SENTRY: Selective Entropy Optimization via Committee Consistency for Unsupervised Domain Adaptation

Viraj Prabhu Shivam Khare Deeksha Kartik Judy Hoffman
Georgia Institute of Technology
{virajp,skhare31,dkartik3,judy}@gatech.edu

Abstract

Many existing approaches for unsupervised domain adaptation (UDA) focus on adapting under only data distribution shift and offer limited success under additional cross-domain label distribution shift. Recent work based on self-training using target pseudolabels has shown promise, but on challenging shifts pseudolabels may be highly unreliable and using them for self-training may lead to error accumulation and domain misalignment. We propose Selective Entropy Optimization via Committee Consistency (SENTRY), a UDA algorithm that judges the reliability of a target instance based on its predictive consistency under a committee of random image transformations. Our algorithm then selectively minimizes predictive entropy to increase confidence on highly consistent target instances, while maximizing predictive entropy to reduce confidence on highly inconsistent ones. In combination with pseudolabel-based approximate target class balancing, our approach leads to significant improvements over the state-of-the-art on 27/31 domain shifts from standard UDA benchmarks as well as benchmarks designed to stress-test adaptation under label distribution shift. Our code is available at https://github.com/virajprabhu/SENTRY.

1. Introduction

Unsupervised domain adaptation (UDA) learns to transfer a predictive model from a labeled source domain to an unlabeled target domain. The particular instantiation of learning under covariate shift has been extensively studied within the computer vision community [13, 18, 25, 34, 44, 45]. However, many modern UDA methods, such as distribution matching based techniques, implicitly assume that the task label distribution does not change across domains, i.e. \( P_S(y) = P_T(y) \). When such an assumption is violated, distribution matching is not expected to succeed [22, 49].

In many real-world adaptation scenarios, one may encounter data distribution (i.e. covariate) shift across domains together with label distribution shift (LDS). For instance, a source dataset can be curated to have a balanced label distribution while a naturally arising target dataset may follow a power law label distribution, as some categories naturally occur more often than others (e.g. DomainNet [31], LVIS [16], and MSCOCO [23]). In order to make domain adaptation broadly applicable, it is critical to develop UDA algorithms that can operate under joint data and label distribution shift.

Recent works have attempted to address the problem of joint data and label distribution shift [22, 43], but these approaches can be unstable as they rely on self-training using often noisy pseudo-labels or conditional entropy minimization [22] over potentially miscalibrated predictions [15, 39]. Thus, when learning with unconstrained self-training, early mistakes can result in error accumulation [6] and significant domain misalignment (see Figure 1, top).

Figure 1: Top: Conventional entropy-minimization based approaches for unsupervised domain adaptation (UDA) operate by increasing model confidence on unlabeled target instances. Under strong distribution shifts, some instances may initially be misaligned and entropy minimization can lead to error accumulation. Bottom: We propose Selective Entropy Optimization via Committee Consistency (SENTRY), a UDA algorithm that i) identifies reliable target instances based on their predictive consistency under a set of random image transformations, and ii) selectively optimizes model entropy on these instances to induce domain alignment.
To address the problem of error accumulation arising from unconstrained self-training, we propose Selective Entropy Optimization via Committee Consistency (SENTRY), a novel selective self-training algorithm for UDA. First, rather than using model confidence which can be miscalibrated under a domain shift [39], we identify reliable target instances for self-training based on their predictive consistency under a committee of random, label-preserving image transformations. Such consistency checks have been found to be a reliable way to detect model errors [2]. Having identified reliable and unreliable target instances, we then perform selective entropy optimization: we consider a highly consistent target instance as likely correctly aligned, and increase model confidence by minimizing predictive entropy for such an instance. Similarly, we consider an instance with high predictive inconsistency over transformations as likely misaligned, and reduce model confidence by maximizing predictive entropy. See Figure 1 (bottom).

Contributions. We propose SENTRY, an algorithm for unsupervised adaptation under simultaneous data and label distribution shift. We make the following contributions:

1. A novel selection criterion that identifies reliable target instances for self-training based on predictive consistency over a committee of random, label-preserving image transformations.
2. A selective entropy optimization objective that minimizes predictive entropy (increasing confidence) on highly consistent target instances, and maximizes it (reducing confidence) on highly inconsistent ones.
3. We propose using class-balanced sampling on the source (using labels) and target (using pseudolabels), and find it to complement adaptation under LDS.
4. SENTRY sets a new state-of-the-art on 27/31 domain shifts belonging to both standard and LDS versions of several DA benchmarks for classification, including DomainNet [31], OfficeHome [46], and VisDA [32].

2. Related Work

Unsupervised Domain Adaptation (UDA). The task of transferring models from a labeled source to an unlabeled target domain has seen considerable progress [13, 18, 34, 45]. Many approaches align feature spaces via directly minimizing domain discrepancy statistics [20, 25, 45]. Recently, distribution matching (DM) via domain-adversarial learning has become a prominent UDA paradigm [13, 26, 36, 44, 53]. Such DM-based methods however achieve limited success in the presence of additional label distribution shift (LDS).

Some prior work has studied the problem of UDA under LDS, proposing class-weighted domain discrepancy measures [47, 51], generative approaches for pair-wise feature matching [42], or asymmetrically-relaxed distribution alignment [52]. Some prior work in UDA under LDS additionally assumes that the conditional input distribution does not change across domains i.e. \( p_S(y) \neq p_T(y), p_S(x|y) = p_T(x|y) \) (referred to as “label shift” [1, 24, 41]). We tackle the problem of UDA under simultaneous covariate and label distribution shift, without making additional assumptions.

Self-training for UDA. Recently, training on model predictions or self-training has proved to be a promising approach for UDA under LDS [22, 43]. This typically involves supervised training on confidently predicted target pseudolabels [43], confidence regularization [54], or conditional entropy minimization [14] on target instances [22]. However, unconstrained self-training can lead to error accumulation. To address this, we propose a selective self-training strategy that first identifies reliable instances for self-training and selectively optimizes model entropy on those.

Predictive Consistency. Predictive consistency under augmentations has been found to be useful in several capacities – as a regularizer in supervised learning [10], self-supervised representation learning [7], semi-supervised learning [3, 38, 40, 50], and UDA [22]. Bahat et al. [2] find consistency under image transformations to be a reliable indicator of model errors. Unlike prior work which optimizes for invariance across augmentations, we use predictive consistency under a committee of random image transforms to detect reliable instances for alignment, and selectively optimize model entropy on such instances.

3. Approach

We address the problem of unsupervised domain adaptation (UDA) of a model trained on a labeled source domain to an unlabeled target domain. In addition to covariate shift across domains, we focus on the practical scenario of additional cross-domain label distribution shift (LDS), and present a selective self-training algorithm for UDA that leads to reliable domain alignment in such a setting.

3.1. Notation

Let \( \mathcal{X} \) and \( \mathcal{Y} \) denote input and output spaces, with the goal being to learn a CNN mapping \( h : \mathcal{X} \rightarrow \mathcal{Y} \) parameterized by \( \Theta \). In unsupervised DA we are given access to labeled source instances \( (x_S, y_S) \sim P_S(\mathcal{X}, \mathcal{Y}) \), and unlabeled target instances \( x_T \sim P_T(\mathcal{X}) \), where \( S \) and \( T \) correspond to source and target domains. We consider DA in the context of C-way image classification: the inputs \( x \) are images, and labels \( y \) are categorical variables \( y \in \{1, 2, \ldots, C\} \). For an instance \( x \), let \( p_\Theta(y|x) \) denote the final probabilistic output from the model. For each target instance \( x_T \sim P_T(\mathcal{X}) \), we estimate a pseudolabel \( \hat{y} = \arg\max_{y} p_\Theta(y|x_T) \).

3.2. Preliminaries: UDA via entropy minimization

Unsupervised domain adaptation typically follows a two-stage training pipeline: source training, followed by target adaptation. In the first stage, a model is trained on the labeled source domain in a supervised fashion, minimizing a cross-
entropy loss with respect to ground truth labels.

$$L_{CE} = \mathbb{E}_{(x_S, y_S) \sim P_S}[L_{CE}(h(x_S), y_S)]$$ \hspace{1cm} (1)

In the second stage, the trained source model is adapted to the target with the use of unlabeled target and labeled source data. Recently, self-training via conditional entropy minimization (CEM) [14] has been shown to lead to strong performance for domain adaptation [35]. This approach optimizes model parameters to minimize conditional entropy on unlabeled target data $H_\theta(y|x)$. The entropy minimization objective $L_{ENT}$ is given by:

$$L_{ENT} = \mathbb{E}_{x_T \sim P_T}[H_\theta(y|x_T)]$$

$$= \mathbb{E}_{x_T \sim P_T} \left[ \sum_{c=1}^{C} -p_\theta(y=c|x_T) \log p_\theta(y=c|x_T) \right]$$ \hspace{1cm} (2)

However, in many real-world scenarios, in addition to covariate shift, label distributions across domains might also shift. Further, there might also be significant label imbalance within the target domain. In such cases, naive CEM has been found to potentially encourage trivial solutions of only predicting the majority class [22]. Li et al. [22] regularize CEM with an “information-entropy” objective $L_{IE}$ that encourages the model to make diverse predictions over unlabeled target instances. This is achieved by computing a distribution over classes predicted by the model for the last-$Q$ instances, denoted by $q(\hat{y})$, and updating parameters to maximize entropy over these predictions. This method is shown to help with domain alignment in the presence of label-distribution shift [22]. $L_{IE}$ is defined as:

$$L_{IE} = \mathbb{E}_{x_T \sim P_T} \left[ \sum_{c=1}^{C} p_\theta(y=c|x_T) \log q(\hat{y}=c) \right]$$ \hspace{1cm} (3)

CEM and error accumulation. While conditional entropy minimization has been a part of many successful approaches for semi-supervised learning [3, 14], few-shot learning [12], and more recently, UDA [22, 35], it suffers from a key challenge in the case of domain adaptation. Intuitively, conditional entropy minimization encourages the model to make confident predictions on unlabeled target data. This makes its success highly dependent on its initialization. Under a good initialization, categories may be reasonably aligned across source and target domains after source training, and such self-training works well. However, under strong domain shifts, several categories may initially be misaligned across domains, often systematically so, and entropy minimization will only lead to reinforcing such errors.

3.3. SENTRY: Selective Entropy Optimization via Committee Consistency

Predictive consistency-based selection. To address the problem of error accumulation under CEM, we propose selective optimization on well-aligned instances. The question then becomes: how can we identify reliable instances? One possibility is to use top-1 softmax confidence (or alternatively, predictive entropy), and only self-train on highly confident instances, as done in prior work [43]. However, under a distribution shift, such confidence measures tend to be miscalibrated and are often unreliable [39]. Instead, we propose using predictive consistency under a committee of label-preserving image transformations as a more robust measure for instance selection.

For a target instance $x_T \sim P_T$, we generate a committee of $k$ transformed versions $\{a_1(x_T), a_2(x_T), ..., a_k(x_T)\}$. We make predictions for each of these $k$ transformed versions, and measure consistency between the model’s prediction for the original image and for each of its $k$ augmented versions. In practice, we use a simple majority voting scheme: if the model’s prediction for a majority of
augmented versions matches its prediction on the original image, we consider the instance as “consistent”. Similarly, if the prediction for a majority of augmented versions does not match the original prediction, we mark it as “inconsistent”.

Selective Entropy Optimization. Having identified consistent and inconsistent instances, we perform Selective Entropy Optimization (SENTRY). First, for an instance marked as consistent, we increase model confidence by minimizing predictive entropy [14] with respect to one of its consistent augmented versions.

As described previously, some target instances may be misaligned under a domain shift. Entropy minimization on such instances would increase model confidence, reinforcing such errors. Instead, having identified such an instance via predictive inconsistency, we reduce model confidence by maximizing predictive entropy [33] with respect to one of its inconsistent augmented versions. While the former encourages confident predictions on highly consistent instances, the latter reduces model confidence on highly inconsistent and likely misaligned instances. In Sec. 4.6, we provide further intuition into the behavior of entropy maximization by illustrating its similarity to a binary cross-entropy loss with respect to the ground truth label for an incorrectly classified example in the binary classification case.

Without loss of generality, we minimize/maximize entropy with respect to the last consistent/inconsistent transformed version in our experiments. Our selective entropy optimization objective $L_{SENTRY}$ is given by:

$$L_{SENTRY}(x_T) = \begin{cases} +\mathcal{H}_\Theta(y|a_i(x_T)) & \text{if consistent} \\ -\mathcal{H}_\Theta(y|a_j(x_T)) & \text{if inconsistent} \end{cases}$$

(4)

Here $i$ and $j$ denote the index of the last consistent and inconsistent transformed version, respectively.

Such an approach may raise two concerns: First, that entropy minimization only on consistent instances might lead to the exclusion of a large percentage of target instances. Second, that indefinite entropy maximization on inconsistent target instances might prove detrimental to learning. Both of these concerns are addressed via the augmentation invariance regularizer built into our objective, which leads to an adaptive selection strategy that we now discuss.

Adaptive selection via augmentation invariance regularization. For instances marked as consistent, our approach minimizes entropy with respect to its last consistent augmented version rather than with respect to the original image itself. This yields two benefits: First, this builds data augmentation into the entropy minimization objective, which helps reduce overfitting. Second, it encourages invariance to the same set of augmentations that is used for selecting instances. We find that this makes our selection strategy adaptive, wherein an increasing percentage of target instances are selected for entropy minimization over the course of training, and consequently a decreasing percentage of target instances are selected for entropy maximization.

3.4. Overcoming LDS via pseudo class balancing

Under LDS, methods often have to adapt in the presence of severe label imbalance. While label imbalance on the source domain often leads to poor performance on tail classes [11, 48], adapting to an imbalanced target often results in poor performance on head classes [22, 49]. To overcome this, we employ a simple class-balanced sampling strategy. On the source domain, we perform class-balanced sampling using ground truth labels. On the target domain, we approximate the label distribution via pseudolabels, and perform approximate class-balanced sampling [55].

Such balancing also complements the target information-entropy loss $L_{IE}$ [22] (Eq. 3). To recap, $L_{IE}$ encourages a uniform distribution over predictions. Under severe label imbalance, it is possible to sample highly label-imbalanced batches (with most instances belonging to head classes) and so encouraging a uniform distribution over predictions can adversely affect learning. However, our class-balanced sampling strategy reduces the probability of such a scenario, and we find that it consistently improves performance.

Algorithm 1 details our full approach. The complete objective we optimize is given by:

$$\arg\min_{\Theta} \mathbb{E}_{(x_S,y_S)\sim P_S} \mathcal{L}_C + \mathbb{E}_{x_T\sim p_T \lambda_{IE} L_{IE} + \lambda_{SENTRY} L_{SENTRY}}$$

(5)

where the $\lambda$’s denote loss weights, and $p_{bal}$ and $p_{\text{bal}}$ denote balanced and pseudo class-balanced sampling.
4. Experiments

We first describe our experimental setup: datasets and metrics (Sec. 4.1), implementation details (Sec. 4.2), and baselines (Sec. 4.3). We then present our results (Sec. 4.4), ablation studies (Sec. 4.5), and analyze our approach (Sec. 4.6).

4.1. Datasets and Metrics

We report results on a mix of standard UDA benchmarks and specialized benchmarks designed to stress-test UDA methods under label distribution shift.

**DomainNet.** DomainNet [31] is a large UDA benchmark for image classification, containing 0.6 million images belonging to 6 domains spanning 345 categories. Due to labeling noise prevalent in its full version, we instead use the subset proposed in Tan et al. [43], which uses 40-commonly seen classes from 4 domains: Real (R), Clipart (C), Painting (P), and Sketch (S). As seen in Fig. 3 (left), there exists a natural label distribution shift across domains, which makes it suitable for testing our method without manual subsampling.

**OfficeHome.** OfficeHome [46] is an image classification-based benchmark containing 65 categories of objects found in office and home environments, spanning 4 domains: Real-world (Rw), Clipart (Cl), Product (Pr), and Art (Ar). We report performance on two versions: i) standard: the original dataset proposed in Venkateswara et al. [46], and ii) RS-UT: The Reverse-unbalanced Source (RS) and Unbalanced-Target (UT) version from Tan et al. [43], wherein source and target label distributions are manually long-tailed to be reversed versions of one another (see Fig. 3 (right)).

**VisDA.** VisDA2017 [32] is a large dataset for synthetic→real adaptation with 12 classes and >200K images.

**DIGITTS.** We use the SVHN [29]→MNIST [21] shift for 10-way digit recognition.

**Metric.** On LDS DA benchmarks (DomainNet and OfficeHome RS-UT), consistent with prior work in UDA under LDS [19, 43], we compute a mean of per-class accuracy on the target test split as our metric, that weights performance on all classes equally. On standard DA benchmarks (OfficeHome and VisDA2017) we report standard accuracy.

4.2. Implementation details

We use PyTorch [30] for all experiments. On DomainNet, OfficeHome, and VisDA2017, we modify the standard ResNet50 [17] CNN architecture to a few-shot variant used in recent DA work [8, 35, 43]: we replace the last linear layer with a C− way (for C classes) fully-connected layer with Xavier-initialized weights and no bias. We then L2-normalize activations flowing into this layer and feed its output to a softmax layer with a temperature $T = 0.05$. We match optimization details to Tan et al. [43]. On DIGITS, we make similar modifications to the LeNet architecture and use $T = 0.01$ [18]. For augmenting images for consistency checking, we use RandAugment [10], which sequentially applies $N$ label-preserving image transformations randomly sampled from a set of 14 transforms. We set $N = 3$, use transformation severity $M = 2.0$, and use a committee of $k = 3$ transforms. We use class-balanced sampling on the source domain and pseudo class-balanced sampling on the target. We set $\lambda_{IE}$ and $\lambda_{SENTRY}$ to 0.1 and 1.0, and match InstaPBM to set $Q=256$ for the information entropy loss.

4.3. Baselines

As our primary baselines we use four state-of-the-art UDA methods from prior work specifically designed for DA under LDS: i) **COAL [43]:** Co-aligns feature and label distributions, using prototype-based conditional alignment via MME [35], and self-training on confidently-predicted pseudo-labels. ii) **MDD + Implicit Alignment (LA) [19]:** Uses target pseudolabels to construct $N$−way (# classes per-batch) $K$−shot (# examples per class) dataloaders that are “aligned” (i.e. sample the same set of classes within a batch for both source and target), in conjunction with Margin Disparity Discrepancy [53], a strong UDA method. iii) **InstaPBM [22]:** Proposes “predictive-behavior” matching, which entails matching properties of $p_0(y|x)$ between source and target. This is achieved via optimizing a combination of mutual information maximization, contrastive, and mixup losses, and iv) **F-DANN [49]:** Proposes an asymmetrically-relaxed distribution matching-based version of DANN [13] to deal with LDS. COAL, InstaPBM, and
For completeness, we also include results for additional baselines from Tan et al. [43]: i) Conventional feature alignment-based UDA methods: DAN [25], JAN [28], DANN [13], MCD [37], and MDD [53], ii) BBSE [22] which only aligns label distributions, iii) Methods that assume non-overlapping labeling spaces: PADA [4], ETN [5], and UAN [52]. We also report results for FixMatch [40], a state-of-the-art self-training method for semi-supervised learning, on two benchmarks.

4.4. Results

Results on label-shifted DA benchmarks. We present results on 12 shifts from DomainNet (Table 1) and 6 shifts from OfficeHome RS→UT (Table 2). On DomainNet, SENTRY outperforms the next best performing method InstaPBM [22] on every shift, and by 3.55% mean accuracy averaged across shifts. On OfficeHome RS-UT, SENTRY outperforms the next best performing method MDD+I.A [19] on 5 out of 6 shifts, and on average by 3.58% mean accuracy. Our method also significantly outperforms F-DANN [49] (by 11.87% and 11.37%) and COAL [43] (by 5.50% and 6.85%), which are both UDA strategies designed for adaptation under LSD.

Table 1: Per-class average accuracies on DomainNet. Bold and underscore denote the best and second-best performing methods respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>R → C</th>
<th>R → P</th>
<th>R → S</th>
<th>C → R</th>
<th>C → P</th>
<th>S → R</th>
<th>P → S</th>
<th>P → R</th>
<th>P → C</th>
<th>C → S</th>
<th>S → C</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>65.75</td>
<td>68.84</td>
<td>59.15</td>
<td>77.71</td>
<td>60.60</td>
<td>57.87</td>
<td>84.45</td>
<td>62.35</td>
<td>65.07</td>
<td>77.10</td>
<td>63.00</td>
<td>59.72</td>
</tr>
<tr>
<td>BBSE [24]</td>
<td>55.38</td>
<td>63.62</td>
<td>47.44</td>
<td>64.58</td>
<td>42.18</td>
<td>42.36</td>
<td>81.55</td>
<td>49.04</td>
<td>54.10</td>
<td>68.54</td>
<td>48.19</td>
<td>46.07</td>
</tr>
<tr>
<td>PADA [4]</td>
<td>65.91</td>
<td>67.13</td>
<td>58.43</td>
<td>74.69</td>
<td>53.09</td>
<td>52.86</td>
<td>79.84</td>
<td>59.33</td>
<td>57.87</td>
<td>76.52</td>
<td>66.97</td>
<td>61.08</td>
</tr>
<tr>
<td>MCD [37]</td>
<td>61.97</td>
<td>69.33</td>
<td>56.26</td>
<td>79.78</td>
<td>56.61</td>
<td>53.66</td>
<td>83.38</td>
<td>58.31</td>
<td>60.98</td>
<td>81.74</td>
<td>56.27</td>
<td>66.78</td>
</tr>
<tr>
<td>DAN [25]</td>
<td>64.36</td>
<td>70.65</td>
<td>58.44</td>
<td>79.44</td>
<td>56.78</td>
<td>60.05</td>
<td>84.56</td>
<td>61.62</td>
<td>62.21</td>
<td>79.69</td>
<td>65.01</td>
<td>62.04</td>
</tr>
<tr>
<td>F-DANN [49]</td>
<td>66.15</td>
<td>71.80</td>
<td>61.53</td>
<td>81.85</td>
<td>60.06</td>
<td>61.22</td>
<td>84.46</td>
<td>66.81</td>
<td>62.84</td>
<td>81.38</td>
<td>69.62</td>
<td>66.50</td>
</tr>
<tr>
<td>UAN [52]</td>
<td>71.10</td>
<td>69.80</td>
<td>67.10</td>
<td>83.15</td>
<td>63.30</td>
<td>64.66</td>
<td>83.95</td>
<td>65.35</td>
<td>67.06</td>
<td>82.22</td>
<td>70.64</td>
<td>68.09</td>
</tr>
<tr>
<td>JAN [28]</td>
<td>65.57</td>
<td>73.58</td>
<td>67.61</td>
<td>85.02</td>
<td>64.96</td>
<td>67.17</td>
<td>87.06</td>
<td>67.92</td>
<td>66.10</td>
<td>84.54</td>
<td>72.77</td>
<td>67.51</td>
</tr>
<tr>
<td>ETN [5]</td>
<td>69.22</td>
<td>72.14</td>
<td>63.63</td>
<td>86.54</td>
<td>65.33</td>
<td>63.34</td>
<td>85.04</td>
<td>65.69</td>
<td>68.78</td>
<td>84.93</td>
<td>72.17</td>
<td>68.99</td>
</tr>
<tr>
<td>BSP [9]</td>
<td>67.29</td>
<td>73.47</td>
<td>69.31</td>
<td>86.50</td>
<td>67.52</td>
<td>70.90</td>
<td>86.83</td>
<td>70.33</td>
<td>68.75</td>
<td>84.34</td>
<td>72.40</td>
<td>71.47</td>
</tr>
<tr>
<td>DANN [13]</td>
<td>63.37</td>
<td>73.56</td>
<td>72.63</td>
<td>86.47</td>
<td>65.73</td>
<td>70.58</td>
<td>86.94</td>
<td>73.19</td>
<td>70.15</td>
<td>85.73</td>
<td>75.16</td>
<td>70.04</td>
</tr>
<tr>
<td>COAL [43]</td>
<td>73.85</td>
<td>75.37</td>
<td>70.50</td>
<td>89.63</td>
<td>69.98</td>
<td>71.29</td>
<td>89.81</td>
<td>68.01</td>
<td>70.49</td>
<td>87.97</td>
<td>73.21</td>
<td>70.53</td>
</tr>
<tr>
<td>InstaPBM [22]</td>
<td>80.10</td>
<td>75.87</td>
<td>70.84</td>
<td>89.67</td>
<td>70.21</td>
<td>72.76</td>
<td>89.60</td>
<td>74.41</td>
<td>72.19</td>
<td>87.00</td>
<td>79.66</td>
<td>71.75</td>
</tr>
</tbody>
</table>

Table 2: Per-class average accuracies on OfficeHome RS→UT (right) benchmarks. Bold and underscore denote the best and second-best performing methods respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>Rw → Pr</th>
<th>Rw → Cl</th>
<th>Pr → Rw</th>
<th>Pr → Cl</th>
<th>Cl → Rw</th>
<th>Cl → Pr</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>70.74</td>
<td>44.24</td>
<td>67.33</td>
<td>38.68</td>
<td>55.51</td>
<td>51.85</td>
<td>54.39</td>
</tr>
<tr>
<td>BSP [9]</td>
<td>72.80</td>
<td>23.82</td>
<td>66.19</td>
<td>20.05</td>
<td>32.59</td>
<td>30.36</td>
<td>40.97</td>
</tr>
<tr>
<td>PADA [4]</td>
<td>60.77</td>
<td>32.28</td>
<td>57.09</td>
<td>26.76</td>
<td>40.71</td>
<td>38.34</td>
<td>42.66</td>
</tr>
<tr>
<td>BBSE [24]</td>
<td>61.10</td>
<td>33.27</td>
<td>62.66</td>
<td>31.35</td>
<td>39.70</td>
<td>38.08</td>
<td>44.33</td>
</tr>
<tr>
<td>MCD [37]</td>
<td>66.03</td>
<td>33.17</td>
<td>62.95</td>
<td>29.99</td>
<td>44.47</td>
<td>39.01</td>
<td>45.94</td>
</tr>
<tr>
<td>DAN [25]</td>
<td>69.35</td>
<td>40.84</td>
<td>66.93</td>
<td>34.66</td>
<td>53.55</td>
<td>52.09</td>
<td>52.90</td>
</tr>
<tr>
<td>F-DANN [49]</td>
<td>68.56</td>
<td>40.67</td>
<td>67.32</td>
<td>37.33</td>
<td>55.84</td>
<td>53.67</td>
<td>53.88</td>
</tr>
<tr>
<td>JAN [28]</td>
<td>67.20</td>
<td>43.60</td>
<td>68.87</td>
<td>39.21</td>
<td>57.98</td>
<td>48.57</td>
<td>54.24</td>
</tr>
<tr>
<td>DANN [13]</td>
<td>71.62</td>
<td>46.51</td>
<td>68.40</td>
<td>38.07</td>
<td>58.83</td>
<td>58.05</td>
<td>56.91</td>
</tr>
<tr>
<td>MDD [53]</td>
<td>71.21</td>
<td>44.78</td>
<td>69.31</td>
<td>42.56</td>
<td>52.10</td>
<td>52.70</td>
<td>55.44</td>
</tr>
<tr>
<td>COAL [43]</td>
<td>73.65</td>
<td>42.58</td>
<td>73.26</td>
<td>40.61</td>
<td>59.22</td>
<td>57.33</td>
<td>58.40</td>
</tr>
<tr>
<td>InstaPBM [22]</td>
<td>75.56</td>
<td>42.93</td>
<td>70.30</td>
<td>39.32</td>
<td>61.87</td>
<td>63.40</td>
<td>58.90</td>
</tr>
<tr>
<td>MDD+I.A [19]</td>
<td>76.08</td>
<td>50.04</td>
<td>74.21</td>
<td>45.38</td>
<td>61.15</td>
<td>63.15</td>
<td>61.67</td>
</tr>
</tbody>
</table>

SENTRY (Ours) 76.12 56.80 73.60 54.75 65.94 64.29 65.25

Table 3: Accuracies on standard DA benchmarks.

(a) OfficeHome (12 shift avg) (b) VisDA2017

Results on standard DA benchmarks. Table 3 presents results on 2 standard DA benchmarks: OfficeHome and VisDA 2017. As seen, SENTRY improves mean accuracy over the next best method by 2.7% averaged over 12 shifts (full table in supp.) on OfficeHome, and by 0.4% on VisDA. Varying degree of label imbalance. To perform a controlled study of adapting to targets with varying degrees of label imbalance, we use the SVHN→MNIST shift. Since MNIST is class-balanced, we manually long-tail its train-
We now present ablations of SENTRY, our proposed approach, on the Clipart→Sketch from DomainNet and the Real World→Clipart shift from OfficeHome RS-UT.

**Selective optimization helps significantly (Tab. 5).** We first measure the effect of performing entropy minimization on all samples, as done in prior work. We find this to perform significantly worse (by 10.7%, 11.9%) than our method! Clearly, consistency-based selective optimization is crucial.

**Entropy maximization helps consistently (Tab. 5).** Next, we opt to only minimize entropy on consistent target instances, but do not perform entropy maximization. We find this to underperform against our min-max optimization (by 1.8%, 1.5%). Further, as an oracle approach, we use ground truth target labels to determine whether an instance is correctly or incorrectly classified, and perform two experiments: entropy minimization on correct instances (and no maximization), and min-max entropy optimization on correct and incorrect instances. Selective min-max optimization again outperforms just minimization by 2% and 2.4%, showing that reducing confidence on misaligned instances helps.

**Ablating consistency checker.** In Tables 6a, 6b, we vary the committee size \( k \) and num. consecutive transforms in RandAugment (N). c) Varying voting strategy: maj. and unan. denote majority and unanimous. Gray is our method.

### Table 4: Left: Per-class average accuracy after adapting from SVHN to manually long-tailed (LT) training sets of MNIST (test set is unchanged). The degree of label imbalance is measured by the imbalance factor (IF). All long-tailed versions use an identical amount of data. For each IF, we construct 3 long-tailed versions and report mean and 1 standard deviation. Right: Label distribution for each IF.

<table>
<thead>
<tr>
<th>Method</th>
<th>IF=1</th>
<th>IF=20</th>
<th>IF=50</th>
<th>IF=100</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>68.1</td>
<td>68.1</td>
<td>68.1</td>
<td>68.1</td>
<td>68.1</td>
</tr>
<tr>
<td>MMD [27]</td>
<td>53.4±0.9</td>
<td>56.7±1.2</td>
<td>56.2±1.4</td>
<td>55.1±0.7</td>
<td>55.4±1.1</td>
</tr>
<tr>
<td>DANN [13]</td>
<td>68.0±0.9</td>
<td>71.5±1.0</td>
<td>66.9±0.5</td>
<td>60.6±2.2</td>
<td>66.8±1.5</td>
</tr>
<tr>
<td>COAL [43]</td>
<td>78.8±1.0</td>
<td>67.1±1.4</td>
<td>70.2±1.5</td>
<td>70.0±1.8</td>
<td>71.5±1.4</td>
</tr>
<tr>
<td>InstaPBM [22]</td>
<td>90.7±0.2</td>
<td>77.9±3.5</td>
<td>68.9±1.3</td>
<td>65.9±2.2</td>
<td>75.9±1.8</td>
</tr>
<tr>
<td>SENTRY (Ours)</td>
<td>92.9±0.3</td>
<td>93.9±2.2</td>
<td>85.6±4.5</td>
<td>85.6±1.1</td>
<td>89.5±2.0</td>
</tr>
</tbody>
</table>

### Table 5: Ablations of our selection strategy on DomainNet C→S and OfficeHome RS-UT Rw→Cl. Gray row is our method. Last two rows are oracle approaches that use target labels.

<table>
<thead>
<tr>
<th>select for</th>
<th>select for</th>
<th>DomainNet</th>
<th>OH (RS-UT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>entmin</td>
<td>entmax</td>
<td>C→S</td>
<td>Rw→Cl</td>
</tr>
<tr>
<td>all</td>
<td>none</td>
<td>68.8</td>
<td>44.9</td>
</tr>
<tr>
<td>consistent</td>
<td>inconsistent</td>
<td>79.5</td>
<td>56.8</td>
</tr>
<tr>
<td>correct</td>
<td>none</td>
<td>84.3</td>
<td>77.7</td>
</tr>
<tr>
<td>correct</td>
<td>incorrect</td>
<td>86.3</td>
<td>80.1</td>
</tr>
</tbody>
</table>

### Table 6: Ablating the consistency checker on C→S and Rw→Cl:

<table>
<thead>
<tr>
<th>( N=1 )</th>
<th>( N=3 )</th>
<th>( N=5 )</th>
<th>( N=1 )</th>
<th>( N=3 )</th>
<th>( N=5 )</th>
<th>voting</th>
<th>( C→S )</th>
<th>Rw→Cl</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k=1 )</td>
<td>78.2</td>
<td>78.6</td>
<td>78.9</td>
<td>53.8</td>
<td>57.5</td>
<td>55.6</td>
<td>maj.</td>
<td>79.5</td>
</tr>
<tr>
<td>( k=3 )</td>
<td>76.8</td>
<td>79.5</td>
<td>77.8</td>
<td>55.3</td>
<td>56.8</td>
<td>56.2</td>
<td>unan.</td>
<td>77.8</td>
</tr>
<tr>
<td>( k=5 )</td>
<td>77.5</td>
<td>78.4</td>
<td>77.7</td>
<td>54.7</td>
<td>58.4</td>
<td>54.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) C→S  (b) Rw→Cl  (c) Vary voting
Gains are not simply due to stronger augmentation. To verify this, we continue using RandAugment (with N=1) for consistency checking, but backpropagate on the original (rather than augmented) target instances, effectively removing data augmentation entirely. On C→S and Rw→Cl, this achieves 73.1% and 52.6%, which is still 0.3% and 2.6% better than the next best baseline on each shift, despite not using any data augmentation for optimization at all.

Pseudo class-balanced sampling helps. We find that class-balanced sampling using pseudolabels on the target improves per-class average accuracy over random sampling by 0.91% and 0.52% on C→S and Rw→Cl. In the absence of the target information entropy regularizer $L_{1E}$, this performance gap grows to 2.9% and 3.7%. As explained in Sec 3.4, both objectives contribute towards overcoming LDS in similar ways, and we find here that using both together works best.

4.6. Analysis

% of target instances selected over time. Fig. 4 (left) shows that the % of seen target instances selected for entropy minimization steadily increases over time, while that selected for entropy maximization decreases. This adaptive nature is a result of the augmentation invariance regularization built into our method (Sec. 3). In Fig. 4 (middle), we measure the % of target instances selected for entropy minimization, per-class, at the end of the first and last epoch of adaptation. Despite no explicit class-conditioning, we find that this measure increases for all classes.

Precision of consistency checker. Fig 4 (right) shows the precision of our consistency and inconsistency-based selection strategies at identifying instances that are actually correct and incorrect. As seen, committee-based consistency and inconsistency are both 75-80% precise at identifying correct and incorrect instances respectively.

Per-class accuracy change. In the supplementary, we report the per-class accuracy after adaptation using our method, and contrast it against InstaPBM [22]. On the C→S shift, SENTRY outperforms InstaPBM across 37/40 categories.

Computational efficiency. Compared to standard entropy minimization, SENTRY requires $k$ (for committee size $k$) forward passes per iteration (to determine consistency), but no additional backward passes. SENTRY thus does not add a sizeable computational overhead over prior work.

Correctness of entropy maximization. For simplicity, consider 2-way classification. For output score $p$ and true class $y$, entropy maximization loss $L_{EM} = p \log p + (1 - p) \log(1 - p)$, and binary cross-entropy (BCE) loss $L_{BCE} = -[y \log(p) + (1 - y) \log(1 - p)]$. Without loss of generality, assume an incorrect prediction with $y = 0$ and $0.5 < p < 1$. In Fig. 5 we show that in this case, gradients ($\nabla_p L$) for entropy maximization and BCE (wrt true class) are strongly correlated. Thus, after identifying misaligned target instances based on predictive inconsistency, entropy maximization has a similar effect as supervised training with respective to the true class.

5. Conclusion

We propose SENTRY, an algorithm for unsupervised domain adaptation (UDA) under simultaneous data and label distribution shift. Unlike prior work that suffers from error accumulation arising from unconstrained self-training, SENTRY first judges the reliability of a target instance based on its predictive consistency under a committee of random image transforms, and then selectively minimizes entropy (increasing confidence) on consistent instances, while maximizing entropy (reducing confidence) on inconsistent ones. We show that SENTRY significantly improves upon the state-of-the-art across 27/31 shifts from several UDA benchmarks.

Acknowledgements. This work was supported in part by funding from the DARPA LwLL project.
References


[29] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco,
Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5


[34] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In European conference on computer vision, pages 213–226. Springer, 2010. 1, 2


[51] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2272–2281, 2017. 2

