experiments

For the image classification task, we evaluate performance of our method via extensive experiments on major UDA benchmarks including, Digit5 (MNIST [131], SVHN [132], USPS [133], MNIST-M [134], and Synthetic-Digits [134]), PACS [135], VisDA-C [136], and DomainNet [111].

For single-source and multi-target UDA, we consider one pre-trained source model to adapt on every target domain. For multi-source UDA, one domain is selected as the target domain while the source model is assumed to be pre-trained on rest of the (aggregated) source domains. As there are two stages in the proposed pipeline, we discuss adopted settings for each stage separately in the following sections.

5.4.1 Source-Free UDA

For stage 1, we use an ensemble network comprising 3 members, $\alpha = 0.9$, $batch\text{-}size = 32$, $Adam$ as an optimizer with a $weight\text{-}decay = 5e^{-4}$, and learning rate of $1e^{-4}$ and $1e^{-5}$ for fully-connected layers and feature extractor, respectively. After ablating such parameters for the SVHN→MNIST case, we kept them fixed for all the other benchmarks. The detailed ablation is available in the Appendix.

We accomplish Source-Free UDA with Stage 1 alone: the outcome is called pseudo-label refinement (**PLR**).The obtained percentage of pseudo-labels correctly refined in Stage 1, in fact corresponds to the accuracy of our method on target domain. However, for a fair comparison with other methods, we use output of the Stage 2 (refined pseudo-labels) to train a ***single*** model for 200 epochs; the outcome is called **SF-UDA**. For all datasets, backbones are the standard ones employed in previous UDA literature.

Digit5 refers to a set of five digit benchmarks. Following [111], we sample a subset of 25000 images from the training and 9000 images from the testing set for MNIST, MINST-M, SVHN, and Synthetic-Digits. USPS contains a total of 9298 images, so we use standard train-test splits. We resize images to 32×32 and a naive 3-layer CNN is used as ensemble member. For single-source & multi-source UDA, label refinement takes 750 and 300 epochs, respectively.

PACS contains 4 domains, namely *(Art-Painting, Cartoon, Photo, and Sketch)*. There are only 9991 images of 227×227 resolution from 7 object categories that accommodates a large domain shift due to the different image style depictions. We use ResNet-18 as ensemble members For single-source, multi-target and multi-source UDA, label refinement takes 200, 200 and 100 epochs, respectively.

Visda-C is a challenging large-scale benchmark attempting to bridge the significant synthetic-to-real domain gap across 12 object categories. We follow standard protocol in which the source domain contains $152K$ synthetic images. We use images of 256×256 resolution and use ResNet-101 as ensemble members. Label refinement takes 150 epochs.

DomainNet is by far the largest UDA benchmark with 6 domains, $600K$ images and 345 categories. We use images of 256×256 resolution and use ResNet-101 as ensemble members. It was mainly developed for multi-source UDA task for which our proposed label refinement takes 100 epochs for Infogragh and Quickdraw domain, while 40 epoch were enough for the rest.

5.4.2 CSF-UDA with Synthetic Source-Style Images

We now examine the effectiveness of stage 2 *i.e.,* the usefulness of synthesized images in preserving good performance on the real source domain. Note that in stage 2, we have access to the pre-trained source model and the target images with associated refined pseudo-labels (output of stage 1) only. For single-source and multi-source continual source-free UDA, we synthesis a relevant source for each target. For multi-target case, one synthesized source is enough for all related targets.

Table 5.1: Classification accuracy on PACS with ResNet18. Legends: ***Sc:*** *Source (Real),* ***Tg:*** *Target (Real),* ***SynSc****: Synthetic Source (Generated),* ***SF-UDA:*** *Output of Stage 1 (Source-Free UDA),* ***CSF-UDA:*** *Output of Stage 2 (Continual Source-Free UDA),* ***A****: Art-Painting,* ***C****: Cartoon,* ***P****: Photo, and* ***S****: Sketch.*

Single-Source CSF-UDA (PACS dataset)								
Train		Sc	Tg		Tg+***SynSc***		Tg+Sc	
Test		Tg	Tg	*Sc*	Tg	*Sc*	Tg	*Sc*
Sc	Tg		SF-UDA		CSF-UDA		Baseline	
A	**C**	58.1	84.3	*63.0*	84.5	*81.0*	98.6	*98.1*
	P	96.0	98.4	*62.5*	98.0	*75.9*	99.5	*98.7*
	S	43.9	56.2	*17.9*	55.7	*65.5*	96.4	*98.6*
C	**A**	67.3	89.0	*61.9*	88.5	*78.5*	98.0	*98.2*
	P	85.6	97.2	*21.2*	96.4	*70.4*	98.9	*98.9*
	S	60.6	77.6	*46.9*	77.3	*71.2*	96.6	*98.8*
P	**A**	60.9	82.6	*87.9*	83.1	*92.0*	98.1	*99.1*
	C	24.8	80.5	*68.2*	80.6	*90.8*	99.1	*99.5*
	S	26.5	32.3	*13.4*	33.2	*90.9*	96.2	*98.2*
S	**A**	18.1	67.6	*42.6*	67.2	*74.7*	97.8	*96.7*
	C	32.6	83.8	*52.8*	83.9	*72.3*	99.0	*95.1*
	P	24.3	77.1	*17.9*	77.0	*72.6*	99.6	*96.7*
Avg.		49.9	77.2	*46.3*	77.1	*78.0*	98.1	*98.0*

In all experiments related to image synthesis, we synthesize 32 images per class—all together as one batch. The batch is initialized with real target samples carrying lowest self-entropy according to Eq. 5.7. We use $Adam$ with a learning rate of $1e-1$ (with cosine annealing schedule) for images optimization. We set $\lambda_{TV} = 1e-4$, $\lambda_{BN} = 1e-2$, and batch receives $10K$ updates. Sample images are provided in Fig. 5.3. The detailed ablation study is available in the Appendix.

5.4.3 Results

The reported results in Tab. 5.1-5.5 present average accuracy of 3 runs. To the best of our knowledge, we are the first to report single-source UDA results on PACS. So, in Tab. 5.1, we report results for all the possible pairs. Also, we report upper-bound performance (Baseline), however, we skip this information in rest of the tables as the numbers are often in saturation. In Tab. 5.2-5.5, the *Sc*→*Tg* row reports the amount of correct target pseudo-labels acquired using the frozen source model. The *PLR* row indicates the amount of correctly refined pseudo-labels (output of stage 1). We also report the performance loss on source due to source-free UDA (**SF-UDA (Sc)**) as well as the effectiveness of our method (**CSF-UDA (Sc)**) in retrieving the performance on source.

Table 5.2: Classification accuracy on DomainNet with ResNet101. Legend: ***C***: *Clipart*, ***I***: *Infograph*, ***P***: *Painting*, ***Q***: *Quickdraw*, ***R***: *Real*, *and* ***S***: *Sketch*.

Multi-Source UDA							
Tg	***C***	***I***	***P***	***Q***	***R***	***S***	Avg.
MM [111]	58.6	26.0	52.3	6.3	62.7	49.5	42.6
OML [141]	62.8	21.3	50.5	15.4	64.5	50.4	44.1
CMSS [142]	64.2	28.0	53.6	16.0	63.4	53.8	46.5
Sc→Tg	68.5	23.6	53.5	17.6	65.9	55.2	47.4
PLR (Tg)	**71.1**	**28.1**	**59.5**	**21.6**	**70.4**	**61.3**	**52.0**
SF-UDA (Tg)	**70.8**	27.2	**58.1**	**24.1**	**69.5**	**60.1**	**51.6**
SF-UDA (Sc)	*29.6*	*30.7*	*35.3*	*7.9*	*35.1*	*36.7*	*29.2*
CSF-UDA (Tg)	**71.4**	26.5	**57.1**	**24.2**	**67.9**	**59.0**	**51.0**
CSF-UDA (Sc)	***64.5***	***67.0***	***63.3***	***55.8***	***59.9***	***65.2***	***62.6***

Table 5.3: Classification accuracy on Visda-C with ResNet101.

Methods	***plane***	***bcycl***	***bus***	***car***	***horse***	***knife***	***mcycl***	***person***	***plant***	***skate***	***train***	***truck***	Avg.
Inferred	64.2	6.3	75.2	21.7	55.9	95.7	22.8	1.4	79.8	0.7	82.8	19.8	46.3
GPDA [149]	83.0	74.3	80.4	66.0	87.6	75.3	83.8	73.1	90.1	57.3	80.2	37.9	73.3
DADA [115]	92.9	74.2	82.5	65.0	90.9	**93.8**	87.2	74.2	89.9	71.5	86.5	48.7	79.8
SHOT [152]	94.3	**88.5**	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
A^2Net [153]	94.0	87.8	85.6	66.8	93.7	95.1	85.8	81.2	91.6	88.2	86.5	56.0	84.3
c→Tg	64.2	6.3	75.2	21.7	55.9	95.7	22.8	1.4	79.8	0.7	82.8	19.8	46.3
PLR (Tg)	**95.2**	64.8	**90.8**	**89.7**	87.4	93.7	**91.5**	**88.5**	56.4	82.9	**97.1**	**93.8**	**85.1**
SF-UDA (Tg)	**94.8**	68.1	**89.5**	**88.1**	86.5	90.4	**87.4**	**89.0**	53.2	81.5	**96.9**	**93.0**	**84.8**
SF-UDA (Sc)	*45.2*	*18.5*	*55.9*	*52.7*	*54.8*	*44.3*	*12.5*	*41.4*	*24.6*	*35.1*	*40.2*	*51.2*	*39.7*
CSF-UDA (Tg)	**94.9**	67.3	**89.2**	**87.8**	86.1	90.0	86.6	**88.7**	53.1	80.9	**96.5**	**94.6**	**84.6**
CSF-UDA (Sc)	***47.6***	***21.4***	***58.2***	***54.3***	***61.1***	***49.5***	***27.9***	***41.9***	***44.8***	***36.2***	***43.1***	***55.4***	***45.1***

Table 5.4: Classification accuracy on Digit5 with a naive 3-layer CNN. Legend: ***T***: *MNIST*, ***S***: *SVHN*, ***U***: *USPS*, ***M***: *MNIST-M, and* ***D***: *Synthetic-Digits.*

Single-Source UDA							**Multi-Source UDA**						
Sc	***T***	***T***	***T***	***S***	***U***	Avg.		***M,S, D,U***	***T,S, D,U***	***T,M, D,U***	***T,M, S,U***	***T,M, S,D***	Avg.
Tg	***U***	***S***	***M***	***T***	***T***			***T***	***M***	***S***	***D***	***U***	
SBA [139]	97.1	50.9	**98.4**	74.2	87.5	81.6	DCTN [138]	–	70.9	77.5	–	–	–
MALT [140]	97.0	**78.7**	71.4	98.7	20.7	73.3	MM [111]	98.4	72.8	81.3	89.5	96.1	87.6
MTDA [6]	94.2	52.0	85.5	84.6	91.5	81.5	OML [141]	98.7	71.7	84.8	91.1	97.8	88.8
GPLR [5]	89.3	63.4	94.3	97.3	91.8	87.5	CMSS [142]	99.0	75.3	88.4	**93.7**	97.7	90.8
Sc→Tg	86.3	34.7	63.8	67.0	70.2	64.4	Sc→Tg	98.6	69.1	52.0	40.3	88.7	69.8
PLR (Tg)	**99.1**	62.0	97.5	**99.2**	**99.2**	**91.4**	PLR (Tg)	**98.8**	**94.2**	**84.6**	87.8	**98.6**	**92.8**
SF-UDA (Tg)	**97.4**	61.6	95.4	**99.2**	**99.2**	**90.6**	SF-UDA (Tg)	**99.1**	**95.5**	**89.6**	90.0	**97.8**	**94.4**

Table 5.5: Classification accuracy on PACS with ResNet18. * results are taken from [6]. Legend: ***A***: *Art-Painting*, ***C***: *Cartoon*, ***P***: *Photo, and* ***S***: *Sketch.*

Multi-Target UDA								**Multi-Source UDA**					
Sc		***P***			***A***		Avg.	Sc	***C,P,S***	***A,P,S***	***A,C,S***	***A,C,P***	Avg.
Tg	***A***	***C***	***S***	***P***	***C***	***S***		Tg	***A***	***C***	***P***	***S***	
ADDA*	24.3	20.1	22.4	32.5	17.6	18.9	22.6	SIB [144]	88.9	89.0	98.3	82.2	89.6
DSN*	28.4	21.1	25.6	29.5	25.8	24.6	25.8	OML [141]	87.4	86.1	97.1	78.2	87.2
ITA*	31.4	23.0	28.2	35.7	27.0	28.9	29.0	RABN [145]	86.8	86.5	98.0	71.5	85.7
KD [146]	24.6	32.2	**33.8**	35.6	46.6	**57.5**	46.6	JiGen [147]	84.8	81.0	97.9	79.0	85.7
Sc→Tg		37.7			57.9		47.8	Sc→Tg	78.4	77.9	95.3	64.5	79.0
PLR (Tg)		**57.3**			**73.8**		**65.6**	PLR (Tg)	**89.3**	87.2	98.1	**83.2**	**89.5**
SF-UDA (Tg)	**80.1**	**76.1**	25.9	**96.0**	**82.8**	49.8	**68.4**	SF-UDA (Tg)	**90.8**	**89.5**	**98.8**	**85.2**	**91.1**
SF-UDA (Sc)		*56.5*			*47.8*		*52.2*	*SF-UDA (Sc)*	*69.1*	*62.2*	*34.1*	*35.5*	*50.2*
CSF-UDA (Tg)	**79.9**	**77.1**	25.1	**95.6**	**83.2**	47.6	**68.1**	CSF-UDA (Tg)	**89.5**	88.4	97.6	**84.6**	**90.0**
CSF-UDA (Sc)		***91.2***			***74.1***		***82.7***	*CSF-UDA (Sc)*	***81.9***	***78.5***	***68.9***	***66.1***	***73.8***

In Tab. 5.2, with comparable performance in one case, CSF-UDA consistently outperforms existing methods despite the large number of classes and discrepancy across domains. Also in Tab. 5.3, the proposed method achieves state-of-the-art average accuracy on such a challenging benchmark. In Tab. 5.4 *(left)*, we compare our method with the existing methods which address the challenging *MNIST→SVHN* and *MNIST→MNIST-M* tasks in a single-source UDA framework. In both cases, the source contains gray-scale images, and the target holds colored (RGB) images, carrying a massive distribution gap across domains for which our method achieves third and second-best performance, respectively. Nevertheless, by outperforming in 3 out of 5 cases, our method achieves state-of-the-art average accuracy. For multi-source UDA task in Tab. 5.4 *(right)*, the performance of proposed method is slightly affected while adapting to *Synthetic-Digits* benchmark. In the remaining 4 out of 5 cases, our method outperforms existing methods and achieves state-of-the-art average accuracy. Especially, the difference is substantial in the case of *MNIST-M*.

In Tab. 5.5 *(left)*, we compare our method with the existing approaches addressing multi-target UDA on PACS. As can be noticed, with comparable performance in 2 cases, our method achieves superior average accuracy. For multi-source UDA, we compare recent works in Tab. 5.5 *(right)*. Also in this framework, our method consistently outperforms existing methods, with only in one case getting lower, yet comparable, accuracy.

5.4.4 Discussion of Limitations

The proposed method is unquestionably limited for what concerns Stage 2 for small networks where BatchNorm (BN) layers are not prevalent. More in detail, we could not generate good source style images using standard source backbones (*i.e.,* a naive 3-layer CNN) for Digits5, while we verified that a deeper architecture with more BN layers (*e.g.,* ResNet18) could easily accomplish the task. However, we kept to the standard backbone CNN in order to be fair to previous methods. More details to be followed in Appendix.

5.5 Summary

This work proposes Continual Source-Free Unsupervised Domain Adaptation as a realistic adaptation scenario where source samples are not available, but the performance is to be preserved on the source domain. Our method is composed of two stages, a Source-Free UDA technique based on pseudo-label refinement, and a procedure for synthesizing source-style images to avoid catastrophic forgetting. The proposed pipeline effectively solves the task by only assuming a pre-trained source model. We empirically demonstrate that our proposed method achieves state-of-the-art performance on major UDA benchmarks.

5.6 Appendix

5.6.1 Performance Analysis of Negative Learning Against Different Noise Distributions

As discussed in the main text, Negative Learning (NL) is an indirect learning method in which a model is optimized to produce low confidence to a randomly chosen complementary label for the given input image. Such a method offers better performance in the case of a noisy label set. Nevertheless, the applicability of a naively applied NL method [3] is mostly limited to noise showing uniform distribution (*Symmetric-Noise*). To better highlight such limitation, figure 5.4 shows the confusion matrix for the *initial* noise along with the histograms of the noisy and clean sample's confidence distribution *obtained* after training.

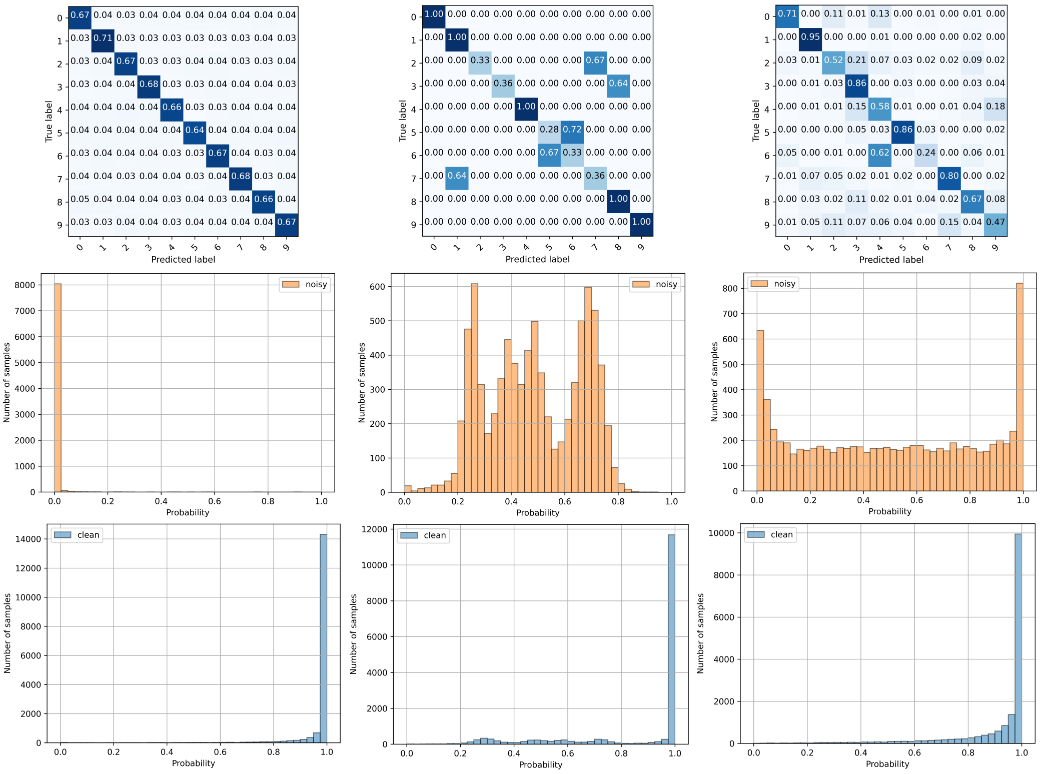
As shown in Fig. 5.4a, when labels are initially affected by *Symmetric-Noise*, noisy samples are classified with low confidence whereas the clean samples are leaned to high confidence. Consequently, effective noise separation is achieved by NL. For the *Asymmetric-Noise*, mimicking some of the structures of real errors [4] *i.e.,* for MNIST, mapping 2→7, 3→8, 7→1, and 5↔6, existing NL method's effectiveness degrades considerably. As can be seen in Fig. 5.4b, noisy samples are overfitted with quite high confidence (even higher than 50%). Yet, sub-optimal separation still exists if we consider samples carrying confidence higher than 90% only (though this is not generalizable for other benchmarks where all clean samples do not carry such high confidence).

Nevertheless, noisy samples are overfitted with very high confidence when *shift-noise* associated with inferred pseudo-labels [5] is considered (Fig. 5.4c). This happens because, in the shift-noise, inferred labels are affected by a noise skewed towards some of the classes *e.g.,* class 3 and 4 in Fig. 5.4c). So, instead of struggling with low confidence (the central idea of NL), noisy samples obtain higher confidence in relation to skewed (noisy) classes. Consequently, subsequent Positive Learning achieves sub-optimal performance with such noise distribution in the UDA framework.

Please note that in all cases, the *amount* of noise is same *i.e.,* 32.97%, similar to the amount of shift-noise observed in the case of SVHN → MNIST UDA task.

5.6.2 Online Adaptive Pseudo-Label Refinement

The proposed noise filtering approach based on Ensemble Learning using Disjoint Residual Labels is capable of disentangling noisy samples (affected by shift-noise) from the clean ones. However, a naive *two-step* approach *i.e.,* filtering noise followed by pseudo-label refinement (using a model trained on only high confidence samples) results in sub-optimal



(a) Symmetric-Noise (b) Asymmetric-Noise (c) Shift-Noise

Figure 5.4: Noise filtering capability of the existing Negative Learning method [3] over various noise distributions. Column (a) Symmetric Noise, column (b) Asymmetric artificial noise [4], column (c) Shift noise [5]. First row: the confusion matrix shows how the noise is distributed in the beginning. Second row: confidence prediction for the noisy samples after training with NL. Third row: confidence prediction for the clean samples after training with NL. The amount of initial noise is same in magnitude (*i,e.,* 32.97%) for all the cases.

performance as some noisy samples unavoidably attain higher confidence if they remain a part of training. On the contrary, superior performance is achieved when adaptively filtered noisy samples are progressively refined using the proposed online pseudo-label refinement approach (see Fig. 5.5).

In Fig. 4.4, the profiles related to the right y-axis (dashed lines) show the capability of the proposed threshold γ to adapt to different single-source UDA cases, while the profiles related to the left y-axis show the corresponding noise reduction achieved by Stage 1—our proposed Pseudo-Label Refinement (PLR) method. In particular, it is worth noting that no pseudo-label refinement is applied during the first few epochs until the ensemble gets a mature state of generalization capability. As the ensemble members get more and more confident in the clean examples, label reassignment becomes more rigorous preventing the network to overfit to noisy samples.

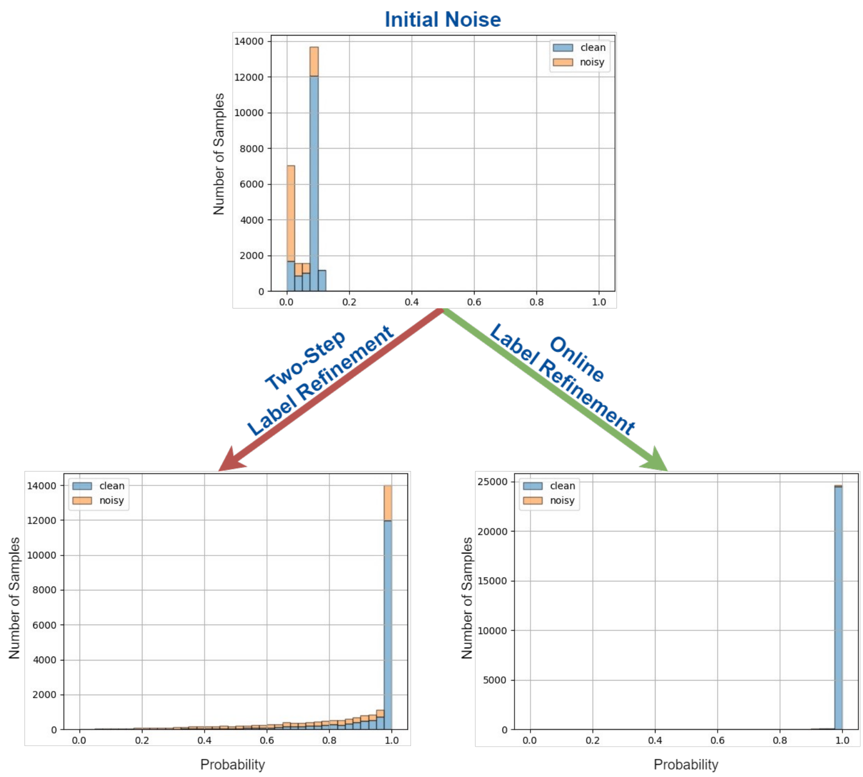


Figure 5.5: Performance comparison between *Two-Step* and Online pseudo-label refinement considering SVHN $\rightarrow$ MNIST UDA task.

5.6.3 Ablation Study for Source-Free UDA

We ablate the design choices on the SVHN$\rightarrow$MNIST adaptation task. It is important to note that the parameters estimated here are then used in all subsequent experiments, demonstrating the little sensitivity of the proposed method to such coefficients.

The adaptive property of the proposed is a function of α. We consider $N_e = 1$ (*i.e.,* no ensemble) to ablate the different values of α. Fig. 4.6a shows that $\alpha = 0.90$ results in the highest noise reduction, whereas $\alpha = 0.50$ is the least effective. These findings make sense, since with $\alpha = 0.50$, the reassignment would start too early (*i.e.,* as soon as some example has confidence greater than 0.5, since γ is always zero beforehand), when the network is not yet "ready" for it. Instead, with $\alpha = 0.90$, reassignment will start only when some samples have very high confidence. From that moment on, γ starts adapting, so that the more the samples with high confidence, the more permissive the threshold γ will be. The why reason $\alpha = 0.95$ slightly under-performs is to be ascribed to the fact that a relatively higher number of noisy samples are overfitted by the model before they could be reassigned.

Fixing $\alpha = 0.9$, we then ablate on the different number of members N_e in ensemble

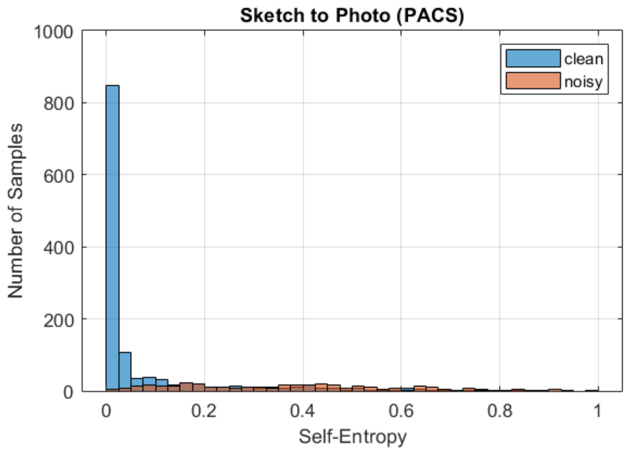


Figure 5.6: Self-Entropy of target samples corresponding to their clean/noisy refined pseudo-labels. As can be noticed, samples with low self entropy usually carry clean pseudo labels.

network. The results shown in Fig. 4.6c exhibit that a faster noise reduction is achieved with higher number of ensemble members; $N_e = 3$ can be regarded as the optimal choice considering performance *vs.* computational cost trade-off. Such improvement is a consequence of the strong consensus obtained from the ensemble network since $N_e - 1$ members always receive noiseless feedback, while only one member may be partially affected by wrong pseudo-labels. Further, results in Fig. 4.6d shows the ineffectiveness of using same complementary labels for all members (tough still randomly chosen). We show that using Disjoint Residual labels gives its best when combined with different data augmentations.

5.6.4 Target Samples Selection and Ablation Study For Image Synthesis

As shown in Fig. 5.3 (*left*), optimizing random *noise* into synthetic images lead to a distribution far different from natural (or original training) images. We empirically observe that a model trained on such synthetic images leads to unsatisfactory classification performance on the real source domain and thus do not help in preserving performance on the source.

On the contrary, we discovered that optimizing the *target-images* into synthetic images leads to a distribution identical to the real source domain. However, to preserve essential semantic information, chosen target-images should carry correct labels, *i.e.,* correctly refined pseudo labels form stage 1 of our pipeline. For this reason we compute prediction uncertainty of the target samples quantified by self-entropy where smaller entropy indicates more confident prediction (see Fig. 5.6). We sort the target samples in ascending order and select only the first N_h samples from each class to generate class-conditioned source-style images. Fig. 5.3 (*right*) shows the higher fidelity, realism, and most of all diversity in synthesized images obtained using our approach.

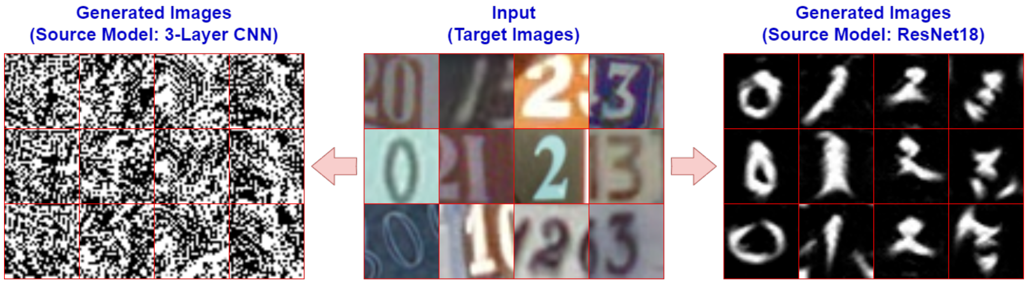


Figure 5.7: Generated synthetic source-style (MNIST) images using source-model as naive 3-layer CNN (*left*) and ResNet18 (*right*) starting from target images (SVHN).

Based on qualitative analysis of synthesized images for source domain *photo*, starting with target images from domain *sketch*, we found that $\lambda_{TV} > 1e-4$ introduces unrealistic patches while $\lambda_{TV} < 1e-4$ results insufficient contrast in synthesized images. The optimization gets unstable as we increase $\lambda_{BN} > 1e-2$ while decreasing $\lambda_{BN} < 1e-2$ prolongs the optimization process more than $10K$ updates are required. Therefore, for all the experiments, we set $\lambda_{TV} = 1e-4$ and $\lambda_{BN} = 1e-2$ which require each batch to be optimized for $10K$ iterations.

5.6.5 Limitation of Image Synthesis with Small-Scale Models

While synthesizing source-style images, we leverage the fact that CNNs store rich data statistics in Batch-Norm (BN) layers. This is exploited by means of a feature distribution regularizer which is heavily dependent on the information stored in the pre-trained source model's BN layers and helps in generating realistic class-conditioned source-style images. However, if a model is too simple or holds few BN layers (e.g only 3 BN layers in the naive CNN used for digit5 benchmarks), our method struggles in generating good quality source-style images (see Fig. 5.7 (*left*)). Nevertheless, it is to be noted that such images do minimize the classification loss for the target model *i.e.,* they are well classified despite their bad appearance.

To verify this, we train ResNet18 on MNIST (using similar image resolution of 32×32) as the source model and use it to optimize target images (SVHN). As can be noticed in Fig. 5.7 (*right*), synthesized images contain vital semantic information that we found handy in preserving good performance on source. Especially, it is worth noticing that despite the input target images carry varying colours and clutter, the synthesized output images contain crisp semantic details in black and white style. However, in order to be fair to previous methods, *we kept our evaluation to the standard backbone CNN (3-layer)* and reported it as limitation *i.e.,* our method does not synthesize good quality images employing a naive small-scale model.

Chapter 6

Conclusions

In this dissertation, we investigated the applicability of collaborative learning toward addressing heterogeneous computer vision problems such as model compression, action recognition, source-free unsupervised domain adaptation, and catastrophic forgetting associated with domain adaptation. From another perspective, our study is framed around the concept of machines-teaching machines. In proposed methods, either we employ privileged information, *i.e.,* having access to more knowledge from the teacher network for training than for testing, or we employ ensemble learning where models help each other to collaboratively attain better generalization.

Understanding the relationship between modalities and leveraging the complementarity of information is critical to address challenging real-world problems. With the proposed methods, we have shown that collaborative learning is a promising framework for developing better models. However, this dissertation focused on fully-supervised methods for model compression and action recognition. The future of feature learning is now moving towards a self-supervised framework, so there is an exciting opportunity to employ collaborative learning in such a realistic framework. Nevertheless, our later two studies already lie in the challenging unsupervised learning framework for the image classification task, which can be extended to video classification as well.

We hope this dissertation is a step forward in the direction of collaborative learning, and we would like to encourage the development of more methods in this field.

References

[1] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018.

[2] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[3] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 101–110, 2019.

[4] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

[5] Pietro Morerio, Riccardo Volpi, Ruggero Ragonesi, and Vittorio Murino. Generative pseudo-label refinement for unsupervised domain adaptation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3130–3139, 2020.

[6] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing*, 29:3993–4002, 2020.

[7] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[9] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[10] Suraj Srinivas, Ravi Kiran Sarvadevabhatla, Konda Reddy Mopuri, Nikita Prabhu, Srinivas SS Kruthiventi, and R Venkatesh Babu. A taxonomy of deep convolutional neural nets for computer vision. *Frontiers in Robotics and AI*, 2:36, 2016.

[11] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029, 2020.

[12] D. Lopez-Paz, B. Schölkopf, L. Bottou, and V. Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, November 2016.

[13] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, 2009. Advances in Neural Networks Research: IJCNN2009.

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

[15] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.

[16] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–8, 2019.

[17] W Ahmed, M Amjad, KN Junejo, T Mahmood, and AH Khan. Is the performance of a cricket team really unpredictable? a case study on pakistan team using machine learning. *Indian Journal of Science and Technology*, 13(34):3586–3599, 2020.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2410–2419, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.

[21] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5353–5360, 2015.

[22] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[23] Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*, 2018.

[24] Ariel Gordon, Elad Eban, Ofir Nachum, Bo Chen, Hao Wu, Tien-Ju Yang, and Edward Choi. Morphnet: Fast & simple resource-constrained structure learning of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1586–1595, 2018.

[25] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: similarity control and knowledge transfer. *Journal of machine learning research*, 16(2023-2049):2, 2015.

[26] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H Lampert. Learning to rank using privileged information. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 825–832, 2013.

[27] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019.

[28] Ji Wang, Weidong Bao, Lichao Sun, Xiaomin Zhu, Bokai Cao, and S Yu Philip. Private model compression via knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1190–1197, 2019.

[29] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. *arXiv preprint arXiv: 1802.05668*, 2018.

[30] Vasileios Belagiannis, Azade Farshad, and Fabio Galasso. Adversarial network compression. In *Proceedings of the European Conference on Computer Vision Wokshops (ECCVW)*, pages 0–0, 2018.

[31] Miguel A Carreira-Perpinán and Yerlan Idelbayev. “learning-compression” algorithms for neural net pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8532–8541, 2018.

[32] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11264–11272, 2019.

[33] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized

convolutional neural networks for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4820–4828, 2016.

[34] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[35] Sachin Mehta, Mohammad Rastegari, Linda Shapiro, and Hannaneh Hajishirzi. Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9190–9200, 2019.

[36] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

[37] Tien-Ju Yang, Andrew Howard, Bo Chen, Xiao Zhang, Alec Go, Mark Sandler, Vivienne Sze, and Hartwig Adam. Netadapt: Platform-aware neural network adaptation for mobile applications. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[38] Wei Wen, Feng Yan, and Hai Li. Autogrow: Automatic layer growing in deep convolutional networks. *arXiv preprint arXiv:1906.02909*, 2019.

[39] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[41] Neha Tiwari, Damodar Reddy Edla, Shubham Dodia, and Annushree Bablani. Brain computer interface: A comprehensive survey. *Biologically inspired cognitive architectures*, 2018.

[42] X. Hou, Y. Liu, O. Sourina, Y. R. E. Tan, L. Wang, and W. Mueller-Wittig. Eeg based stress monitoring. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 2015.

[43] Adam D Gerson, Lucas C Parra, and Paul Sajda. Cortically coupled computer vision for rapid image search. *IEEE Transactions on neural systems and rehabilitation engineering*, 14(2):174–179, 2006.

[44] Evelyn Jungnickel and Klaus Gramann. Mobile brain/body imaging (mobi) of physical interaction with dynamically moving objects. *Frontiers in human neuroscience*, 10:306, 2016.

[45] J. L. Pérez-Benítez, J. A. Pérez-Benítez, and J. H. Espina-Hernández. Development of a brain computer interface using multi-frequency visual stimulation and deep neural networks. In *International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pages 18–24, 2018.

[46] Ashish Kapoor, Desney Tan, Pradeep Shenoy, and Eric Horvitz. Complementary computing for visual tasks: Meshing computer vision with human visual processing. In *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–7, 2008.

[47] Ashish Kapoor, Pradeep Shenoy, and Desney Tan. Combining brain computer interfaces with vision for object categorization. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[48] Jason Omedes, Inaki Iturrate, Luis Montesano, and Javier Minguez. Using frequency-domain features for the generalization of eeg error-related potentials among different tasks. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5263–5266. IEEE, 2013.

[49] Pouya Bashivan, Irina Rish, Mohammed Yeasin, and Noel Codella. Learning representations from eeg with deep recurrent-convolutional neural networks. *Internetional Confernece on Learning Representations (ICLR)*, 2016.

[50] Sarwat Fatima and Awais M Kamboh. Decoding brain cognitive activity across subjects using multimodal m/eeg neuroimaging. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3224–3227. IEEE, 2017.

[51] C. Spampinato, S. Palazzo, I. Kavasidis, D. Giordano, N. Souly, and M. Shah. Deep learning human mind for automated visual classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[52] Mojtaba Khomami Abadi, Ramanathan Subramanian, Seyed Mostafa Kia, Paolo Avesani, Ioannis Patras, and Nicu Sebe. Decaf: Meg-based multimodal database for decoding affective physiological responses. *IEEE Transactions on Affective Computing*, 6(3):209–222, 2015.

[53] Britta U. Westner, Sarang S. Dalal, Simon Hanslmayr, and Tobias Staudigl. Across-

subjects classification of stimulus modality from human meg high frequency activity. *PLOS Computational Biology*, 14, 03 2018.

[54] S. Ehrlich, A. Wykowska, K. Ramirez-Amaro, and G. Cheng. When to engage in interaction — and how? eeg-based enhancement of robot's ability to sense social signals in hri. In *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014.

[55] Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

[56] Nuno Garcia, Pietro Morerio, and Vittorio Murino. Modality distillation with multiple stream networks for action recognition. *European Conference on Computer Vision*, 2018.

[57] Nuno Garcia, Pietro Morerio, and Vittorio Murino. Learning with privileged information via adversarial discriminative modality distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[58] Wiro J Niessen, Koen L Vincken, Joachim Weickert, BM Ter Haar Romeny, and Max A Viergever. Multiscale segmentation of three-dimensional mr brain images. *International Journal of Computer Vision*, 31(2-3):185–202, 1999.

[59] James V Haxby, M Ida Gobbini, Maura L Furey, Alumit Ishai, Jennifer L Schouten, and Pietro Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425–2430, 2001.

[60] John-Dylan Haynes and Geraint Rees. Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience*, 7(7):523–534, 2006.

[61] Thomas O'Donnell, Gareth Funka-Lea, Hüseyin Tek, Marie-Pierre Jolly, Matthias Rasch, and Randolph Setser. Comprehensive cardiovascular image analysis using mr and ct at siemens corporate research. *International Journal of Computer Vision*, 70(2):165–178, 2006.

[62] Shinji Nishimoto, An T Vu, Thomas Naselaris, Yuval Benjamini, Bin Yu, and Jack L Gallant. Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology*, 21(19):1641–1646, 2011.

[63] Andrei Barbu, Daniel P Barrett, Wei Chen, Narayanaswamy Siddharth, Caiming Xiong, Jason J Corso, Christiane D Fellbaum, Catherine Hanson, Stephen José Hanson, Sébastien Hélie, Evguenia Malaia, Barak A. Pearlmutter, Jeffrey Mark Siskind, Thomas Micheal Talavage, and Ronnie B. Wilbur. Seeing is worse than believing:

Reading people's minds better than computer-vision methods recognize actions. In *European Conference on Computer Vision*, pages 612–627. Springer, 2014.

[64] Junwei Han, Xiang Ji, Xintao Hu, Jungong Han, and Tianming Liu. Clustering and retrieval of video shots based on natural stimulus fmri. *Neurocomputing*, 144:128–137, 2014.

[65] Xintao Hu, Lei Guo, Junwei Han, and Tianming Liu. Decoding semantics categorization during natural viewing of video streams. *IEEE Transactions on Autonomous Mental Development*, 7(3):201–210, 2015.

[66] Junwei Han, Xiang Ji, Xintao Hu, Lei Guo, and Tianming Liu. Arousal recognition using audio-visual features and fmri-based brain response. *IEEE Transactions on Affective Computing*, 6(4):337–347, 2015.

[67] Mohammad Tavakolian and Abdenour Hadid. A spatiotemporal convolutional neural network for automatic pain intensity estimation from facial dynamics. *International Journal of Computer Vision*, 127(10):1413–1425, 2019.

[68] Pouya Ghaemmaghami, Mojtaba Khomami Abadi, Seyed Mostafa Kia, Paolo Avesani, and Nicu Sebe. Movie genre classification by exploiting meg brain signals. In *International Conference on Image Analysis and Processing*, pages 683–693. Springer, 2015.

[69] Halme Hanna-Leena and Parkkonen Lauri. Across-subject offline decoding of motor imagery from meg and eeg. *Scientific reports*, 8(1):10087, 2018.

[70] Bruce N Cuthbert, Harald T Schupp, Margaret M Bradley, Niels Birbaumer, and Peter J Lang. Brain potentials in affective picture processing: covariation with autonomic arousal and affective report. *Biological psychology*, 52(2):95–111, 2000.

[71] K. Takahashi and A. Tsukaguchi. Remarks on emotion recognition from bio-potential signals. In *2nd International conference on Autonomous Robots and Agents*, volume 3, pages 1148–1153, 2004.

[72] Guillaume Chanel, Joep JM Kierkels, Mohammad Soleymani, and Thierry Pun. Short-term emotion assessment in a recall paradigm. *International Journal of Human-Computer Studies*, 67(8):607–627, 2009.

[73] Sander Koelstra and Ioannis Patras. Fusion of facial expressions and eeg for implicit affective tagging. *Image and Vision Computing*, 31(2):164–174, 2013.

[74] Sidi Liu, Jinglei Lv, Yimin Hou, Ting Shoemaker, Qinglin Dong, Kaiming Li, and Tianming Liu. What makes a good movie trailer?: Interpretation from simultaneous

eeg and eyetracker recording. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 82–86. ACM, 2016.

[75] Xin Chai, Qisong Wang, Yongping Zhao, Xin Liu, Ou Bai, and Yongqiang Li. Unsupervised domain adaptation techniques based on auto-encoder for non-stationary eeg-based emotion recognition. *Computers in biology and medicine*, 79:205–214, 2016.

[76] Zhong Yin, Yongxiong Wang, Li Liu, Wei Zhang, and Jianhua Zhang. Cross-subject eeg feature selection for emotion recognition using transfer recursive feature elimination. *Frontiers in neurorobotics*, 11:19, 2017.

[77] Xiang Li, Dawei Song, Peng Zhang, Yazhou Zhang, Yuexian Hou, and Bin Hu. Exploring eeg features in cross-subject emotion recognition. *Frontiers in Neuroscience*, 12:162, 2018.

[78] Mohammad Soleymani, Sander Koelstra, Ioannis Patras, and Thierry Pun. Continuous emotion detection in response to music videos. In *Face and Gesture 2011*, pages 803–808. IEEE, 2011.

[79] Yachen Zhu, Shangfei Wang, and Qiang Ji. Emotion recognition from users' eeg signals with the help of stimulus videos. In *2014 IEEE international conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2014.

[80] Jia-Yi Zhu, Wei-Long Zheng, and Bao-Liang Lu. Cross-subject and cross-gender emotion classification from eeg. In *World Congress on Medical Physics and Biomedical Engineering, June 7-12, 2015, Toronto, Canada*, pages 1188–1191. Springer, 2015.

[81] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberye. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering*, 47(9):1185–1194, 2000.

[82] Mario Giovanni Terzano, Liborio Parrino, Adriano Sherieri, Ronald Chervin, Sudhansu Chokroverty, Christian Guilleminault, Max Hirshkowitz, Mark Mahowald, Harvey Moldofsky, Agostino Rosa, Robert Thomas, and Arthur Walters. Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (cap) in human sleep. *Sleep Medicine*, 2(6):537 – 553, 2001.

[83] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on Biomedical Engineering*, 51(6):1034–1043, 2004.

[84] B. Blankertz, K. Muller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlogl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schroder, and N. Bir-

baumer. The bci competition 2003: progress and perspectives in detection and discrimination of eeg single trials. *IEEE Transactions on Biomedical Engineering*, 51(6):1044–1051, 2004.

[85] Ralph G. Andrzejak, Kaspar Schindler, and Christian Rummel. Nonrandomness, nonlinear dependence, and nonstationarity of electroencephalographic recordings from epilepsy patients. *Phys. Rev. E*, 86:046206, Oct 2012.

[86] Sebastian Stober, Avital Sternin, Adrian M. Owen, and Jessica A. Grahn. Towards music imagery information retrieval: Introducing the openmiir dataset of eeg recordings from music perception and imagination. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[87] S. Zhao and F. Rudzicz. Classifying phonological categories in imagined and articulated speech. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[88] Arman Savran, Koray Ciftci, Guillaume Chanel, Javier Mota, Luong Hong Viet, Blent Sankur, Lale Akarun, Alice Caplier, and Michele Rombaut. Emotion detection in the loop from brain signals and facial images. In *Proceedings of the eNTERFACE 2006 Workshop*, 2006.

[89] Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. Deap: A database for emotion analysis; using physiological signals. *IEEE transactions on affective computing*, 3(1):18–31, 2012.

[90] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic. A multimodal database for affect recognition and implicit tagging. *IEEE Transactions on Affective Computing*, 3(1):42–55, 2012.

[91] Wei-Long Zheng and Bao-Liang Lu. Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development*, 7(3):162–175, 2015.

[92] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[93] Jonathan W Peirce. Psychopy—psychophysics software in python. *Journal of neuroscience methods*, 162(1-2):8–13, 2007.

[94] Scott A Huettel and Gregory McCarthy. What is odd in the oddball task?: Prefrontal cortex is activated by dynamic changes in response strategy. *Neuropsychologia*, 42(3):379–386, 2004.

[95] Robert Oostenveld, Pascal Fries, Eric Maris, and Jan-Mathijs Schoffelen. Fieldtrip: open source software for advanced analysis of meg, eeg, and invasive electrophysiological data. *Computational intelligence and neuroscience*, 2011:1, 2011.

[96] Joseph Dien. Issues in the application of the average reference: Review, critiques, and recommendations. *Behavior Research Methods, Instruments, & Computers*, 30(1):34–43, 1998.

[97] Li-Chen Shi, Ying-Ying Jiao, and Bao-Liang Lu. Differential entropy feature for eeg-based vigilance estimation. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6627–6630. IEEE, 2013.

[98] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[99] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. Lstm fully convolutional networks for time series classification. *IEEE Access*, 6:1662–1669, 2018.

[100] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[101] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, 2013.

[102] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456, 2015.

[103] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2001.

[104] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[105] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 826–834, 2016.

[106] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *arXiv preprint arXiv:1902.06162*, 2019.

[107] Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. A survey of deep learning and its applications: A new paradigm to machine learning. *Archives of Computational Methods in Engineering*, pages 1–22, 2019.

[108] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1521–1528, 2011.

[109] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 627–636, 2019.

[110] Safa Cicek and Stefano Soatto. Unsupervised domain adaptation via regularized conditional alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1416–1425, 2019.

[111] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[112] Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. Homm: Higher-order moment matching for unsupervised domain adaptation. In *AAAI*, 2020.

[113] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 2019.

[114] Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. Learning disentangled semantic representation for domain adaptation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2060–2066. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[115] Hui Tang and Kui Jia. Discriminative adversarial domain adaptation. In *AAAI*, pages 5940–5947, 2020.

[116] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci. Unsupervised domain adaptation using feature-whitening and con-

sensus loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9471–9480, 2019.

[117] Jun Wen, Nenggan Zheng, Junsong Yuan, Zhefeng Gong, and Changyou Chen. Bayesian uncertainty matching for unsupervised domain adaptation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3849–3855. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[118] Debasmit Das and C. S. George Lee. Graph matching and pseudo-label guided deep unsupervised domain adaptation. In Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 342–352, Cham, 2018. Springer International Publishing.

[119] Boris Chidlovskii, Stephane Clinchant, and Gabriela Csurka. Domain adaptation in the absence of source domain data. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 451–460, 2016.

[120] Arun Reddy Nelakurthi, Ross Maciejewski, and Jingrui He. Source free domain adaptation using an off-the-shelf classifier. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 140–145. IEEE, 2018.

[121] Jogendra Nath Kundu, Naveen Venkat, Rahul M V, and R. Venkatesh Babu. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[122] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

[123] Yixiao Ge, Dapeng Chen, and Hongsheng Li. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *International Conference on Learning Representations*, 2020.

[124] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[125] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, pages 1–18, 2020.

[126] Xu-Cheng Yin, Kaizhu Huang, Hong-Wei Hao, Khalid Iqbal, and Zhi-Bin Wang.

A novel classifier ensemble method with sparsity and diversity. *Neurocomputing*, 134:214–221, 2014.

[127] Nuno C. Garcia, Sarah Adel Bargal, Vitaly Ablavsky, Pietro Morerio, Vittorio Murino, and Stan Sclaroff. DMCL: distillation multiple choice learning for multimodal action recognition. *CoRR*, abs/1912.10982, 2019.

[128] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537, 2018.

[129] Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9294–9303, 2020.

[130] Bruno Klaus de Aquino Afonso and Lilian Berton. Identifying noisy labels with a transductive semi-supervised leave-one-out filter. *Pattern Recognition Letters*, 2020.

[131] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[132] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *IEEE*, 2011.

[133] John S Denker, WR Gardner, Hans Peter Graf, Donnie Henderson, Richard E Howard, W Hubbard, Lawrence D Jackel, Henry S Baird, and Isabelle Guyon. Neural network recognizer for hand-written zip code digits. In *Advances in neural information processing systems*, pages 323–331, 1989.

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

[135] 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.

[136] Xingchao Peng, Ben Usman, Neela Kaushik, Dequan Wang, Judy Hoffman, and Kate Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2021–2026, 2018.

[137] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 2988–2997. PMLR, 2017.

[138] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3964–3973, 2018.

[139] Paolo Russo, Fabio M Carlucci, Tatiana Tommasi, and Barbara Caputo. From source to target and back: symmetric bi-directional adaptive gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8099–8108, 2018.

[140] Breton Minnehan and Andreas Savakis. Deep domain adaptation with manifold aligned label transfer. *Machine Vision and Applications*, 30(3):473–485, 2019.

[141] Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 382–403, Cham, 2020. Springer International Publishing.

[142] Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[143] Massimiliano Mancini, Lorenzo Porzi, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Boosting domain adaptation by discovering latent domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3771–3780, 2018.

[144] Shell Xu Hu, Pablo Garcia Moreno, Yang Xiao, Xi Shen, Guillaume Obozinski, Neil Lawrence, and Andreas Damianou. Empirical bayes transductive meta-learning with synthetic gradients. In *International Conference on Learning Representations*, 2020.

[145] Jiaolong Xu, Liang Xiao, and Antonio M López. Self-supervised domain adaptation for computer vision tasks. *IEEE Access*, 7:156694–156706, 2019.

[146] Atif Belal, Madhu Kiran, Jose Dolz, Louis-Antoine Blais-Morin, Eric Granger, et al. Knowledge distillation methods for efficient unsupervised adaptation across multiple domains. *Image and Vision Computing*, page 104096, 2021.

[147] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pages 2229–2238, 2019.

[148] 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*, pages 3723–3732, 2018.

[149] Minyoung Kim, Pritish Sahu, Behnam Gholami, and Vladimir Pavlovic. Unsupervised visual domain adaptation: A deep max-margin gaussian process approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4380–4390, 2019.

[150] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1426–1435, 2019.

[151] Shaokai Ye, Kailu Wu, Mu Zhou, Yunfei Yang, Sia Huat Tan, Kaidi Xu, Jiebo Song, Chenglong Bao, and Kaisheng Ma. Light-weight calibrator: a separable component for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13736–13745, 2020.

[152] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6028–6039. PMLR, 13–18 Jul 2020.

[153] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9010–9019, October 2021.

[154] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8227, 2021.

[155] Yujun Shi, Li Yuan, Yunpeng Chen, and Jiashi Feng. Continual learning via bit-level information preserving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16674–16683, 2021.

[156] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6568–6577, 2019.

[157] Andreea Bobu, Eric Tzeng, Judy Hoffman, and Trevor Darrell. Adapting to continuously shifting domains. In *ICLR*, 2018.

[158] Peng Su, Shixiang Tang, Peng Gao, Di Qiu, Ni Zhao, and Xiaogang Wang. Gradient regularized contrastive learning for continual domain adaptation. In *AAAI*, 2021.

[159] Riccardo Volpi, Diane Larlus, and Gregory Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4443–4453, June 2021.

[160] Jogendra Nath Kundu, Rahul Mysore Venkatesh, Naveen Venkat, Ambareesh Revanur, and R Venkatesh Babu. Class-incremental domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 53–69. Springer, 2020.

[161] 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*, pages 8978–8987, 2021.

[162] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3012–3021, 2020.

[163] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[164] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. *IEEE*, 2015.

[165] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[166] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

[167] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 5339–5349, 2018.

[168] Youngeun Kim, Donghyeon Cho, Kyeongtak Han, Priyadarshini Panda, and Sungeun Hong. Domain adaptation without source data. *IEEE Transactions on Artificial Intelligence*, pages 1–1, 2021.

[169] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.