Supplementary Material
S^{3}VAADA: Submodular Subset Selection for Virtual Adversarial Active Domain Adaptation

Harsh Rangwani Arihant Jain* Sumukh K Aithal* R. Venkatesh Babu
Video Analytics Lab, Indian Institute of Science, Bengaluru, India
harshr@iisc.ac.in, arihantjain@iisc.ac.in, sumukhaithal6@gmail.com, venky@iisc.ac.in

Supplementary Video

We encourage the readers to go through the accompanying video which illustrates the overview of the different steps in the proposed S^{3}VAADA method.

Organization of Supplementary Document

1. t-SNE Analysis for AADA 1

2. Proofs 2
- 2.1. Lemma 1 . . . 2
- 2.2. Lemma 2 . . . 3
- 2.3. Theorem 1 . . . 3

3. Insight for Diversity Score 3

4. Additional Analysis for S^{3}VAADA 3
- 4.1. Budget Ablation . . . 3
- 4.2. Convergence: When does the Active DA performance stop improving? . . . 4

5. Analysis of VAADA training 4
- 5.1. Analysis of Learning Rate . . . 4
- 5.2. Analysis of using Gradient Clipping . . . 4
- 5.3. Comparison of VADA with VAADA . . . 5
- 5.4. Visualizing clusters using t-SNE . . . 5
- 5.5. Hyper-Parameter Sensitivity of VAADA . . . 5

6. Implementation Details 5
- 6.1. Configuration for DANN . . . 5
- 6.2. Configuration for SSDA (MME*) . . . 5
- 6.3. Configuration for VAADA . . . 5

7. Comparison with JO-TAL 7

8. Comparison with Alternate Adversarial Perturbation based sampling 7

9. Description of Datasets Used 7

10DomainNet Experiments 8

11Future Extension to Other Applications 8

*Equal Contribution

1. t-SNE Analysis for AADA

We give experimental evidence of the redundancy issue present in the AADA sampling. We perform the training with VAADA training method with the implementation details present in Sec. 6 on Webcam → Amazon. Fig. 1 shows the selected samples in the intermediate cycle, which clearly depicts clusters of the samples selected. The existence of clusters confirms the presence of *redundancy* in selection.

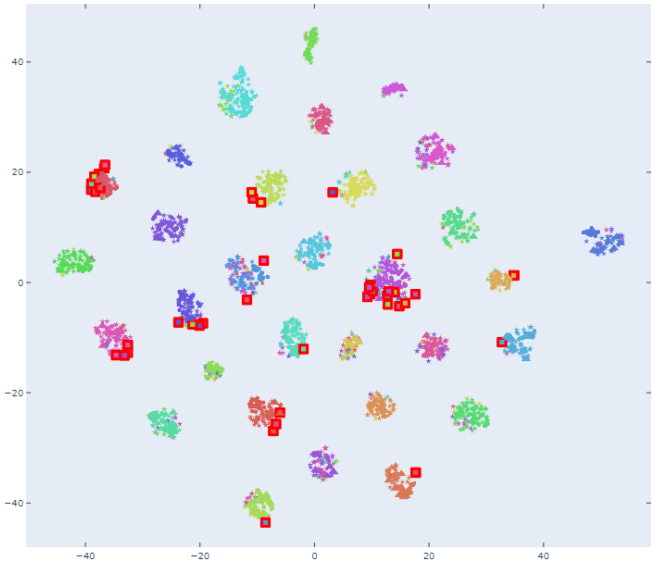


Figure 1. t-SNE analysis of AADA sampling. The selected samples are represented by the red boxes. We see clusters of samples being selected which depict *redundancy* in selection.

2. Proofs

2.1. Lemma 1

We present proof for lemma 1 which is stated in Section 4.1.4 of the main paper.

Lemma 1 The set function $f(S)$ defined by equation below is submodular.

$$f(S \cup \{x_i\}) - f(S) = \alpha VAP(x_i) + \beta d(S, x_i) + (1 - \alpha - \beta) R(S, x_i)$$

We first prove that all the three individual components in the above expression are submodular and then prove that the convex combination of the three terms is submodular.

Submodularity of the VAP Score $VAP(x_i)$: The gain value of the VAP score is given as the following below:

$$f(S \cup \{x_i\}) - f(S) = VAP(x_i)$$

We give below the proof for the submodularity which is based on the *diminishing returns* property as stated in Sec. 3.1, in the main text.

Proof. For two sets S_1, S_2 such that $S_1 \subseteq S_2$ and $x_i \in \Omega \backslash S_2$, if the function is submodular it should satisfy the following property in Sec 3.1.

$$f(S_1 \cup \{x_i\}) - f(S_1) \geq f(S_2 \cup \{x_i\}) - f(S_2)$$
$$VAP(x_i) \geq VAP(x_i)$$

As the left hand side is equal to right hand side, the inequality is satisfied, hence the VAP score function is submodular. □

Submodularity of Diversity Score $d(S, x_i)$: The gain in value for the diversity function is given as:

$$f(S \cup \{x_i\}) - f(S) = \min_{x \in S} D(x, x_i)$$

We provide the proof that the above gain function corresponds to a submodular function $f(S)$ below:

Proof. For two sets S_1, S_2 such that $S_1 \subseteq S_2$ and $x_i \in \Omega \backslash S_2$, if the function is submodular it should satisfy the following property in Sec 3.1:

$$f(S_1 \cup \{x_i\}) - f(S_1) \geq f(S_2 \cup \{x_i\}) - f(S_2)$$
$$min_{x \in S_1} D(x, x_i) \geq min_{x \in S_2} D(x, x_i)$$

$D(x, x_i) \geq 0$ for every x and x_i as it is a divergence function. As S_2 contains more elements than S_1, the minimum of $D(x, x_i)$ will be less then for S_2 in comparison to that of S_1. Hence the final inequality is satisfied which shows that $f(S)$ is submodular. □

Submodularity of Representativeness Score $R(S, x_i)$: We first prove one property which we will use for analysis of Representativeness Score.

***Property**: The sum of two submodular set functions $f(S) = f_1(S) + f_2(S)$, is submodular.*

Proof. Let A and B be any two random sets.

$$\begin{aligned} f(A) + f(B) &= f_1(A) + f_2(A) + f_1(B) + f_2(B) \\ &\geq f_1(A \cup B) + f_2(A \cup B) + f_1(A \cap B) + \\ &\quad f_2(A \cap B) \\ &= f(A \cup B) + f(A \cap B) \end{aligned}$$

Hence the sum of the two submodular functions is also submodular. The result can be generalized to a sum of arbitrary number of submodular functions. □

The representativeness score can be seen as the following set function below:

$$f(S) = \sum_{x_i \in \mathcal{D}_u} \max_{x_j \in S} s_{ij}$$

We calculate the gain for each sample through this function which is equal to $R(S, x_i)$:

$$f(S \cup \{x_i\}) - f(S) = \sum_{x_k \in \mathcal{D}_u} \max_{x_j \in S \cup \{x_i\}} s_{kj} - \sum_{x_k \in \mathcal{D}_u} \max_{x_j \in S} s_{kj}$$
$$R(S, x_i) = \sum_{x_k \in \mathcal{D}_u} \max(s_{ik} - \max_{x_j \in S} s_{kj}, 0)$$

***Property**: The set function defined below is submodular:*

$$f(S) = \sum_{x_i \in \mathcal{D}_u} \max_{x_j \in S} s_{ij}$$

Proof. We first show that the function $f_i(S) = \max_{x_j \in S} s_{ij}$ is submodular. We first use the property, $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ where A, B are two sets, sufficient to show that $f(S)$ is submodular:

$$f_i(A) + f_i(B) \geq f_i(A \cup B) + f_i(A \cap B) \quad (1)$$
$$\max_{x_j \in A} s_{ij} + \max_{x_j \in B} s_{ij} \geq \max_{x_j \in A \cup B} s_{ij} + \max_{x_j \in A \cap B} s_{ij} \quad (2)$$

which follows due to the following:

$$\max(\max_{x_j \in A} s_{ij}, \max_{x_j \in B} s_{ij}) = \max_{x_j \in A \cup B} s_{ij}$$

and

$$\min(\max_{x_j \in A} s_{ij}, \max_{x_j \in B} s_{ij}) \geq \max_{x_j \in A \cap B} s_{ij}$$

As $f_i(S)$ is submodular, the $f(S)$ can be seen as:

$$f(S) = \sum_{x_i \in \mathcal{D}_u} f_i(S)$$

which is submodular according to the property that sum of submodular functions is also submodular proved above. □

Combining the Submodular Functions: We use the property that a convex combination of the submodular functions is also submodular. Hence our sampling function which is the convex combination given by:

$$f(S \cup \{x_i\}) - f(S) = \alpha VAP(x_i) + \beta d(S, x_i) + (1 - \alpha - \beta) R(S, x_i)$$

Also follows the property of submodularity.

2.2. Lemma 2

Here we present proof of lemma 2 stated in Sec. 4.1.4 of main paper.

Lemma 2 The set function f(S) defined by equation below is a non-decreasing, monotone function:

$$f(S \cup \{x_i\}) - f(S) = \alpha VAP(x_i) + \beta d(S, x_i) + (1 - \alpha - \beta) R(S, x_i)$$

Proof. For the function to be non-decreasing monotone for every set S the addition of a new element should increase value of $f(S)$. The gain function for $f(S)$ is given below:

$$f(S \cup \{x_i\}) - f(S) \geq 0$$
$$\alpha VAP(x_i) + \beta d(S, x_i) + (1 - \alpha - \beta) R(S, x_i) \geq 0$$

As the $VAP(x_i)$ and $d(S, x_i)$ are KL-Divergence terms, they have value ≥ 0. The third term $R(S, x_i) = \sum_{x_k \in \mathcal{D}_u} \max(s_{ik} - \max_{x_j \in S} s_{kj}, 0)$ is also ≥ 0. As $0 \leq \alpha, \beta, \alpha + \beta \leq 1$, the value of gain is positive, this shows that the function $f(S)$ is a non-decreasing monotone.

□

2.3. Theorem 1

Theorem 1: Let S^* be the optimal set that maximizes the objective in Eq. 3 then the solution S found by the greedy algorithm has the following approximation guarantee:

$$f(S) \geq \left(1 - \frac{1}{e}\right) f(S^*) \tag{3}$$

Proof: As $f(S)$ is submodular according to Lemma 1 and is also non decreasing, monotone according to Lemma 2. Hence the approximation result directly follows from Theorem 4.3 in [8]. The approximation result shows that the algorithm is guaranteed to get at least 63% of the score of the optimal function $f(S^*)$. However, in practice, this algorithm is often able to achieve 98% of the optimal value in certain applications [5]. As it's a worst case result in practice we get better performance than the worst case.

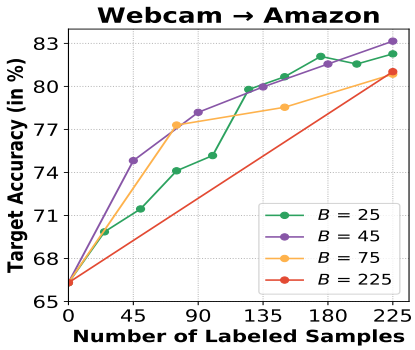


Figure 2. Analysis of S^{3}VAADA for different budget sizes on Webcam → Amazon shift of Office-31 dataset.

3. Insight for Diversity Score

When the $\alpha = 0$ and $\beta = 1$ the gain function $f(S \cup \{x_i\}) - f(S)$ is just $min_{x \in S} D(x, x_i)$. The greedy algorithm described for sampling in Algorithm 1 in main paper, leads to following objective for selecting sample x^*.

$$x^* = \underset{x_i \in \mathcal{D}_u \setminus S}{argmax} \min_{x \in S} D(x, x_i)$$

This objective exactly resembles the K-Center Greedy method objective which is used by Core-Set method [13] and is shown to select samples which cover the entire dataset. The K-Center Greedy method is very effective in practice. This connection shows that diversity component in our framework also tries to cover the dataset as done by Core-Set [14] method which is one of the very effective diversity based active learning method.

4. Additional Analysis for S^{3}VAADA

In this sections we provide additional experiments for analysis of the proposed S^{3}VAADA. Unless specified, we run the experiments with single random seed and report the performance. In case the performance difference is small, we provide average results of three runs with different random seeds.

4.1. Budget Ablation

Keeping in mind the practical constraint of only having a small amount of labeling budget in the target domain, we

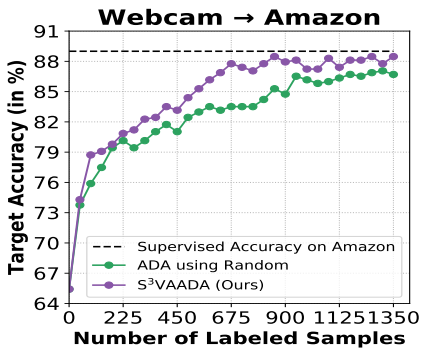


Figure 3. Active DA performance on Webcam → Amazon for 30 cycles. We find that the performance converges to supervised learning performance after around 15 cycles.

restrict ourselves to having a budget size of 2% of the labeled target data. Due to different size of target data in each dataset, the sampling algorithm needs to work robustly under different budget scenario's. For further analysis, we provide results on Webcam to Amazon with different budget sizes B for sampling in Fig. 2. We find that S^{3}VAADA is quite robust for budget sizes greater than 45. We find that small budget of 25 results in more stochasticity in the results.

4.2. Convergence: When does the Active DA performance stop improving?

In all the experiments, we have used a budget of 2% for 5 rounds which corresponds to 10% of the target dataset. We find that the performance of algorithm improves in majority of cycles. This brings up the question, *When does the performance of the model stop improving even after adding more labeled samples?*. For answering this question, we perform experiments on Webcam → Amazon and perform active DA for 30 rounds. Fig. 3 shows the results on Webcam → Amazon with S^{3}VAADA and Random sampling. It can be seen that after around 15 cycles, the gains due to additional samples being added decrease significantly and the performance seems to converge. The performance of the proposed S^{3}VAADA is much better than random sampling in all the rounds. It must also be noted that S^{3}VAADA reaches an accuracy of 89% with 20 rounds (40% of the dataset) which is equal to the performance when trained on all the target data.

5. Analysis of VAADA training

We propose VAADA method which is an enhanced version of VADA, suitable for Active DA. We find that proposed improvements in VAADA have a significant effect on the final active DA performance, which we analyse in detail in the following sections. We have done all our analysis using source dataset as Webcam and target dataset as Amazon which is a part of Office-31.

5.1. Analysis of Learning Rate

It is a common practice [4, 6] in domain adaptation (DA) to use a relatively lower learning rate (usually decreased by a multiplying a factor of 0.1) for convolutional backbone which is ResNet-50 in our case. We find that though this practice helps for Unsupervised DA performance, it was not useful in the case of Active DA. In Fig. 4, we show the comparison of using same learning rate for the backbone network (as proposed in VAADA), to using a smaller learning rate for backbone. The results clearly show that not lowering the learning is specially helpful for Active DA, whereas it is not for Unsupervised DA.

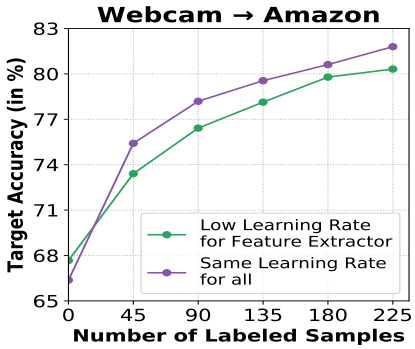


Figure 4. Comparison between Active DA with lower learning rate and a higher learning rate for backbone. The results are the average across three runs with different random seeds.

5.2. Analysis of using Gradient Clipping

In the original implementation of VADA [15] the authors use the method of Exponential Moving Average (EMA) (also known as Polyak Averaging [11]) of model weights, which increases the stability of results. In place of EMA, we find that using proposed Gradient Clipping in VAADA works better for stabilizing the training. In Gradient Clipping, we scale the gradients such that the gradient vector norm has magnitude 1. We find that Gradient Clipping allows the network to train stably, with a relatively high learning rate of 0.01. For showcasing the stabilising effect of Gradient Clipping, in Fig. 5 we compare the performance of the model with and without gradient clipping. We find that Gradient Clipping leads to a increase of accuracy of above 10% for each active learning cycle, with achieving stable increase in performance with the addition of more labels. On the other hand the model without clipping is unable to produce stable increase in performance with addition of labels.

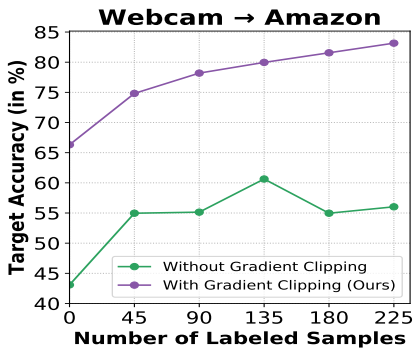


Figure 5. Ablating Gradient clipping on VAADA

5.3. Comparison of VADA with VAADA

In this section we provide additional implementation details and analysis, continuing from Sec. 6 of main paper. The comparison shown with the VADA method corresponds to the original VADA configuration specified in [15]. In the original implementation, the authors propose to use Adam optimizer and EMA for training. We use Adam with learning rate of 0.0001 and use the exact same settings as in [15]. It can be seen in Fig. 6 that VAADA consistently outperforms the VADA training in Active DA for CoreSet and S^{3}VAADA as well.

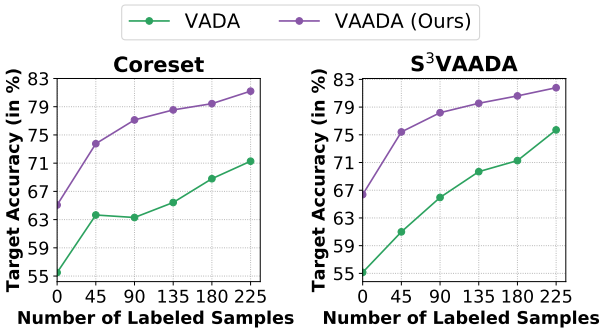


Figure 6. We show the comparison of VAADA and VADA. We see consistent improvement of VAADA over VADA across all cycles.

5.4. Visualizing clusters using t-SNE

In this section, we analyse the t-SNE plot (Fig. 7) of the two different training methods i.e., DANN and VAADA. We find that in VAADA training, there is formation of distinct clusters and also the cluster sizes are similar. Whereas in DANN t-SNE, there is no formation of distinct clusters, and a large portion of sample are clustered in between. This shows that additional losses of conditional entropy and smoothing through Virtual Adversarial Perturbation loss are necessary to enforce the cluster assumption.

5.5. Hyper-Parameter Sensitivity of VAADA

We used the same λ values mentioned as a robust choice by VADA [15] authors, for VAADA training, setting $\lambda_d = 0.01$, $\lambda_s = 1$ and $\lambda_t = 0.01$ across all datasets. For analysing the sensitivity of the performance of VAADA across different hyper-parameter choices, we provide results with varying λ parameters in Fig. 8. We also find that the robust choice recommended for VADA, also works the best for VAADA. Hence, this *fixed-set* of robust λ parameters can be used across datasets with varying degree of domain shifts. This is also enforced by the fact, that in all our experiments these *fixed* hyperparameters were able to achieve state-of-the-art performance across datasets. This decreases the need for hyper-parameter tuning specific to each dataset.

6. Implementation Details

6.1. Configuration for DANN

For the DANN experiments, we use a batch size of 36 with a learning rate of 0.01 for all the linear layers. We use a smaller learning rate of 0.001 for the ResNet-50 backbone. DANN is trained with SGD with a momentum of 0.9 and weight decay value of 0.0005 following the schedule described in [4]. The model architecture and hyperparameters are same as in [6]. The model is trained for 10,000 iterations as done in [6] and the best validation accuracy is reported in the graphs.

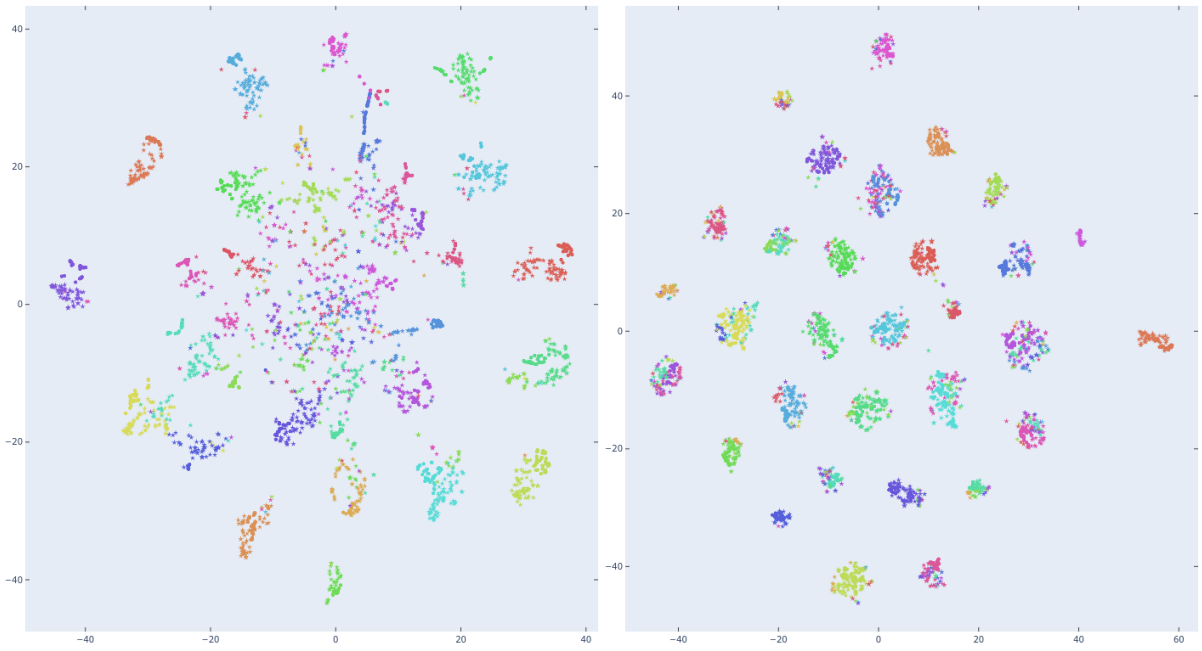
6.2. Configuration for SSDA (MME*)

We use the author's implementation[1] for experiments on Office dataset. We used ResNet-50 as backbone and used same parameters as used in their implementation. For Active DA, we initially train the model with no labeled target data and keeps on adding 2% of the unlabeled target data to labeled target set for 5 cycles. We train the model for 20,000 iterations. A similar procedure of reporting the best validation accuracy on the fixed validation set, as done for other baselines is followed.

6.3. Configuration for VAADA

The model is trained with a batch size of 16 and a learning rate of 0.01 for all the layers using the SGD Optimizer with a momentum of 0.9. A weight decay of 0.0005 was used. The model is trained for 100 epochs and the best accuracy is reported in the graphs. A ResNet-50 backbone is used with pretrained ImageNet weights. The architecture for various model components used are shown in Table 1 and 2. Same architecture is used for all experiments in the paper.

[1] https://github.com/VisionLearningGroup/SSDA_MME



(a) DANN (b) VAADA

Figure 7. Visualization of clusters of data points formed by DANN and VAADA on DA for Webcam → Amazon. Different colors represent different classes. It can seen that VAADA forms much distinct clusters data than DANN.

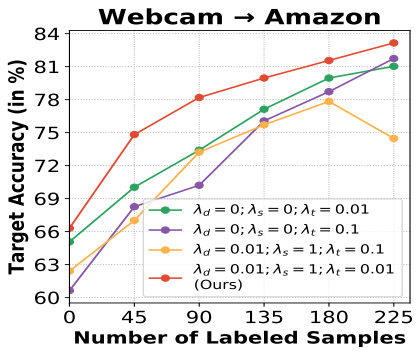


Figure 8. Different Hyperparameters on Webcam → Amazon dataset.

Layer/Component	Output Shape
-	$224 \times 224 \times 3$
ResNet-50	2048
Linear	256

Table 1. **Feature Generation** g_θ: Architecture used for generating the features

The above hyper parameters are used for all our experiments on Office-Home and Office-31 datasets. We just change the batch size to 128 and use the learning rate decay schedule of DANN for experiments on VisDA-18 dataset.

Layer	Output Shape
Feature Classifier (f_θ)	
-	256
Linear	C
Domain Classifier (D_ϕ)	
-	256
Linear	1024
ReLU	1024
Linear	1024
ReLU	1024
Linear	2

Table 2. Architecture used for feature classifier and Domain classifier. C is the number of classes. Both classifiers will take input from feature generator (g_θ).

$$L(\theta; \mathcal{D}_s, \mathcal{D}_t, \mathcal{D}_u) = L_y(\theta; \mathcal{D}_s, \mathcal{D}_t) + \lambda_d L_d(\theta; \mathcal{D}_s, \mathcal{D}_t, \mathcal{D}_u) + \lambda_s L_v(\theta; \mathcal{D}_s \cup \mathcal{D}_t) + \lambda_t (L_v(\theta; \mathcal{D}_u) + L_c(\theta; \mathcal{D}_u))$$

The ϵ used in Eq. 3 and 4 in the main paper refer to the maximum norm of the virtual adversarial perturbation, was set it to 5 in our experiments. The value of the number of random restarts (N) to generate virtual adversarial perturbation for the proposed sampling is set to 5. The α value is set to 0.5 and β value is set to 0.3 across all experiments. We use Gradient Clipping to clip the norm of the gradient

vector to 1 to stabilize and accelerate VAADA.
We used Weights & Biases [1] to track our experiments.

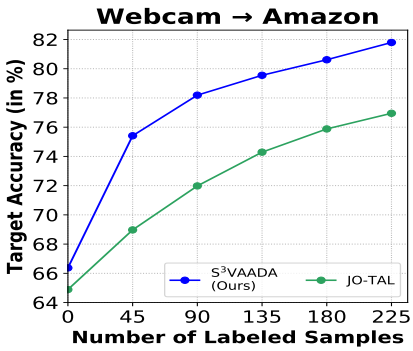


Figure 9. S^{3}VAADA vs. JO-TAL

7. Comparison with JO-TAL

We compare our results with JO-TAL by Chattopadhyay et al. [2] (by implementing in `cvxopt`). JO-TAL performs both active learning and domain adaptation in a single step. Since, JO-TAL was not proposed in the context of deep learning, we use deep features from ImageNet pretraind model and train an SVM classifier on top of them. The optimization problem was implemented in `cvxopt`. Fig. 9 shows S^{3}VAADA achieves significant performance gains across cycles when compared to JO-TAL.

8. Comparison with Alternate Adversarial Perturbation based sampling

There also exists a sampling method [3] based on DeepFool adversarial perturbations [7] Active Learning (DFAL) but due to its higher complexity and computation time, it was unfeasible for us to use it as a baseline for all experiments. We provide the comparison of DFAL with S^{3}VAADA in terms of accuracy on Active DA from Webcam $\rightarrow$ Amazon in Fig. 10. Training is done through VAADA for both sampling methods. We find that S^{3}VAADA significantly outperforms DFAL sampling achieving better results in all cycles.

9. Description of Datasets Used

Office-31 [12]: It has images from 3 domains i.e., Webcam, DSLR and Amazon, belonging to 31 classes.

Office-Home [16]: This dataset has a more severe domain shift across domains compared to Office-31. It is a 65 class dataset and contains images from 4 domains namely, Art, Clipart, Product and Real World.

VisDA-18 [10]: This dataset consist of images from synthetic and real domains. The dataset has annotations for

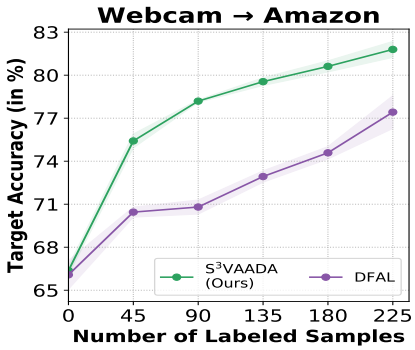
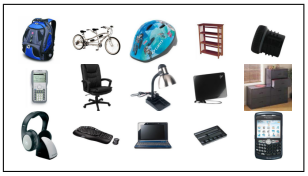


Figure 10. S^{3}VAADA outperforms DFAL in all the cycles, even though both attain same initial accuracy. It shows that S^{3}VAADA selects much more informative samples compared to DFAL.

(a) Webcam (b) DSLR

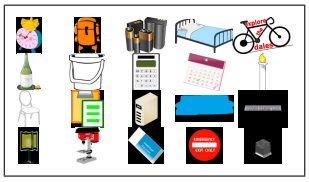


(c) Amazon

Figure 11. Some Office-31 Dataset examples

(a) Art

(b) Product



(c) Clipart

(d) Real World

Figure 12. Some Office-Home Dataset examples

for two tasks: image classification and image segmentation. We used dataset of image classification task. It has 12 different object categories.

Some example images of each dataset are shown in Figs. 11, 12 and 13.

(a) Real (b) Synthetic

Figure 13. Some Visual DA (VisDA-18) Dataset examples

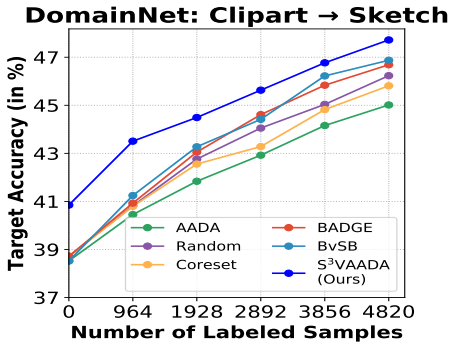


Figure 14. Active Domain Adaptation results on Clipart → Sketch dataset. This shows the proposed method is scalable to larger datasets.

10. DomainNet Experiments

DomainNet [9] consists of about 0.6 million images belonging to 345 classes. The images belong to 5 domains: Clipart, Sketch, Quickdraw, Painting, Real. For showing the scalability of our method, we use Clipart as the source and Sketch as the target domain. The Clipart domain consists of 33,525 images in the train set and 14,604 images in the test set. The Sketch domain consists of 50,416 images in the train set and 21,850 images in the test set. Due to computational limitations we are not able to provide results on different possible domains.

For the DomainNet experiments, we use a batch size of 36 with a learning rate scheduler same as DANN and run each baseline for 30 epochs. We use Gradient Clipping and clip the norm to 10. In this case we find that performance of S^{3}VAADA does not stagnate at 30 epochs but due to limitations of compute we only train for 30 epochs. Hence, there exist scope for improvement in results with parameter tuning and more computational budget.

Fig. 14 shows the results on Clipart → Sketch domain shift. The performance of S^{3}VAADA outperforms all the other techniques in all the cycles. This shows that the efficacy of the proposed method on a large dataset containing 345 classes.

11. Future Extension to Other Applications

The S^{3}VAADA technique is based on the idea of cluster assumption i.e., aligning of clusters of different classes, which is used in the sampling method. Some recent DA techniques for Object Detection [18] and Image Segmentation [17] which aim for classwise alignment of features, can be seen as methods which satisfy the cluster assumption. Hence, we hope that combining such techniques with our method can yield good Active DA techniques tailored for these specific applications. In the current work, we focused on diverse image classification tasks, leaving these applications for future work.

References

[1] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com. 7

[2] Rita Chattopadhyay, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Joint transfer and batch-mode active learning. In *International conference on machine learning*, pages 253–261. PMLR, 2013. 7

[3] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018. 7

[4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 4, 5

[5] Andreas Krause and Carlos Guestrin. Optimizing sensing: From water to the web. *Computer*, 42(8):38–45, 2009. 3

[6] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1645–1655, 2018. 4, 5

[7] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016. 7

[8] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978. 3

[9] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 8

[10] X. Peng, B. Usman, N. Kaushik, D. Wang, J. Hoffman, and K. Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2102–21025, 2018. 7

[11] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 4

[12] 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. 7

[13] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. 3

[14] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2110–2118, 2016. 3

[15] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018. 4, 5

[16] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7

[17] Haoran Wang, Tong Shen, Wei Zhang, Ling-Yu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *European Conference on Computer Vision*, pages 642–659. Springer, 2020. 8

[18] Minghao Xu, Hang Wang, Bingbing Ni, Qi Tian, and Wenjun Zhang. Cross-domain detection via graph-induced prototype alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12355–12364, 2020. 8