Active Domain Adaptation via Clustering Uncertainty-weighted Embeddings

Viraj Prabhu[1] **Arjun Chandrasekaran**[*,2] **Kate Saenko**[3] **Judy Hoffman**[1]
[1]Georgia Tech [2]Max Planck Institute [3]Boston University

Abstract

Generalizing deep neural networks to new target domains is critical to their real-world utility. In practice, it may be feasible to get some target data labeled, but to be cost-effective it is desirable to select a subset that is maximally-informative via active learning (AL). In this work, we study the problem of AL under a domain shift. We empirically demonstrate how existing AL approaches based solely on model uncertainty or representative sampling are suboptimal for active domain adaptation. Our algorithm, Active Domain Adaptation via CLustering Uncertainty-weighted Embeddings (`ADA-CLUE`), i) identifies diverse datapoints for labeling that are both uncertain under the model and representative of unlabeled target data, and ii) leverages the available source and target data for adaptation by optimizing a semi-supervised adversarial entropy loss that is complimentary to our active sampling objective. On standard image classification benchmarks for domain adaptation, `ADA-CLUE` consistently performs as well or better than competing active adaptation, active learning, and domain adaptation methods across shift severities, model initializations, and labeling budgets.

1 Introduction

Deep neural networks learn remarkably well from large amounts of labeled data but struggle to generalize this knowledge to new domains [1, 2]. This limits their real-world utility, as it is impractical to exhaustively label a large corpus of data for every problem of interest. Moreover, even labeling all the available data is not a perfect solution, as a deployed model is still likely to encounter some degree of covariate shift [3]. Finally, the cost of labeling is not uniform across applications, and methods that can effectively transfer the knowledge acquired from cheaper sources of labeled data (e.g., synthetic data) to a real-world target would have tremendous utility.

In practice, one may acquire labels for a subset of the target domain to assist in this transfer, but all instances are not equal. While Active Learning (AL) has extensively studied the problem of identifying maximally-informative instances to label [4, 5, 6, 7, 8, 9], it does not address how to effectively use either the labeled source or unlabeled target data for training. Domain adaptation (DA) however, has studied how to adapt a model trained on a labeled source domain to an unlabeled [1, 10, 11, 12] or partially-labeled [13, 14, 15] target domain. While DA may be insufficient to completely bridge challenging domain shifts, it is a natural complement to AL to leverage all the available data.

In this work, we study active domain adaptation (Active DA) [16, 17] – given labeled data in a source domain, unlabeled data in a target domain and the ability to obtain labels for a fixed budget of target instances, the task is to select target instances for labeling and update the model's representations so as to maximize performance on the target test set. Fig. 1 compares the AL, DA and Active DA tasks.

Active DA presents new challenges that AL or DA do not address. In AL, labels are typically acquired for instances that are highly uncertain [6, 8, 18, 19] or representative in feature space [7, 20]. However, AL often assumes learning from scratch, at which point model uncertainty may be unreliable. In

[*]Work done partially at Georgia Tech. Correspondence to: Viraj Prabhu <virajp@gatech.edu>

Preprint. Under review.

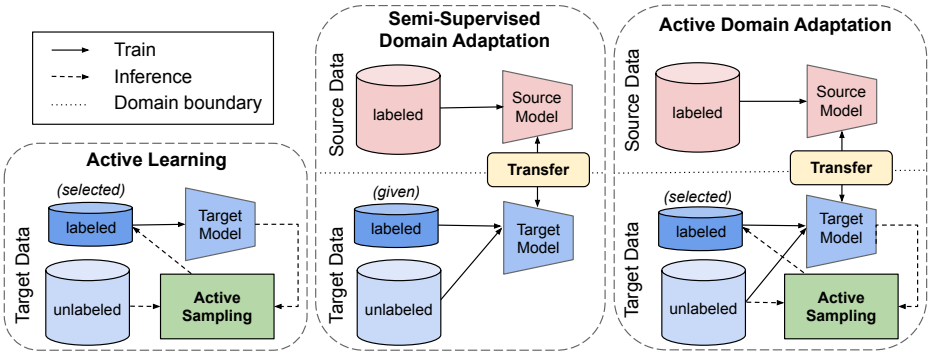


Figure 1: Active learning (*left*) aims to identify the most informative target instances for labeling (deep blue), and using those to train a model. Semi-supervised domain adaptation (SSDA) (*middle*) seeks to generalize a model trained on a large labeled source domain (pink) to an unlabeled target domain (light blue) given a small number of target labels (deep blue). In this work, we address **Active Domain Adaptation** (*right*), where the task is to generalize a source model to an unlabeled target domain by acquiring labels for selected target instances via an oracle.

Active DA, on the one hand it is possible to learn a strong model initialization on the source domain, which may lead to a more reliable feature space and uncertainty estimates. On the other hand, the calibration of these estimates in the target domain depends on the severity of the domain shift [21]. In fact, in our experiments, we find that prior work in Active DA [17] that proposes label acquisition strategies guided solely by uncertainty, do not generalize to challenging domain shifts. This makes it challenging to develop a versatile label acquisition strategy for Active DA. Further, while semi-supervised DA assumes that labeled instances are given, Active DA raises the challenging question of identifying instances that will, once labeled, efficiently transfer "knowledge" from the source to the target. An effective solution to Active DA needs to jointly address these questions.

We present a novel algorithm called Active Domain Adaptation via CLustering Uncertainty-weighted Embeddings (`ADA-CLUE`) for Active DA, which addresses the above challenges. First, we propose `CLUE`, a novel label acquisition strategy for Active DA, which, unlike prior work in active adaptation [17], acquires labels for a diverse set of instances (forming non-redundant training batches) that are uncertain (informative to the model), and representative of the target data distribution (likely to generalize better to the target test set). `CLUE` clusters deep embeddings of target datapoints weighted by the corresponding uncertainty of the target model, and selects nearest neighbors to the inferred cluster centroids for labeling. `ADA-CLUE` then leverages all of the available data to update the model via optimizing an adversarial entropy loss for semi-supervised DA [15], which we demonstrate to be complementary to our label acquisition strategy.

We present results on five standard image-classification based domain adaptation shifts: the relatively simple SVHN [22]→MNIST [23] shift from the DIGITS benchmark, and 4 shifts of increasing difficulty on the large and substantially more challenging DomainNet [24] benchmark. We demonstrate consistent performance gains over competing active adaptation, active learning, and semi-supervised domain adaptation methods on 4 out of 5 shifts and perform as well as the best competing method on the fifth. We analyze the robustness of our method across active sampling strategies, domain adaptation methods, model initializations, and labeling budgets. In addition, we present ablation studies of our model and analyze its behavior via visualizations.

2 Related Work

Active Learning. Classically studied under both streaming (picking one instance at a time) and batch-mode settings (picking a batch of instances), active learning for CNN's has focused on the latter due to the computational inefficiency and instability associated with single-instance updates. Within this setting, the most successful paradigms that have emerged have been uncertainty-based sampling and representative sampling [9]. Uncertainty-based methods pick datapoints with the highest uncertainty under the current model [6, 8, 18, 25]. Several uncertainty measures have been proposed, including maximum entropy [26], minimum classification margins [27], least confidence, etc. On the other hand, representative sampling-based methods pick a set of points that are representative of the entire dataset, and optimize for diversity or coverage, via clustering, or core-set selection [7, 20, 28].

Some approaches combine these two paradigms [9, 29, 30, 31]. Active Learning by Learning [30] formulates this as a multi-armed bandit problem of selecting between coreset and uncertainty sampling at each step. Zhdanov et al. [31] propose using K-Means clustering [32] to increase batch diversity following pre-filtering based on uncertainty. A more recent example is BADGE [9], which first computes "gradient embeddings" on unlabeled points, and then runs a clustering scheme on these to construct diverse batches. In this work, we propose a label acquisition algorithm that captures uncertainty, representativeness, and diversity, for the problem of active learning under a domain shift.

Domain Adaptation. The problem of transferring models trained on a labeled source domain to an unlabeled [1, 10, 11, 12] or partially-labeled [13, 14, 15] target domain has been studied extensively. Initial approaches aligned feature spaces via direct optimization of discrepancy statistics between the source and target [10, 33], while in recent years adversarial learning of a feature space encoder alongside a domain discriminator has been the dominant alignment paradigm [11, 34, 35]. More recently, min-max optimization of model entropy has been shown to successfully achieve domain-alignment in a semi-supervised setting [15]. In this work we apply adversarial entropy optimization to achieve alignment for Active DA in the context of image classification.

Active Domain Adaptation. Rai et al. [16] initially studied the task of active adaptation applied to sentiment classification from text data. They propose ALDA, which employs source domain initialization and a sampling strategy based on a learned domain separator. Chattopadhyay et al. [36] select target samples and learn importance weights for source points by solving a convex optimization problem of minimizing maximum mean discrepancy (MMD) between features. More recently, Su et al. [17] study this task in the context of deep CNN's and propose AADA, an Active DA method wherein points are sampled based on their uncertainty (measured by model entropy) and targetness (measure by a domain discriminator), followed by adversarial domain adaptation [34]. In our work, we propose an Active DA algorithm that selects batches of points that are uncertain, representative, and diverse, followed by semi-supervised DA via adversarial entropy minimization, that results in significantly better performance than prior work across domain shifts of varying difficulty.

3 Approach

We address the problem of active domain adaptation (Active DA), where the goal is to generalize a model trained on a source domain to an unlabeled target domain, with the option to query an oracle for labels for a subset of target instances. While individual aspects of this problem – generalization to a new domain and selective acquisition of labels, have been well studied as the problems of Domain Adaptation (DA) and Active Learning (AL) respectively, Active DA presents new challenges. First, it is unclear as to which target instances will, once labeled, result in the most sample-efficient domain alignment. It is also an open question as to how best to use the labeled data from the source or the unlabeled data from the target for training. Further, the optimal solutions to these questions may vary based on the properties of the specific domain shift. In this section, we present an algorithm for Active DA which performs consistently well across domain shifts of varying difficulty.

3.1 Notation

In Active DA, the learning algorithm has access to labeled instances from the source domain $(X_{\mathcal{S}}, Y_{\mathcal{S}})$ (solid pink in Fig. 2), unlabeled instances from the target domain $X_{\mathcal{UT}}$ (blue outline in Fig. 2), and a budget B ($= 3$ in Fig. 2) which is much smaller than the amount of unlabeled target data. The learning algorithm may query an oracle to obtain labels for at most B instances from $X_{\mathcal{UT}}$, and add them to the set of labeled target instances $X_{\mathcal{LT}}$. The entire target domain data is $X_{\mathcal{T}} = X_{\mathcal{LT}} \cup X_{\mathcal{UT}}$. The task is to learn a function $h : X \rightarrow Y$ (a convolutional neural network (CNN) parameterized by Θ) that achieves good prediction performance on the target. In this work, we consider Active DA in the context of C-way image classification – the samples $\mathbf{x}_{\mathcal{S}} \sim X_{\mathcal{S}}$, $\mathbf{x}_{\mathcal{T}} \sim X_{\mathcal{T}}$ are images and labels $y_{\mathcal{S}} \sim Y_{\mathcal{S}}$, $y_{\mathcal{T}} \sim Y_{\mathcal{T}}$ are categorical variables $y \in \{1, 2, .., C\}$.

3.2 CLUE: CLustering Uncertainty-weighted Embeddings

The goal in active learning (AL) is to identify target instances that, once labeled and used for training the model, minimize its expected future loss. In practice, prior works in AL identify such informative instances based on various proxy measures, e.g., sampling instances that are uncertain, representative, or diverse (see Sec. 2). We propose `CLUE`, a novel sampling strategy for Active DA that uses a combination of these proxies.

Uncertain. Identifying instances that provide the model with new information is essential. Prior work has proposed using several measures of model uncertainty as a proxy for informativeness (see

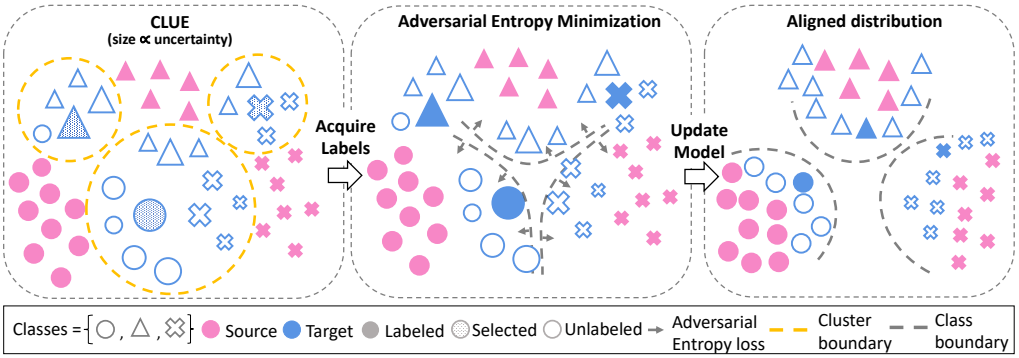


Figure 2: Our approach, Active Domain Adaptation via CLustering Uncertainty-weighted Embeddings (ADA-CLUE), acquires labels for a diverse set of target instances that are informative and representative (left). First, uncertainty-weighted embeddings of target instances, are clustered. The instance closest to each cluster centroid is then acquired for labeling, providing diverse instances that are representative and informative (Eq. 8). Next, ADA-CLUE leverages the available labeled and unlabeled source and target data to update the model via a semi-supervised adversarial entropy objective (Eq. 4) (middle), which results in well-classified target data (right).

Sec. 2). When learning from scratch, model uncertainty may be unreliable and lead to sampling less informative points. In Active DA, however, models benefit from initialization on a related source domain, which leads to correspondingly more reliable uncertainty estimates. We employ the commonly-used entropy measure $\mathcal{H}(Y|\mathbf{x};\Theta)$ as our uncertainty measure [26], $\mathcal{H}(Y|\mathbf{x})$ for brevity. For C-way classification, entropy is defined as:

$$\mathcal{H}(Y|\mathbf{x}) = -\sum_{c=1}^{C} p(Y=c|\mathbf{x};\Theta)\log p(Y=c|\mathbf{x};\Theta) \tag{1}$$

Representative. Acquiring labels solely based on uncertainty may not always be reliable, particularly under a strong domain shift [21], or in the presence of noisy outliers. Without access to target labels, it is difficult to self-diagnose this. A parallel line of work in active learning instead sampling proposes instances that are representative of the unlabeled pool of data. Prior work has framed this as a computational geometry problem of core-set selection [7] or computing "gradient embeddings" that capture feature geometry [9], in a learned high-dimensional space. The success of such methods relies on access to a meaningful embedding space. In the context of Active DA, training on the source domain leads to meaningful priors that are amenable to such representative sampling.

Let $\phi(\mathbf{x})$ denote feature embeddings extracted from model h. One way to identify representative instances is by partitioning X_T into K diverse sets $\mathcal{S} = \{X_1, X_2, ..., X_K\}$, where each set X_k has small variance $\sigma^2(X_k)$. Expressed in terms of pairs of samples, $\sigma^2(X_k) = \frac{1}{2|X_k|^2}\sum_{\mathbf{x_i},\mathbf{x_j}\in X_k}||\phi(\mathbf{x_i}) - \phi(\mathbf{x_j})||^2$ [37]. The goal is to group target instances that are similar in the CNN's feature space, into a set X_k. However, while $\sigma^2(X_k)$ is a function of the target data distribution and feature space $\phi(.)$, it does not account for the informativeness of an instance. Source training followed by domain alignment typically results in some classes being better aligned across domains than others. Thus, it is important to avoid sampling from already well-classified regions of the feature space. We achieve this by weighting samples based on their informativeness (uncertainty given by Eq. 1), and compute the weighted population variance [38]. The overall set-partitioning objective is given by:

$$\underset{\mathcal{S}}{\operatorname{argmin}} \sum_{k=1}^{K}\frac{1}{Z}\sum_{\mathbf{x}\in X_k}\mathcal{H}(Y|\mathbf{x})||\phi(\mathbf{x}) - \mu_\mathbf{k}||^2 \quad \text{where} \quad \mu_\mathbf{k} = \frac{1}{Z}\sum_{\mathbf{x}\in X_k}\mathcal{H}(Y|\mathbf{x})\phi(\mathbf{x}) \tag{2}$$

where the normalization $Z = \sum_{x\in X_k}\mathcal{H}(Y|\mathbf{x})$. In practice, we implement the Weighted K-Means algorithm [39] which is an approximation to Eq. 8, set $K = B$ (budget), and use activations from the penultimate CNN layer as embeddings $\phi(\mathbf{x})$.

Diverse. AL for deep CNN's has primarily focused on the batch active learning setting [40] for better efficiency and computational stability. In this setting, greedily sampling the K most 'uncertain' instances will likely lead to high redundancy within a batch. Since Eq. 8 equivalently maximizes the sum of squared deviations between instances in different sets [41], we construct a batch of instances

Algorithm 1 ADA-CLUE: Our proposed Active DA method, which uses Clustering with Uncertainty-weighted Embeddings (CLUE) to select points for labeling followed by a model update via semi-supervised adversarial entropy minimization.

Require: Neural network $h = f(\phi(.))$, parameterized by Θ, labeled source instances $(X_{\mathcal{S}}, Y_{\mathcal{S}})$, unlabeled target instances $X_{\mathcal{T}}$, Per-round budget B, Total rounds R.
Define: Target labeled set $X_{\mathcal{LT}} = \emptyset$
Update model at the first round Θ^1 by minimizing Eq. 3. ▷ Train source model
Adapt model to unsupervised target data by optimizing Eq. 4. ▷ Unsupervised adaptation
for $\rho = 1$ to R **do**
    CLUE: For all instances $x \in X_{\mathcal{T}} \setminus X_{\mathcal{LT}}$:
    1. Compute entropy-weighted embedding $H(Y|\mathbf{x})\phi(\mathbf{x})$
    2. Solve Eq. 8 via Weighted K-Means ($K = B$)
    3. Acquire labels for nearest-neighbor to centroids $X^{\rho}_{\mathcal{LT}} = \{\mathbf{NN}(\mu_{\mathbf{k}}); k = 1, 2, .., K\}$
    4. $X_{\mathcal{LT}} = X_{\mathcal{LT}} \cup X^{\rho}_{\mathcal{LT}}$
    Semi-supervised Domain Adaptation. Update model $\Theta^{\rho+1}$ by optimizing Eq. 4.
Return: Final model parameters Θ^{R+1}.

with minimum overlap by selecting from different sets. In practice, we sample the nearest neighbor to the weighted-mean of each set $\mu_{\mathbf{k}}$ in Eq. 8, acquiring labels for K instances.

Our full label acquisition approach, Clustering Uncertainty-weighted Embeddings (CLUE), identifies a set of instances that are together informative, diverse, and representative of the data (Fig. 2, *left*).

3.3 Semi-supervised Domain Adaptation

Given labeled samples from the source and (a subset of) the target domain, we first compute the total cross-entropy loss $\mathcal{L}_{TCE}$ of model h over the available labeled data.

$$\mathcal{L}_{TCE} = \lambda_{\mathcal{S}} \mathbb{E}_{(\mathbf{x},y)\in(X_{\mathcal{S}},Y_{\mathcal{S}})}[\mathcal{L}_{CE}(h(\mathbf{x}), y)] + \lambda_{\mathcal{T}} \mathbb{E}_{(\mathbf{x},y)\in(X_{\mathcal{LT}},Y_{\mathcal{LT}})}[\mathcal{L}_{CE}(h(\mathbf{x}), y)] \tag{3}$$

where λ_S and λ_T are scalar weights, and $\mathcal{L}_{CE}$ denotes the cross-entropy loss. In addition, the source and target domains are aligned via an additional objective that uses both labeled and unlabeled data. We use the minimax entropy (MME) alignment strategy for semi-supervised domain adaptation proposed in Saito et al. [15]. While conventional domain classifier-based adaptation methods are effective in unsupervised feature alignment, they tend to generate ambiguous features near class boundaries in the presence of any target supervision. MME was shown to overcome this challenge and excel at semi-supervised domain adaptation. We now describe the MME approach.

Consider that model $h : X \to Y$ is composed of a feature extractor $\phi_\alpha(\mathbf{x}) : X \to Z; Z \in \mathbb{R}^M$ and classifier $f_W : Z \to Y$. Overall, the model's prediction is $p(Y|\mathbf{x}; \Theta) = \sigma(f(\phi(\mathbf{x})))$ where α and W are parameters for ϕ and f respectively, and σ denotes the softmax function. In MME, the classifier weights W are updated to maximize model entropy (Eq. 1) over target instances, and parameters of the feature extractor α are updated to minimize it. The full learning objective is given by:

$$\underset{W}{\operatorname{argmin}}\, \mathcal{L}_{TCE} - \lambda_{\mathcal{H}} \sum_{\mathbf{x}\in X_{\mathcal{T}}} \mathcal{H}(Y|x) \qquad \underset{\alpha}{\operatorname{argmin}}\, \mathcal{L}_{TCE} + \lambda_{\mathcal{H}} \sum_{\mathbf{x}\in X_{\mathcal{T}}} \mathcal{H}(Y|x) \tag{4}$$

where $\lambda_{\mathcal{H}}$ is a hyperparameter that controls the relative weight of the unsupervised alignment term. Our label-acquisition (CLUE) and domain alignment strategies (MME) complement one another. MME explicitly minimizes target entropy, which has the effect of producing a feature space where similar points are more tightly clustered (see Fig. 2, *middle*). This makes it easier to sample diverse points with CLUE. Further, sampling points close to class decision boundaries purely based on uncertainty followed by adversarial entropy optimization can lead to learning ambiguous decision boundaries (details in appendix), whereas CLUE captures both uncertain and representative points.

We call our entire active adaptation approach Active Domain Adaptation via Clustering Uncertainty-weighted Embeddings (ADA-CLUE, see Algorithm 1). Given a model trained on labeled source instances, we align its representations with unlabeled target instances via unsupervised domain adaptation. For R rounds with per-round budget B, we then iteratively i) acquire labels for B target instances that are identified via our proposed sampling approach (CLUE), and ii) Update the model using the semi-supervised domain alignment strategy described above (Eq. 4).

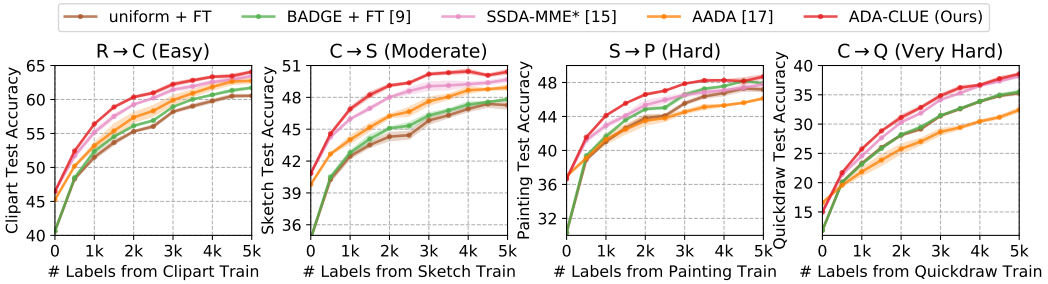


Figure 3: Active DA accuracy across 4 shifts of increasing difficulty from DomainNet [24], over 10 rounds with per-round budget $B = 500$: `ADA-CLUE` consistently performs as well or better than a state-of-the-art active learning (`BADGE` [9]), semi-supervised DA (`MME` [15]), and active DA (`AADA` [17]) method.

4 Experiments

We begin by evaluating the performance of our active domain adaptation method (`ADA-CLUE`) across 5 domain shifts of varying difficulty (Sec 4.1). To demonstrate the importance of each component of our method, we next present ablations of our sampling strategy (`CLUE`) and our choice of semi-supervised adaptation (Sec 4.2). For all experiments, we follow the standard batch active learning setting [42], in which we perform multiple rounds of batch active sampling, label acquisition, and model updates. As our performance metric, we compute model accuracy on the target test split versus the number of labels used from the target train split at each round. We run each experiment 3 times and report accuracy mean and 1 standard deviation.

4.1 Performance on the active adaptation task

DomainNet [24] is the largest domain adaptation benchmark for image classification, containing 0.6 million images belonging to 6 distinct domains spanning 345 categories. For our experiments, we study four shifts of increasing difficulty as measured by source→target transfer accuracy: Real→Clipart (easy), Clipart→Sketch (moderate), Sketch→Painting (hard), and Clipart→Quickdraw (very hard). We use a ResNet34 [43] CNN , and perform 10 rounds of Active DA with per-round budget $= 500$ instances (total of 5000 labels). In addition, we evaluate performance on the SVHN [22]→MNIST [23] shift used in Su et al. [17]. We use a modified LeNet architecture [12], and perform 30 rounds of active adaptation with a per-round budget $= 10$. See appendix for additional details regarding datasets and training.

Baselines. We compare our method against four baselines: State-of-the-art methods for active domain adaptation (`AADA` [17]), semi-supervised DA (`SSDA-MME*` [15]), active learning (`BADGE` [9]), and a simple baseline of uniform sampling with finetuning (`uniform + finetuning (FT)`). We briefly describe the three state-of-the-art methods below.

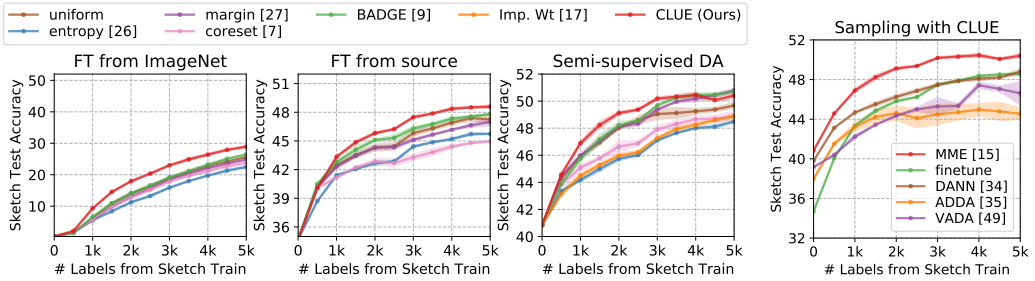
(i) `AADA`**:** Active Adversarial Domain Adaptation [17] performs alternate rounds of active sampling and adversarial domain adaptation via `DANN` [34]. This method samples points that are jointly high entropy and have a high "targetness" score from the domain discriminator.

ii) `SSDA-MME*`**:** [15] The model is optimized via the semi-supervised `MME` loss on randomly sampled target points. The asterisk denotes that for simplicity and fair comparison against baselines, in our implementation we do not use a similarity-based classifier or L2-normalize features [2].

iii) `BADGE + finetuning (FT)`**:** Batch Active Learning by Diverse Gradient Embeddings [9] is a recently proposed, state-of-the-art active learning strategy that constructs diverse batches by running KMeans++ [44] on "gradient embeddings" that incorporate model uncertainty and diversity. The model is then finetuned on acquired labels.

All baseline models are first initialized with pretrained ImageNet weights and then trained to completion on the labeled source domain. `SSDA-MME`, `AADA`, and `ADA-CLUE` additionally employ unsupervised feature alignment to the target domain, which leads to a higher initial performance.

[2] We report results on all 345 classes in DomainNet instead of the 126-class subset that Saito et al. [15] use.



(a) Active Sampling Ablation (b) DA Ablation

Figure 4: ADA-CLUE ablation Clipart→Sketch: (a): Comparing CLUE against prior active sampling strategies across 3 starting points and training strategies: finetuning from pretrained ImageNet weights, finetuning from a source model, and semi-supervised DA (using MME [15]) from a source model. (b): Varying semi-supervised DA strategies while sampling via CLUE.

Results. Fig. 3 and 5 demonstrate our results on DomainNet and DIGITS. We find that ADA-CLUE consistently outperforms alternative methods across shifts and rounds, except for the last few rounds on the very hard C→Q shift where it is on par with the SSDA-MME* baseline. For instance, on the Clipart→Sketch (Fig. 3, C→S) shift, we improve upon the state-of-the-art AADA active adaptation method by 2-4% over rounds. This demonstrates the benefit of jointly capturing model uncertainty, density, and batch diversity for active sampling in combination with strong semi-supervised feature alignment. We observe similar improvements on the DIGITS SVHN→MNIST benchmark (Fig. 5).

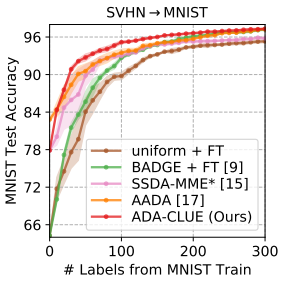


Figure 5: Active DA performance on SVHN [22]→MNIST [23]

Importantly, we find that the performance gap between ADA-CLUE and AADA [17] increases with increasing shift difficulty. As discussed in Sec. 3, the optimal label acquisition criterion may vary across different shifts and stages of training as the model's uncertainty estimates and feature space evolve, and it is challenging for a single approach to work well across all settings. However we find that ADA-CLUE is able to effectively trade-off uncertainty and feature-space coverage and perform well even on difficult shifts, significantly outperforming purely uncertainty-based methods (eg. AADA).

Across shifts, we find that the SSDA-MME* [15] baseline is closest to ADA-CLUE in terms of performance. This is unsurprising since Mittal et al. [45] showed that the benefit of deep active learning strategies is often greatly reduced when deployed in combination with semi-supervised learning and data augmentation. Despite using both of these, we observe that ADA-CLUE provides consistent and significant improvements, on most shifts, over uniform sampling even in the presence of strong semi-supervised alignment. This offers encouraging evidence that there remains value in intelligently sampling instances for labeling even with semi-supervised deep learning.

4.2 Ablating ADA-CLUE

Active sampling ablation. To understand the impact of our active sampling method, CLUE, we consider fixing the semi-supervised domain adaptation method from our full method, ADA-CLUE, and varying the active sampling method using 4 diverse AL strategies from prior work. Labels are acquired based on the following selection strategies: 1) entropy [26]: Instances over which the model has highest predictive entropy. 2) margin [27]: Instances for which the score between the model's top-2 predictions is the smallest. 3) Coreset [7]: Core-set formulates active sampling as a set-cover problem, and solves the K-Center [46] problem. In our experiments, we use the greedy version proposed in Sener et al. [7]. 4) BADGE [9] (described in Sec. 4.1). Strategies (1) and (2) are purely uncertainty based, (3) is purely based on representativeness, and (4) is a hybrid approach.

We evaluate all sampling strategies across three different ways of learning with the acquired labeled data – finetuning a model pretrained on ImageNet [47], finetuning a model trained on the source domain, and semi-supervised domain alignment via minimax entropy [15] (Eq. 4). In the third

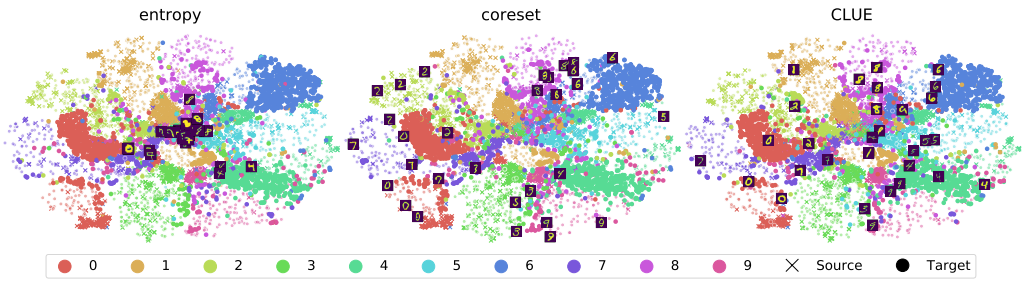


Figure 6: SVHN→MNIST: We visualize the logits of a subset of incorrect (large, opaque circles) and correct (partly transparent circles) model predictions on the target domain after round 0, along with examples sampled by different methods. Entropy (*left*) acquires redundant samples, whereas core-set (*middle*) does not account for areas of the feature space that are already well understood. CLUE (*right*) constructs batches of dissimilar samples from dense regions with high uncertainty.

scenario, we also benchmark the performance of the importance weighting strategy used for sample selection in AADA (see Sec. 4.1). Fig. 4a summarizes results on Clipart→Sketch.

As expected, raw performance is weakest with ImageNet initialization and strongest with semi-supervised DA. Regardless, sampling using CLUE provides the most informative samples leading to the best overall performance in each scenario and across most sampling budgets (lower only than BADGE+MME using the largest budget). In the appendix, we also benchmark the performance of CLUE as an active learning method on SVHN, and find it to be on-par with state-of-the-art methods.

Finally, we provide an illustrative comparison of sampling strategies using TSNE [48]. Fig. 6 shows an initial feature landscape together with points selected from `entropy`, `coreset`, and CLUE at Round 0 on the SVHN→MNIST shift. We find that `entropy` (*left*) samples informative but redundant points, `coreset` samples diverse but not necessarily informative points, while our method, CLUE, samples both diverse and informative points.

Domain adaptation ablation. We next justify our decision to use MME as our semi-supervised domain adaptation method for active domain adaptation. For this experiment, we fix our sampling strategy to CLUE and compare against three domain-classifier based adaptation methods: DANN [34], ADDA [35], and VADA [49], as well as standard finetuning. In Fig. 4b, we observe that domain alignment with MME significantly outperforms all alternative domain adaptation methods as well as finetuning. Interestingly, not all domain adaptation methods perform better than simple finetuning. This finding is consistent with Saito et al. [15], who find that domain-classifier based methods are not as effective in the semi-supervised setting when additional target labels are available.

For further analysis, see appendix. We include additional experiments such as augmenting DANN [34] and ADDA [35] with entropy minimization regularization, as commonly used in semi-supervised learning and in AADA [17] and find that this addition consistently improves performance.

Additional ablations. Due to space limitations, we include our remaining ablations in the appendix. These include comparisons over uncertainty metrics within CLUE, varying the temperature of the softmax before computing entropy weighting within CLUE, and analyzing the robustness of our method when varying the per round budget within a fixed overall budget. With these ablations we justify our choice of entropy as the uncertainty measure in CLUE and show that using small temperatures leads to the best results. Finally, we demonstrate that ADA-CLUE has performance robust against batch sampling budget as long as the budget is greater than the number of dataset categories.

5 Conclusion

We address active domain adaptation, where the task is to generalize a source model to an unlabeled target domain by acquiring labels for selected target instances via an oracle. We present ADA-CLUE, an algorithm for active domain adaptation that first identifies diverse instances from the target domain for labeling that are both uncertain under the model and representative of target data, and then optimizes a semi-supervised adversarial entropy loss to induce domain alignment. We demonstrate its effectiveness on the Active DA task against competing active learning, semi-supervised domain adaptation, and active adaptation methods across domain shifts of varying difficulty.

Broader Impacts

The widespread successes of deep neural networks in recent years have largely relied on large labeled datasets. However, labeling costs can be prohibitively expensive for some applications, requiring specialized expertise (labeling X-rays for medical diagnosis) or significant manual effort (pixel-level annotations for semantic segmentation). As such, reusing knowledge from cheaper sources of labels (eg. synthetic data) to generalize to new tasks and datasets is an important but unsolved challenge.

Our work focuses on cost-efficient generalization by identifying a small subset of target data that will, once labeled, lead to good target performance. We anticipate our line of work to enable new applications to adapt efficiently. In terms of impact on society, this could mean that computer vision systems are able to better handle novel deployments and are less susceptible to dataset bias. For example, our system could adapt a skin-cancer detection system to an image dataset that was taken under different lighting conditions with less annotation. Although we do not experiment on fairness applications, domain adaptation has also been shown to improve the fairness of face recognition systems across race/gender.

On the other hand, identifying informative points with our method does require additional processing of the available unlabeled data, but we believe this cost is offset by the reduced carbon footprint of ultimately having to train on a much smaller subset of data. Other negative impacts of our research on society are harder to predict, but it suffers from the same issues as most deep learning algorithms. These include adversarial attacks, privacy concerns and lack of interpretability, as well as other negative effects of increased automation.

References

[1] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*, pp. 213–226, Springer, 2010.

[2] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1521–1528, IEEE, 2011.

[3] M. Sugiyama, M. Krauledat, and K.-R. MÃžller, "Covariate shift adaptation by importance weighted cross validation," *Journal of Machine Learning Research*, vol. 8, no. May, pp. 985–1005, 2007.

[4] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine learning*, vol. 15, no. 2, pp. 201–221, 1994.

[5] B. Settles, "Active learning literature survey," tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.

[6] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1183–1192, JMLR. org, 2017.

[7] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *International Conference on Learning Representations*, 2018.

[8] M. Ducoffe and F. Precioso, "Adversarial active learning for deep networks: a margin based approach," *arXiv preprint arXiv:1802.09841*, 2018.

[9] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, "Deep batch active learning by diverse, uncertain gradient lower bounds.," in *International Conference on Learning Representations*, 2020.

[10] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[11] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, pp. 1180–1189, 2015.

[12] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International Conference on Machine Learning*, pp. 1989–1998, 2018.

[13] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 668–675, 2013.

[14] T. Yao, Y. Pan, C.-W. Ngo, H. Li, and T. Mei, "Semi-supervised domain adaptation with subspace learning for visual recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2142–2150, 2015.

[15] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8050–8058, 2019.

[16] P. Rai, A. Saha, H. Daumé III, and S. Venkatasubramanian, "Domain adaptation meets active learning," in *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pp. 27–32, Association for Computational Linguistics, 2010.

[17] J.-C. Su, Y.-H. Tsai, K. Sohn, B. Liu, S. Maji, and M. Chandraker, "Active adversarial domain adaptation," in *The IEEE Winter Conference on Applications of Computer Vision*, pp. 739–748, 2020.

[18] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of machine learning research*, vol. 2, no. Nov, pp. 45–66, 2001.

[19] A. Kirsch, J. van Amersfoort, and Y. Gal, "Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning," in *Advances in Neural Information Processing Systems*, pp. 7024–7035, 2019.

[20] D. Gissin and S. Shalev-Shwartz, "Discriminative active learning," *arXiv preprint arXiv:1907.06347*, 2019.

[21] J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J. Dillon, J. Ren, and Z. Nado, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Advances in Neural Information Processing Systems*, pp. 13969–13980, 2019.

[22] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[24] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

[25] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *ICML*, vol. 2, p. 6, Citeseer, 2000.

[26] D. Wang and Y. Shang, "A new active labeling method for deep learning," in *2014 International joint conference on neural networks (IJCNN)*, pp. 112–119, IEEE, 2014.

[27] D. Roth and K. Small, "Margin-based active learning for structured output spaces," in *European Conference on Machine Learning*, pp. 413–424, Springer, 2006.

[28] Y. Geifman and R. El-Yaniv, "Deep active learning over the long tail," *arXiv preprint arXiv:1711.00941*, 2017.

[29] Y. Baram, R. E. Yaniv, and K. Luz, "Online choice of active learning algorithms," *Journal of Machine Learning Research*, vol. 5, no. Mar, pp. 255–291, 2004.

[30] W.-N. Hsu and H.-T. Lin, "Active learning by learning," in *Twenty-Ninth AAAI conference on artificial intelligence*, 2015.

[31] F. Zhdanov, "Diverse mini-batch active learning," *arXiv preprint arXiv:1901.05954*, 2019.

[32] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[33] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning*, pp. 97–105, 2015.

[34] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[35] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7167–7176, 2017.

[36] R. Chattopadhyay, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Joint transfer and batch-mode active learning," in *International Conference on Machine Learning*, pp. 253–261, 2013.

[37] Y. Zhang, H. Wu, and L. Cheng, "Some new deformation formulas about variance and covariance," in *2012 Proceedings of International Conference on Modelling, Identification and Control*, pp. 987–992, IEEE, 2012.

[38] G. R. Price, "Extension of covariance selection mathematics," *Annals of human genetics*, vol. 35, no. 4, pp. 485–490, 1972.

[39] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 5, pp. 657–668, 2005.

[40] Z. Wang and J. Ye, "Querying discriminative and representative samples for batch mode active learning," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 9, no. 3, pp. 1–23, 2015.

[41] H.-P. Kriegel, E. Schubert, and A. Zimek, "The (black) art of runtime evaluation: Are we comparing algorithms or implementations?," *Knowledge and Information Systems*, vol. 52, no. 2, pp. 341–378, 2017.

[42] K. Brinker, "Incorporating diversity in active learning with support vector machines," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 59–66, 2003.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[44] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," tech. rep., Stanford, 2006.

[45] S. Mittal, M. Tatarchenko, Ö. Çiçek, and T. Brox, "Parting with illusions about deep active learning," *arXiv preprint arXiv:1912.05361*, 2019.

[46] G. W. Wolf, "Facility location: concepts, models, algorithms and case studies. series: Contributions to management science: edited by zanjirani farahani, reza and hekmatfar, masoud, heidelberg, germany, physica-verlag, 2009," 2011.

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[48] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[49] R. Shu, H. H. Bui, H. Narui, and S. Ermon, "A dirt-t approach to unsupervised domain adaptation," in *Proc. 6th International Conference on Learning Representations*, 2018.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[52] A. David, "Vassilvitskii s.: K-means++: The advantages of careful seeding," in *18th annual ACM-SIAM symposium on Discrete algorithms (SODA), New Orleans, Louisiana*, pp. 1027–1035, 2007.

[53] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," in *Proceedings of the eleventh international conference on Information and knowledge management*, pp. 600–607, 2002.

[54] C. Elkan, "Using the triangle inequality to accelerate k-means," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 147–153, 2003.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[56] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *International Conference on Learning Representations*, 2019.

6 Appendix

Contents

1 Introduction **1**

2 Related Work **2**

3 Approach **3**

3.1 Notation 3

3.2 CLUE: CLustering Uncertainty-weighted Embeddings 3

3.3 Semi-supervised Domain Adaptation 5

4 Experiments **6**

4.1 Performance on the active adaptation task 6

4.2 Ablating ADA-CLUE 7

5 Conclusion **8**

6 Appendix **12**

7 Further Ablations for ADA-CLUE **13**

7.1 Varying uncertainty measure in CLUE 13

7.2 Softmax temperature to trade-off uncertainty and representativeness in ADA-CLUE. . 13

7.3 How many active adaptation rounds are optimal? 13

7.4 Convergence: When do gains saturate? 14

7.5 Role of Entropy Minimization 14

8 Performance of CLUE on Standard Active Learning **14**

9 Dataset details **15**

10 Code and Implementation Details **15**

10.1 Baseline Implementations 16

11 Understanding ADA-CLUE with Qualitative Examples **16**

12 Extended Description of the CLUE Objective **18**

13 Future Work **19**

7 Further Ablations for ADA-CLUE

7.1 Varying uncertainty measure in CLUE

In Fig. 7, we consider alternative uncertainty measures used in CLUE, our proposed label acquisition strategy. We show the Clipart→Sketch (C→S) shift from DomainNet [24]. As before, we perform 10 rounds of active domain adaptation with a per-round budget = 500, and report accuracy on the target test split as a function of the number of labels from the target train split. We repeat experiments thrice and report accuracy mean with 1 standard deviation.

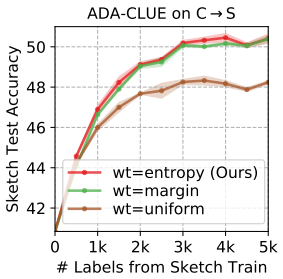


Figure 7: Varying uncertainty measures in CLUE.

We show that our proposed use of sample entropy significantly outperforms a uniform sample weight and performs comparably to an alternative uncertainty measure - sample margin score (difference between scores for top-2 most likely classes). This implies that either measure which considers the prediction uncertainty may be used to bias towards informative samples and that simple K-Means without sample specific uncertainty weighting is inferior.

We also run an additional experiment (not shown in the figure) where we use embeddings from the last layer CNN instead of the penultimate layer for CLUE. We observe near-identical performance in both cases across multiple shifts, suggesting that our method is not very sensitive to this choice.

7.2 Softmax temperature to trade-off uncertainty and representativeness in ADA-CLUE.

CLUE captures an implicit tradeoff between model uncertainty (via entropy weighting) and feature-space coverage (via clustering). In Sec. 4.1 in the main paper, we have shown this implicit tradeoff leads to consistent improvements across shifts of varying difficulty and model initializations. We now study this tradeoff in more detail.

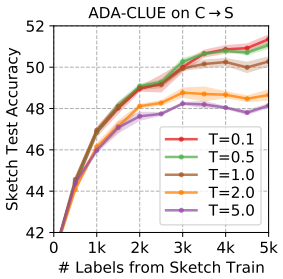


Figure 8: Varying softmax temperature.

We observe that by modulating the temperature of the softmax temperature, we can control this tradeoff. For example, by increasing the temperature, we obtain more diffuse distributions for all points leading to similar uncertainty estimates across points; correspondingly, we expect density to play a bigger role. Similarly, at lower values of temperature we expect uncertainty to have greater influence on the sampling strategy. In Fig. 8 we run a sweep over temperature values and report performance on the Clipart→Sketch shift. As seen, lower values of temperature appear to improve performance, particularly at later rounds (when uncertainty estimates are more reliable). In practice, we do not actually have access to a validation split to tune this hyperparameter and so for our experiments we simply use the default temperature value of 1. We provide this experiment as initial evidence that CLUE may be further tuned for different domain shift difficulties.

7.3 How many active adaptation rounds are optimal?

Given a fixed total budget of 5000 target labels, we now vary the per round budget (and consequently the total number of active adaptation rounds) and report performance on the Clipart→Sketch shift. As seen in Fig. 9, the performance appears fairly robust to the per-round budget across values of 500, 1000, and 2500 but suffers at very small budgets (100). We conjecture that this is possibly due to the large number of classes in DomainNet (345), which cannot be adequately represented at such small budgets.

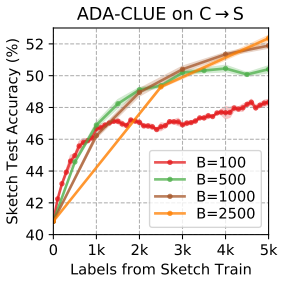
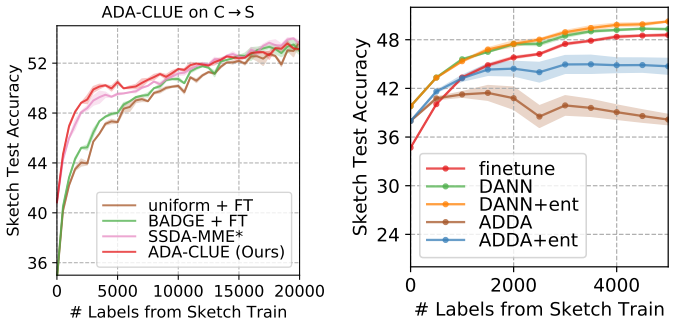


Figure 9: C→S: Varying per-round budget.



(a) Convergence results (b) Role of Entropy Minimization

Figure 10: C→S: Additional `ADA-CLUE` ablations.

7.4 Convergence: When do gains saturate?

Due to the computational expense of running active domain adaptation on multiple shifts with a large CNN (ResNet34 [43]) on a large dataset (DomainNet [24]), in the main paper we restrict ourselves to 10 rounds with a per-round budget of 500. As a check of when performance gains saturate, we benchmark performance on Clipart→Sketch for 40 rounds of with per-round budget of 500 ($= 20k$ labels in total). Results are presented in Fig. 10a. As seen, performance begins to roughly saturate around the 15k labels mark, and performance differences across methods narrow.

7.5 Role of Entropy Minimization

We experiment with augmenting two popular domain-classifier based adaptation methods, `DANN` and `ADDA`, with entropy minimization regularization, as commonly used in semi-supervised learning and in `AADA` [17]. See Fig. 10b. We find that in both cases, this addition consistently improves performance – slightly in the case of a `DANN` and significantly in the case of `ADDA`.

8 Performance of `CLUE` on Standard Active Learning

To study the applicability of our proposed active sampling method to traditional active learning, we benchmark its performance against competing methods on the standard SVHN active learning benchmark. We match the setting in [9], initializing a ResNet18 [43] CNN with random weights and perform 100 rounds of active learning with per-round budget of 100. As summarized in Figure 11, `CLUE`'s performance is on-par with state of the art AL methods, and statistically significantly better than uniform sampling over most rounds.

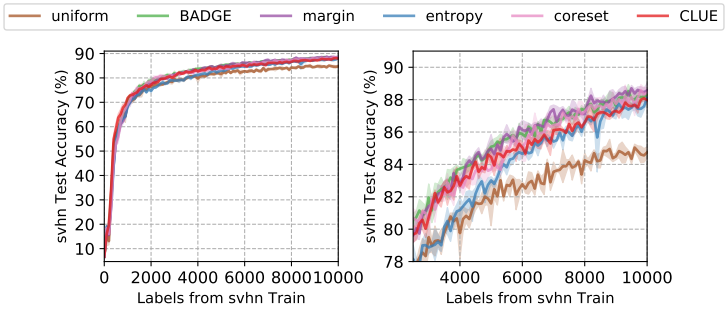


Figure 11: Active Learning performance on SVHN over 100 rounds with per-round budget $= 100$. `CLUE` performs on -par with state-of-the art active learning methods.

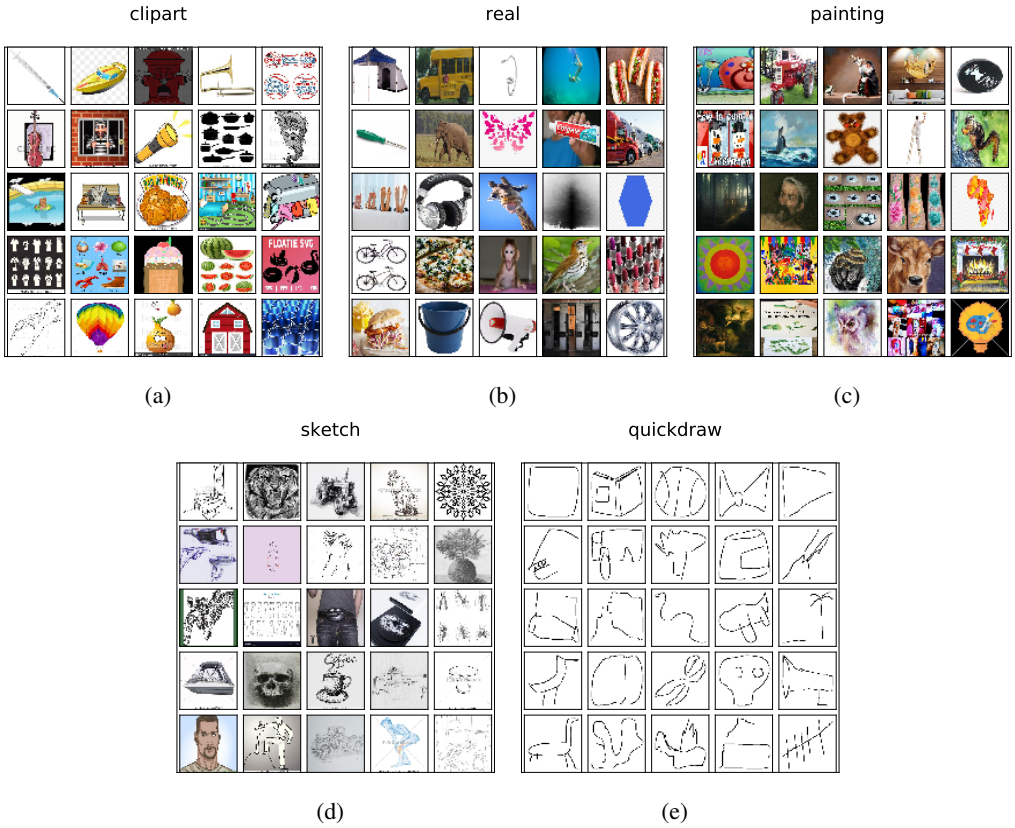


Figure 12: DomainNet [24] qualitative examples

	Real	Clipart	Painting	Sketch	Quickdraw
Train	120906	33525	48212	50416	120750
Test	52041	14604	20916	21850	51750

Table 1: DomainNet [24] train/test statistics

9 Dataset details

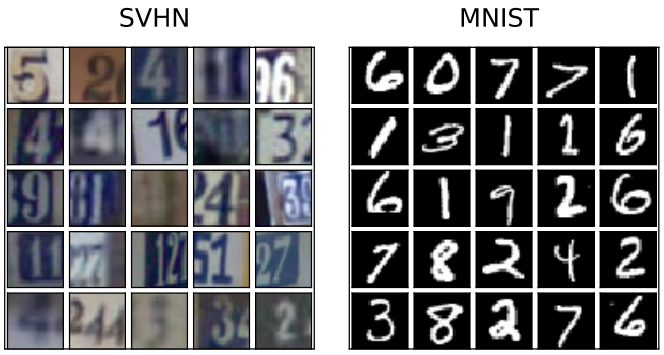
DomainNet. For our primary experiments, we use the DomainNet [24] dataset that consists of 0.6 million images spanning 6 domains, available at http://ai.bu.edu/M3SDA/. For our experiments, we use 4 shifts from 5 domains: Real, Clipart, Sketch, Painting, and Quickdraw. Table 1 summarizes the train/test statistics of each of these domains, while Fig. 12 provides representative examples from each. As models use ImageNet initialization, we avoid using Real as a target domain.

DIGITS. We present results on the SVHN [22]→MNIST [23] domain shift. Both datasets consist images of the digits 0-9. SVHN consists of 99289 (73257 train, 26032 test) RGB images whereas MNIST contains 70k (60k train, 10k test) grayscale images. Fig. 13 shows representative examples.

10 Code and Implementation Details

We use PyTorch [50] for all our experiments. Most experiments were run on an NVIDIA TitanX GPU. All code will be publicly released. We provide code for our DIGITS experiments with this submission as well as an exported notebook for visualization of code/results: Results Notebook

CLUE. We use the weighted K-Means implementation in scikit-learn [51] to implement `CLUE`. Cluster centers are initialized via K-means++ [52]. The implementation uses the Elkan algorithm [53]



(a) (b)

Figure 13: DIGITS qualitative examples

to solve K-Means. For n objects, k clusters, and e iterations ($= 300$ in our experiments), the time complexity of the Elkan algorithm is roughly $\mathcal{O}(nke)$ [54], while its space complexity is $\mathcal{O}(nk)$.

DomainNet experiments. We utilize a ResNet34 [43] CNN architecture. For active adaptation (round 1 and onwards), we use the Adam [55] optimizer with a learning rate of 1e-5, weight decay of 1e-5 and train for 20 epochs per round (with an epoch defined as a complete pass over labeled target data) with a batch size of 64. For unsupervised adaptation (round 0), we use Adam with a learning rate of 3e-7, weight decay of 1e-5, and train for 50 epochs. Across all adaptation methods, we tune loss weights to ensure that the average labeled loss is approximately 10 times as large as the average unsupervised loss. We use random cropping and random horizontal flips for data augmentation. We set loss weights $\lambda_{\mathcal{S}} = 0.1$, $\lambda_{\mathcal{T}} = 1$ and $\lambda_{\mathcal{H}} = 0.1$ (Section 3).

DIGITS experiments. We use the modified LeNet architecture proposed in Hoffman et al. [12] and exactly match the experimental setup in `AADA` [17]. We use the Adam [55] optimizer with a learning rate of 2e-4, weight decay of 1e-5, batch size of 128, and perform 60 epochs of training per-round. We halve the learning rate every 20 epochs. We set loss weights $\lambda_{\mathcal{S}} = 0.1$, $\lambda_{\mathcal{T}} = 1$ and $\lambda_{\mathcal{H}} = 1$ (Section 3). For `AADA`, consistent with the paper we add an entropy minimization objective with a loss weight of 0.1. Images are converted from RGB to grayscale.

10.1 Baseline Implementations

We elaborate on our implementation of the `BADGE` [9] and `AADA` [17] baselines.

`BADGE`. BADGE "gradient embeddings" are computed by taking the gradient of model loss with respect to classifier weights, where the loss is computed as cross-entropy between the model's predictive distribution and its most confidently predicted class. Next, K-Means++ [52] is run on these embeddings to yield a batch of samples.

`AADA`. In `AADA`, a domain discriminator G_d is learned to distinguish between source and target features obtained from an extractor G_f, in addition to a task classifier G_y. For active sampling, points are scored via the following importance weighting-based acquisition function ($\mathcal{H}$ denotes model entropy): $s(x) = \frac{1-G_d(G_f(x))}{G_d^*(G_f(x))}\mathcal{H}\left(G_y\left(G_f(x)\right)\right)$, and top B instances are selected for labeling. In practice, to generate less redundant batches we randomly sample B instances from the top-2% scores, as recommended by the authors. Consistent with the original work, we also add an entropy minimization objective with a loss weight of 0.01.

11 Understanding `ADA-CLUE` with Qualitative Examples

In this section, we attempt to get a sense of the behavior of `ADA-CLUE` versus other methods via visualizations and qualitative examples on the SVHN→MNIST shift. Fig. 14 shows confusion matrices of model predictions before (*left*) and after (*right*) performing unsupervised adaptation (via

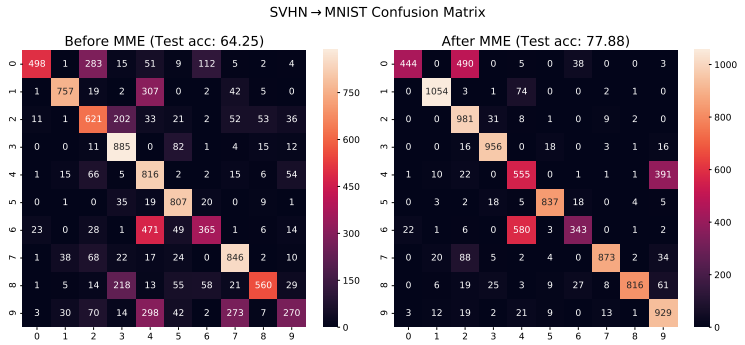
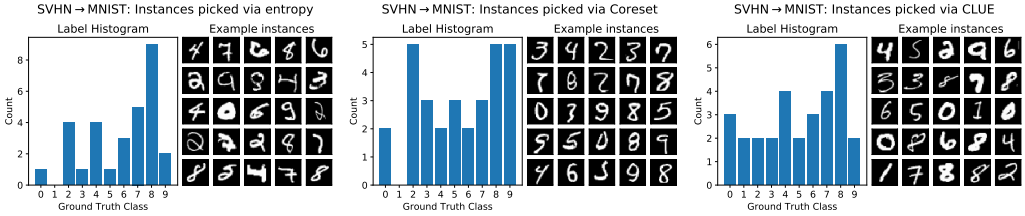
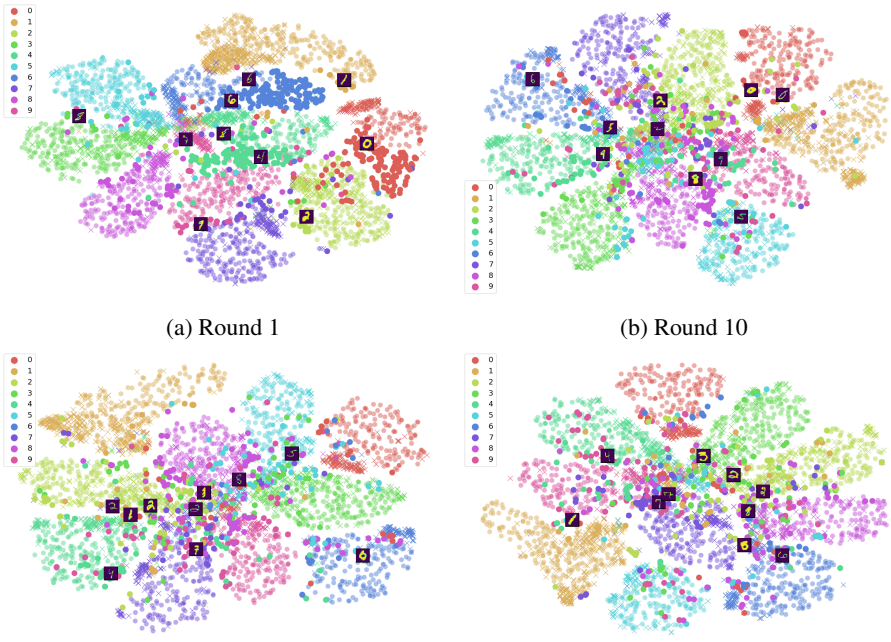


Figure 14: SVHN→MNIST: Confusion matrix of model predictions before and after MME at round 0.



(a) (b) (c)

Figure 15: SVHN→MNIST: Label histograms and examples of instances selected by entropy, coreset, and CLUE at Round 1 with $B = 30$.



(a) Round 1 (b) Round 10

(c) Round 20 (d) Round 30

Figure 16: SVHN→MNIST: TSNE visualization of feature space and instances picked by CLUE at rounds 1, 10, 20, and 30. Circles denote target points and crosses denote source points.

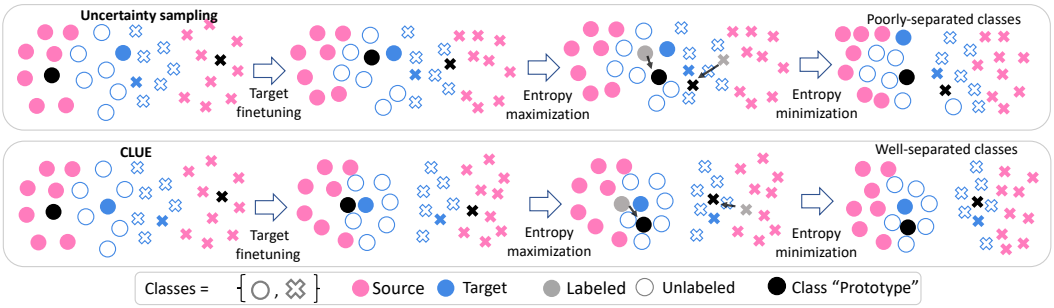


Figure 17: Coupling between MME and uncertainty sampling versus CLUE.

MME) at round 0. As seen, MME aligns some classes (eg. 1's and 9's) remarkably well even without access to target labels. However, large misalignments remain for some other classes (0, 4, and 6).

Visualizing selected points. In Fig. 15, we visualize instances selected by three strategies at Round 0 – entropy, coreset, and CLUE, with $B = 30$. We visualize the ground truth label distribution of the selected instances, as well as qualitative examples. As seen, strategies vary across methods. Entropy tends to pick a large number of 8's, and selects high-entropy examples that (on average) appear challenging even to humans. Coreset tends to have a wider spread over classes. CLUE appears to interpolate between the behavior of these two methods, selecting a large number of 8's (like entropy) but also managing to sample atleast a few instances from every class (like coreset).

TSNE viz over rounds. In Fig. 16, we illustrate the sampling behavior of CLUE over rounds via TSNE [48] visualizations. We follow the same conventions as Fig.6 of the main paper, and visualize the logits of a subset of incorrect (large, opaque circles) and correct (partly transparent circles) model predictions on the target domain, along with instances sampled via CLUE. We oversample incorrect target predictions to emphasize regions of the feature space on which the model currently underperforms. Across all four stages, we find that ADA-CLUE samples instances that are uncertain (often present in a cluster of incorrectly classified instances), representative (spanning the entire feature space), and diverse (dissimilar from one another). This behavior is seen even at later rounds when classes appear better separated.

CLUE and MME. While our results demonstrate CLUE and MME to work very well in conjunction with one another, we seek to explain why. One reason already we note is MME's target entropy minimization objective, which has the effect of producing a feature space where similar points are more tightly clustered and makes it conductive to sample diverse instances via CLUE.

In Fig. 17, we provide an additional hypothesis. As described previously, MME incorporates a labeled ground truth cross-entropy loss, and an unlabeled minimax entropy loss. Consistent with [15], we can consider each column of the classifier matrix to be a "class prototype" (denoted as a black circle). We conjecture that finetuning on instances on the decision boundary that acquired purely based on uncertainty (*top* panel), followed by adversarial entropy minimization, can lead to learning ambiguous decision boundaries. On the other hand, CLUE incorporates both uncertain and representative points to select instances, which is more conducive to adversarial entropy optimization and leads to better separated classes (*bottom* panel).

12 Extended Description of the CLUE Objective

We describe in more detail, the CLUE objective presented (Eq. 2) in the main paper. Recall that we identify instances that are representative of the unlabeled target data distribution based on similarity of instances. Considering the L2 distance in the CNN representation space $\phi(\cdot)$ as a dissimilarity measure, we quantify the dissimilarity between instances in a set X_k in terms of its variance $\sigma^2(X_k)$

given by [37]:

$$\begin{aligned}\sigma^2(X_k) &= \frac{1}{2|X_k|^2}\sum_{\mathbf{x_i},\mathbf{x_j}\in X_k}||\phi(\mathbf{x_i})-\phi(\mathbf{x_j})||^2 \\ &= \frac{1}{|X_k|}\sum_{\mathbf{x}\in X_k}||\phi(\mathbf{x})-\mu_\mathbf{k}||^2 \\ \text{where}\quad \mu_\mathbf{k} &= \frac{1}{|X_k|}\sum_{\mathbf{x}\in X_k}\phi(\mathbf{x})\end{aligned} \tag{5}$$

A small $\sigma^2(X_k)$ indicates that a set X_k contains instances that are similar to one other. Our goal is to identify sets of instances that are representative of the unlabeled target set, by partitioning the unlabeled target data into K sets, each with small $\sigma^2(X_k)$. Formulating this as a set-partitioning problem, where $\mathcal{S} = \{X_1, X_2, ..., X_K\}$, we minimize the sum of variance over all sets:

$$\underset{\mathcal{S}}{\operatorname{argmin}}\sum_{k=1}^{N}\sigma^2(X_k) \tag{6}$$

where $\sigma^2(X_k)$ is defined in Eq. 5.

To ensure that the more informative/uncertain instances play a larger role in identifying representative instances, we employ weighted-variance, where an instance is weighted by its informativeness. The weighted variance $\sigma^2_\mathcal{H}(X_k)$ of a set of instances is given by [38]:

$$\sigma^2_\mathcal{H}(X_k) = \frac{1}{\sum_{\mathbf{x_i}\in X_k}h_i}\sum_{\mathbf{x_i}\in X_k}h_i||\phi(\mathbf{x_i})-\mu_\mathbf{k}||^2 \quad \text{where} \quad \mu_\mathbf{k} = \frac{1}{\sum h_i}\sum_{\mathbf{x_i}\in X_k}h_i\phi(\mathbf{x_i}) \tag{7}$$

where h_i is the scalar weight corresponding to the instance $\mathbf{x_i}$.

Considering the informativeness (weight) of an instance to be its uncertainty under the model, given by $\mathcal{H}(Y|\mathbf{x})$ (defined in Eq. 1 in main paper), we rewrite the set-partitioning objective in Eq. 6 to minimize sum of weighted variance of a set (from Eq. 7):

$$\begin{aligned}\underset{\mathcal{S}}{\operatorname{argmin}}\sum_{k=1}^{N}\sigma^2_\mathcal{H}(X_k) &= \underset{\mathcal{S}}{\operatorname{argmin}}\sum_{k=1}^{N}\frac{1}{\sum_{x\in X_k}\mathcal{H}(Y|\mathbf{x})}\sum_{\mathbf{x}\in X_k}\mathcal{H}(Y|\mathbf{x})||\phi(\mathbf{x})-\mu_\mathbf{k}||^2 \\ \text{where}\quad \mu_\mathbf{k} &= \frac{1}{\sum_{x\in X_k}\mathcal{H}(Y|\mathbf{x})}\sum_{\mathbf{x}\in X_k}\mathcal{H}(Y|\mathbf{x})\phi(\mathbf{x})\end{aligned} \tag{8}$$

This, gives us the overall set-partitioning objective for CLUE (Eq. 2 in main paper).

13 Future Work

Our work suggests a few promising directions of future work. First, one could experiment with alternative uncertainty measures in `ADA-CLUE` instead of model entropy, including those (such as uncertainty from deep ensembles) that have been shown to be more reliable under a dataset shift [21]. Further, one could incorporate specialized model architectures from few-shot learning [56, 15] to deal with the label sparsity in the target domain. Finally, while we restrict our task to image classification in this paper, it is important to also study active domain adaptation in the context of tasks like object detection and semantic segmentation.