Supplementary Materials

A. Proof of Proposition 1

Proof. Since the linear classifier is supervised via the margin loss for each training sample (x, y)

$$\mathcal{L}_m(x,y) = \sum_{i \neq y} [m - c(g(x))_y + c(g(x))_i]_+ \tag{1}$$

We can rewrite the loss for binary classification in a batch-wise representation as

$$\begin{aligned} \mathcal{L} &= \sum_{(x,y)\in\mathcal{S}} \mathcal{L}_m(x,y) \\ &= \sum_{x_p \in \mathcal{S}^+} \left[m - w_+^T x_p + w_-^T x_p\right]_+ \\ &+ \sum_{x_n \in \mathcal{S}^-} \left[m + w_+^T x_n - w_-^T x_n\right]_+ \end{aligned} \tag{2}$$

after one-step of gradient descending, the newly updated classifiers are represented as

$$w'_+ = w_+ - \eta \bigtriangledown_{w_+} \mathcal{L} \quad w'_- = w_- - \eta \bigtriangledown_{w_-} \mathcal{L} \tag{3}$$

Where $\eta > 0$ is the learning rate of gradient descending and the gradient w.r.t weight parameter is in the form as

$$\begin{aligned} \bigtriangledown_{w_+} \mathcal{L} &= \sum_{x_n \in \mathcal{S}^-} \delta(m > w_-^T x_n - w_+^T x_n) x_n \\ &- \sum_{x_p \in \mathcal{S}^+} \delta(m > w_+^T x_p - w_-^T x_p) x_p \end{aligned} \tag{4}$$

$$\begin{aligned} \bigtriangledown_{w_-} \mathcal{L} &= \sum_{x_p \in \mathcal{S}^+} \delta(m > w_+^T x_p - w_-^T x_p) x_p \\ &- \sum_{x_n \in \mathcal{S}^-} \delta(m > w_-^T x_n - w_+^T x_n) x_n \end{aligned} \tag{5}$$

With simple variable substitution, we can derive the predicted logit value of x_t as

$$\begin{aligned} w'^T_+ x_t &= w_+^T x_t + \eta \mathcal{I}(x_t; \mathcal{S}) \\ w'^T_- x_t &= w_-^T x_t - \eta \mathcal{I}(x_t; \mathcal{S}) \end{aligned} \tag{6}$$

Here we take the situation where $p^+(x_t) > p^-(x_t)$ as example, the other side can be proved in a similar way. With the softmax activation and Eq (6), the query function $Q(x_t)$ can be expressed as

$$\begin{aligned} Q(x_t) &= 1 - (p^+(x_t) - p^-(x_t)) \\ &= 1 - \frac{e^{w_+^T x_t} e^{\eta \mathcal{I}} - e^{w_-^T x_t} e^{-\eta \mathcal{I}}}{e^{w_+^T x_t} e^{\eta \mathcal{I}} + e^{w_-^T x_t} e^{-\eta \mathcal{I}}} \\ &= \frac{2 e^{w_-^T x_t} e^{-\eta \mathcal{I}}}{e^{w_+^T x_t} e^{\eta \mathcal{I}} + e^{w_-^T x_t} e^{-\eta \mathcal{I}}} \\ &\triangleq \widetilde{Q}(\mathcal{I}; x_t) \end{aligned} \tag{7}$$

where $\mathcal{I}$ is abbreviation for $\mathcal{I}(x_t; \mathcal{S})$. With Eq (7), the query function $Q(x_t)$ can be regarded as a function of $\mathcal{I}$. Since both $p^+(x_t)$ and $p^-(x_t)$ are probabilities and $p^+(x_t) > p^-(x_t)$, it is easy to verify $\widetilde{Q}(\mathcal{I}; x_t) \in (0, 1)$, thus we can validate the corresponding monotonicity by the derivative w.r.t $\mathcal{I}(x_t; \mathcal{S})$

$$\frac{\partial \widetilde{Q}(\mathcal{I}; x_t)}{\partial \mathcal{I}} = \eta \left[\left(1 - \widetilde{Q}\left(\mathcal{I}; x_t\right) \right)^2 - 1 \right] < 0 \tag{8}$$

Thus $Q(x_t)$ is decreasing monotonically w.r.t $\mathcal{I}(x_t; \mathcal{S})$. □

B. Proof of Proposition 2

Proof. The proof follows the theoretical insight from MME [7], in terms of [1], the risk on target domain can be bounded by

$$\epsilon_t(h) \leq \epsilon_s(h) + d_{\mathcal{H}}(\mathcal{P}_s, \mathcal{P}_t) + C_0 \tag{9}$$

where $d_{\mathcal{H}}(\mathcal{P}_s, \mathcal{P}_t)$ represents the $\mathcal{H}$-divergence between source distribution $\mathcal{P}_s$ and target distribution $\mathcal{P}_t$

$$d_{\mathcal{H}}(\mathcal{P}_s, \mathcal{P}_t) = 2 \sup_{h \in \mathcal{H}} \left| \mathcal{P}(h(x_s) = 1) - \mathcal{P}(h(x_t) = 1) \right| \tag{10}$$

From the main text, the formulation of domain classifier family is defined as

$$\mathcal{H} = \left\{ \delta(|w_+^T x - w_-^T x| \geq m) | w_+, w_- \in \mathcal{R}^D \right\} \tag{11}$$

where $w_+, w_-, m, \delta(\cdot)$ follow the same definition as the main part of paper. Further, with the assumption $\mathcal{P}(h(x_t) =$

$1) \leq \mathcal{P}(h(x_s) = 1)$, the Eq (10) can be rewritten as

$$\begin{aligned} d_{\mathcal{H}}(\mathcal{P}_s, \mathcal{P}_t) &= 2 \sup_{h \in \mathcal{H}} |\mathcal{P}(h(x_s) = 1) - \mathcal{P}(h(x_t) = 1)| \\ &= 2 \sup_{h \in \mathcal{H}} (\mathcal{P}(h(x_s) = 1) - \mathcal{P}(h(x_t) = 1)) \\ &\leq 2 \sup_{h \in \mathcal{H}} \mathcal{P}(h(x_s) = 1) \\ &= 2 \sup_{w_+, w_-} \mathcal{P}(|w_+^T x_s - w_-^T x_s| \geq m) \qquad (12) \end{aligned}$$

Therefore, the derivation of Eq (12) indicates that the $\mathcal{H}$-divergence is bounded by the maximum ratio of source samples with score margin larger than parameter m, therefore, when the parameters w_+, w_- are optimized to maximize the margin between different classes, it is equivalent to find the upper bound of Eq (12), and further optimization over feature will minimize such upper bound, thus minimize the domain gap. □

C. Complexity Analysis

In this section, we briefly discuss how the complexity of our query function is computed. Recall that in the setting of SDM-AG, the query function is calculated as

$$\widetilde{Q}(x) = Q(x) + \lambda \left\langle \nabla_{\mathbf{f}} \mathcal{L}_m(x, y), \nabla_{\mathbf{f}} Q_m(x) \right\rangle \qquad (13)$$

For the simple margin sampling function $Q(x)$, the operation is to get the maximum and second maximum score over each class, and the operation is conducted on all samples, the complexity is $\mathcal{O}(NK)$. Suppose the weight of the i-th class of linear classifier $c(x)$ is denoted as $\mathbf{w}_i \in \mathcal{R}^D$, the gradient of $\nabla_{\mathbf{f}} \mathcal{L}_m(x, y)$ can be calculated as

$$\nabla_{\mathbf{f}} \mathcal{L}_m(x, y) = \sum_{i \neq y} \delta(m > \mathbf{w}_y^T \mathbf{f} - \mathbf{w}_i^T \mathbf{f})(\mathbf{w}_i - \mathbf{w}_y) \quad (14)$$

Similarly, the gradient of $\nabla_{\mathbf{f}} Q(x)$ can be calculated as

$$\nabla_{\mathbf{f}} Q(x) = \mathbf{p}_{2^*} \mathbf{w}_{2^*} - \mathbf{p}_{1^*} \mathbf{w}_{1^*} - (\mathbf{p}_{2^*} - \mathbf{p}_{1^*}) \sum_{i=1}^{K} \mathbf{p}_i \mathbf{w}_i \qquad (15)$$

In terms of Eq (14) and Eq (15), for each sample, the gradient is calculated as the form of summation over all classifiers, therefore the complexity is $\mathcal{O}(NKD)$. Finally, after calculating the $\widetilde{Q}(x)$, a sort function is applied, thus the total complexity of our query is $\mathcal{O}(NKD + N \log N)$

D. Visualization Results

Finally, we visualize the process of SDM at different steps by t-SNE [9]. Figure 1 shows visualization results of features for each data on target domain. For clarity, we only visualize the feature distribution of samples from 9 classes. The figure is plotted after each sampling step and selected samples are emphasized by a black box surrounding it. Finally, the feature distribution after the whole training process is also appended. From the Figure, we can observe that SDM algorithm can properly select some hard and informative examples in the target domain, which roughly distributes outside the envelope of the clusters corresponding to their class labels. Along with the sampling steps and training afterward, the number of ambiguous points in the t-SNE plane becomes less and the final feature distribution is compact for a classifier to recognize.

Method	Top-1 Acc.
ResNet	44.7±0.1
Random	78.1±0.6
UCN [5]	81.3±0.4
QBC [2]	80.5±0.3
AADA [8]	80.4±0.4
ADMA [4]	81.4±0.4
TQS [3]	83.1±0.4
SDM-AG (ours)	**85.0** ± **0.3**

Table 1. Classification accuracy (%) on the VisDA-2017 dataset with the budget of 5% data.

E. Results on VisDA dataset

We further conduct experiments on the dataset of VisDA-2017 [6], which involves larger scale of training samples. The results are compared with other state-of-the-art approaches in Table 1. Based on Table 1, we can observe that SDM-AG still outperforms state-of-the-art competitors significantly. To be specific, our SDM-AG approach surpasses a naive Random sampling baseline by about 7% top-1 accuracy. Compared with more advanced active domain adaptation methods like ADMA [4] or TQS [3], SDM-AG outperforms them by 3.6% and 1.9% respectively. These comparison demonstrate that our designed ADA paradigm can also cope with transfering scenarios with large scale of data samples and is able to mine informative training data from pools of unlabeled target data.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010. 1

[2] Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier, 1995. 2

[3] Bo Fu, Zhangjie Cao, Jianmin Wang, and Mingsheng Long. Transferable query selection for active domain adaptation. In

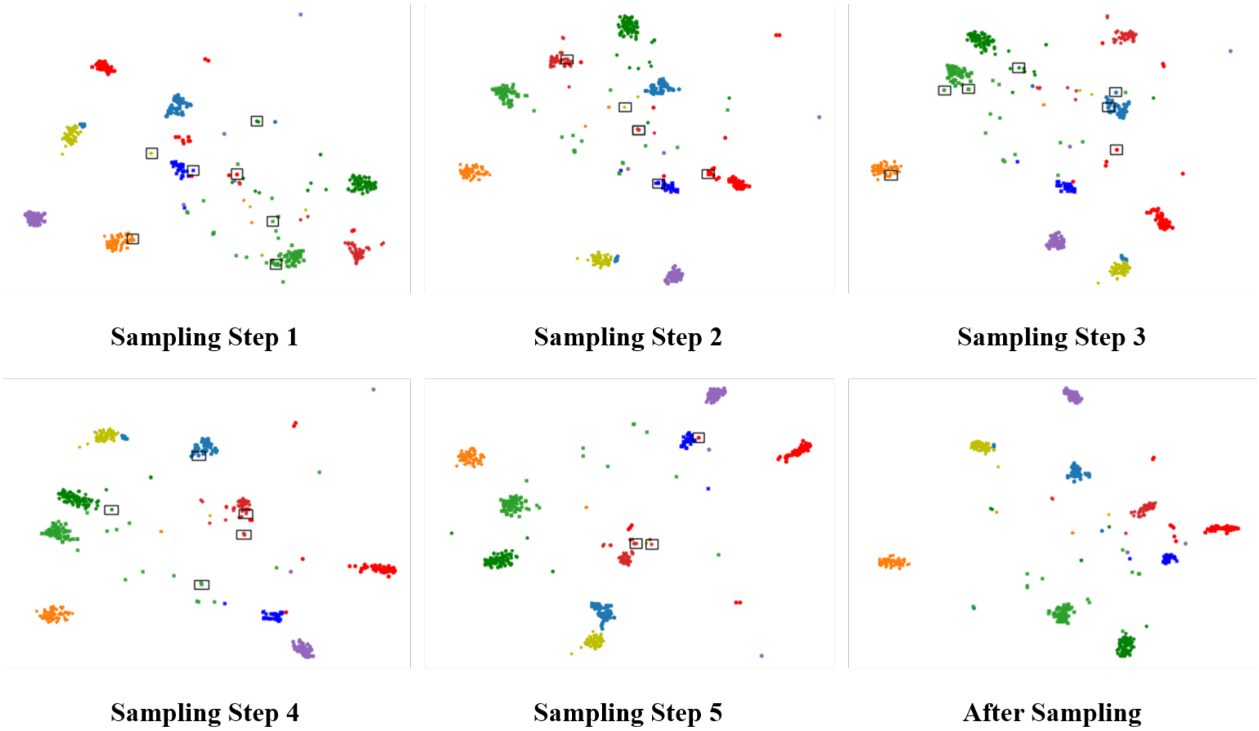

Figure 1. Visualization of SDM sampling process during training via t-SNE on Office-Home dataset. In the figure, each point is a data sample in target domain. The class type of a sample is represented by different colors. At each sampling step, the selected data sample is surrounded by a black box.

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7272–7281, 2021. 2

[4] Sheng-Jun Huang, Jia-Wei Zhao, and Zhao-Yang Liu. Cost-effective training of deep cnns with active model adaptation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1580–1588, 2018. 2

[5] Ajay J Joshi, Fatih Porikli, and Nikolaos P Papanikolopoulos. Scalable active learning for multiclass image classification. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2259–2273, 2012. 2

[6] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 2

[7] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019. 1

[8] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 739–748, 2020. 2

[9] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 2