STRONG-WEAK INTEGRATED SEMI-SUPERVISION FOR UNSUPERVISED DOMAIN ADAPTATION

Xiaohu Lu and Hayder Radha

Michigan State University, East Lansing, MI 48824, United States

(luxiaohu,radha)@msu.edu

ABSTRACT

Unsupervised domain adaptation (UDA) focuses on transferring knowledge learned in the labeled source domain to the unlabeled target domain. Semi-supervised learning is a proven strategy for improving UDA performance. In this paper, we propose a novel *strong-weak integrated semi-supervision (SWISS)* learning strategy for unsupervised domain adaptation. Under the proposed SWISS-UDA framework, a strong representative set with high confidence but low diversity target domain samples and a weak representative set with low confidence but high diversity target domain samples are updated constantly during the training process. Both sets are fused randomly to generate an augmented strong-weak training batch with pseudo-labels to train the network during every iteration. Moreover, a novel adversarial logit loss is proposed to reduce the intra-class divergence between source and target domains, which is back-propagated adversarially with a gradient reverse layer between the classifier and the rest of the network. Experimental results based on two popular benchmarks, Office-Home, and DomainNet, show the effectiveness of the proposed SWISS framework with our method achieving the best performance in both Office-Home and DomainNet.

Index Terms— Domain adaptation, Semi-supervised, Adversarial logit, Intra-class divergence

1. INTRODUCTION

The success of deep neural networks in tackling critical tasks, such as image classification, object detection, semantic segmentation, and image captioning is highly dependent on the availability of large amount of labeled training samples. Meanwhile, the generalization ability of deep neural networks is poor when applied to different target domains. The inclusion of additional labeled samples for the target domains can be a straightforward solution to this deficiency; but such a solution is arguably inefficient and costly.

Unsupervised domain adaptation (UDA) [1] tackles this problem by transferring the knowledge learned in the labeled source domain to the unlabeled target domain. This transference is usually accomplished by aligning the distributions of data points in source domain and target domain such that the classifier trained on the source domain can also be applied onto the target domain. Most current domain alignment methods can be classified into two categories: moment matching based and adversarial learning based. The idea of moment matching is based on the observation that two distributions are similar if their moments in different orders are all close to each other [2]. The Maximum Mean Discrepancy (MMD) [3] approach is widely used by this type of methods, which attempt to align domains through minimizing the distance between weighted sums of all raw moments. Another popular paradigm is leveraging the idea of adversarial learning [4] that is rooted in Generative Adversarial Networks (GANs) [5]. This approach is based on (a) training a domain classifier to align domains' distributions; yet (b) trying to trick or confuse the domain classifier in differentiating between source and target domain data by generating domain-invariant features. Hence, the adversarial learning is usually performed between the feature generator and the domain classifier so that when the domain classifier is fully confused by the feature generator the goal of domains' alignment is achieved.

Similar to unsupervised domain adaptation, semi-supervised learning [6, 7, 8] also focuses on tackling labeled and unlabeled samples. In order to make use of unlabeled data, semi-supervised learning methods assume that there exist some underlying relationships between distributions of data. Based on this assumption, several categories of methods are developed. Pseudo-label based methods [6, 9] select high-confidence predictions as the label for unlabeled samples. Information maximization based methods [10] consider that a good distribution should be individually certain and globally diverse, and utilize the information maximization loss to regularize the unlabeled samples. Regularization and normalization based methods [8, 7] adopt regularization and normalization strategies, e.g., batch normalization, to reduce the model's bias to the source domain such that the model's performance in target domain can be improved. Recently, an increasing number of researchers in unsupervised domain adaptation seek to borrow ideas from semi-supervised learning. For example, [11] assigned pseudo-labels with highest confidence to unlabeled samples, while in the work of [12] the information maximization loss is adopted to improve performance of domain adaptation.

In this paper, we propose a novel strong-weak integrated semi-supervision (SWISS) learning strategy that maintains a strong representative set with high confidence but low diversity samples and a weak representative set with low confidence but high diversity samples in the full training process. Furthermore, the proposed SWISS framework generates augmented training samples with pseudo-labels from these two representative sets to train the network in each iteration. Moreover, we also propose a novel adversarial logit loss to improve the performance by reducing the intra-class domain divergence. In summary, our contributions are:

- We introduce a novel *week semi-supervised learning* approach that provides a diverse set of training samples.
- We develop a new strong-weak integrated semi-supervision learning strategy for UDA.
- We propose a novel adversarial logit loss function that can reduce the intra-class domain divergence.
- Our method achieves state-of-the-art performance in several public benchmarks.

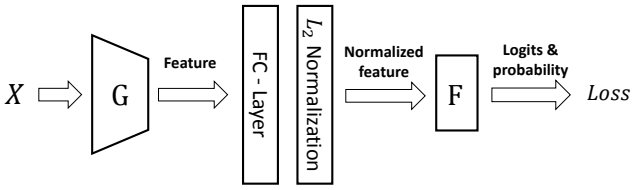

Fig. 1. The high-level architecture of the baseline network used in our unsupervised domain adaptation method. For forwarding, training samples are fed into a generator **G**, followed by a bottleneck network formed by a Fully Connected (FC) layer and a L_2 normalization layer, and a classifier **F**.

2. METHODOLOGY

2.1. Network Architecture

Fig. 1 shows the architecture of our baseline network that includes a feature generator network **G**, a bottleneck network formed by a Fully Connected (FC) layer and a L_2 normalization layer, and a classifier network **F**. For **G**, we employ the pre-trained ResNet-50 or ResNet-101 [13] as the backbone. For the bottleneck network, the output dimension of the FC-layer is 1024, the L_2 normalization layer normalizes and scales the feature vector of each input training sample into a normalized feature vector with the L_2 norm being a constant value T. The classifier network **F** is formed by k sub-classifiers corresponding to the k classes. In the forwarding path, for an image x from the input batch X, regardless of its domain, we firstly feed it into **G** to generate a 1024-d feature vector **G**(x). Then, we calculate the logits $\{l_1, l_2, ..., l_k\}$ between the normalized feature and each sub-classifier using the inner product, and estimate the prediction probabilities of the normalized feature belonging to each class as $\{p_1, p_2, ..., p_k\} = \sigma(\{l_1, l_2, ..., l_k\})$ with σ being the softmax function. After that, we calculate the loss terms based on the logits and probabilities.

2.2. Strong-weak Semi-supervision UDA

Fig. 2 shows the pipeline of our strong-weak supervision method for the typical single-source single-target domain adaptation. The key idea is to obtain a strong representative set $\mathcal{X}^{st}$ and a relatively weak representative set $\mathcal{X}^{wk}$ and then fuse them to generate reliable and diverse augmented samples to train the network. The weak representative set $\mathcal{X}^{wk}$ is updated by target samples with the prediction probability higher than a threshold after every iteration. While the strong representative set $\mathcal{X}^{st}$ is formed by target samples with the highest confidence of belonging to each class, which is updated every pre-specified number of iterations via self-learning. In the i_{th} iteration of training, the samples of each class in $\mathcal{X}^{wk}$ and $\mathcal{X}^{st}$ are fused to form an augmented supervision set with pseudo-labels $(X_i^{sw}, \hat{Y}_i^{sw})$, which along with the target-domain batch X_i are utilized to train the baseline network. The rational for this strong-weak supervision is that the strong representative set $\mathcal{X}^{st}$ is of highest prediction confidence but lower diversity, while the weak representative set $\mathcal{X}^{wk}$ is less reliable in prediction but with much higher in diversity. Hence, the combination of both sets gives a good balance between prediction confidence and diversity for the augmented training samples.

2.2.1. *Strong Supervision via Self-learning*

We adopt the same self-learning strategy as in [14] to get the strong representative set $\mathcal{X}^{st}$. Namely, given the target domain sample set $\{x_i\}_{i=1}^n$ with n samples and the baseline network as shown in Fig. 1,

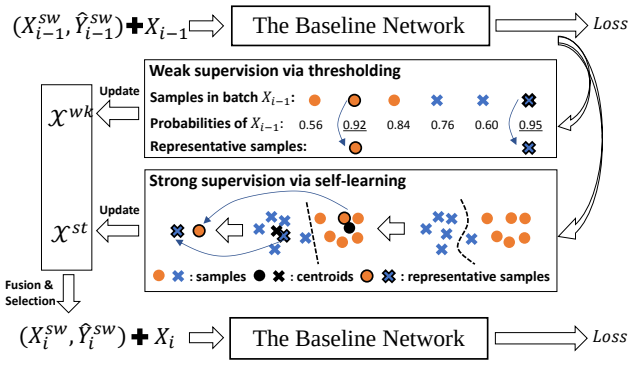

Fig. 2. The pipeline of our strong-weak supervision method for typical single-source single-target domain adaptation. A weak representative set $\mathcal{X}^{wk}$ and a strong representative set $\mathcal{X}^{st}$ are maintained during the training process. A strong-weak supervision set of training samples $(X_i^{sw}, \hat{Y}_i^{sw})$ is generated to train the baseline network.

there are three steps. Firstly, the prediction probabilities and the normalized feature vectors of training samples in the target domain are obtained by feeding the target domain samples into the baseline network. The concatenation of these probabilities P is a $n \times k$ matrix, and that of the normalized feature vectors V is a $n \times d$ matrix. Here n is the number of target domain samples, k is the number of classes, $d = 1024$ is the dimension of feature vector.

Then, the initial centroid of class j is obtained as:

$$c_j^{(0)} = \frac{P_{,j}^T V}{\sum_j P_{,j}}, \tag{1}$$

where $P_{,j}$ denotes the j_{th} column of the probability matrix, which corresponds to class j. After that, the initial self-supervised pseudo-label can be obtained by assigning each sample to the class with closest centroid as:

$$\hat{y}^{(0)} = arg\min_j D(v, c_j^{(0)}) \tag{2}$$

where v represents the normalized feature vector of a target domain sample, and $D(,)$ measures the cosine distance between two vectors.

Finally, the class centroids are recalculated by replacing the probability matrix P with a $n \times k$ one-hot distribution matrix $\mathbb{1}$ obtained from the initial self-supervised pseudo-label $\hat{y}^{(0)}$, and the strong representative set $\mathcal{X}^{st}$ is obtained by selecting the target samples closest to each class centroids. Namely, for a specific class j,

$$\begin{aligned} c_j^{(1)} &= \frac{\mathbb{1}_{,j}^T V}{\sum_j \mathbb{1}_{,j}}, \\ x_j^{st} &= x_{arg\min_{i \in [1,n]} D(v_i, c_j^{(1)})} \end{aligned} \tag{3}$$

where $\mathcal{X}^{st} = \{x_j^{st}\}_{j \in [1,k]}$, n is the number of target samples, k is the number of classes.

Under the self-learning strategy above, the distribution of the pseudo-labels will be slightly shifted toward the target domain by assigning each target sample to the closest class in a self-supervised way. This self-learning procedure is performed every pre-specified number of iterations, e.g., 200 iterations, and only the samples with highest confidence are selected. Therefore this set is considered relatively "strong" in comparison with the weak supervision set that is introduced below.

2.2.2. Weak Supervision via Thresholding

For weak supervision, we employ a higher updating frequency than the strong supervision. And, we adopt a simple thresholding strategy to update $\mathcal{X}^{wk}$ for each batch of target samples. More specifically, after inputting a target batch $X = \{x_i\}_{i=1}^{b_s}$ with size b_s to the baseline network, we identify the highest probabilities $\boldsymbol{p} = \{p_i\}_{i=1}^{b_s}$ and the corresponding pseudo-labels $\boldsymbol{l} = \{l_i\}_{i=1}^{b_s}$ for the b_s samples in the batch. Then, for each class j that has an entry in the pseudo-label vector $\boldsymbol{l}$, the sample with the highest probability (which must be greater than a threshold λ) of belonging to this class j is selected as the new representative sample for that class j. In other words, under the weak supervision learning process, there is a single and unique representative sample x_j^{wk} for any class j at any given time. Hence, a new sample could potentially be used to update the corresponding representative sample x_j^{wk} for class j in the current weak set $\mathcal{X}^{wk}$. Namely,

$$\begin{aligned} x_j^{wk} &= x_{arg\max_i p_i}, \\ \text{s.t. } l_i &= j,\ p_i > \lambda, \end{aligned} \tag{4}$$

where $\mathcal{X}^{wk} = \{x_j^{wk}\}_{j\in[1,k]}$, k is the number of classes.

2.2.3. Strong-weak Fusion

In practice, we find that simply using the strong supervision can only slightly reduce the domain divergence. This is because the network can easily over-fit on the strong representative set by remembering them instead of learning knowledge from them due to the small amount of samples in $\mathcal{X}^{st}$ (one sample per class). Meanwhile, in the case of weak supervision only, the discriminability of the network maybe impaired under challenging scenarios due to the error associated with predicting the pseudo-labels in $\mathcal{X}^{wk}$. Fusing both $\mathcal{X}^{st}$ and $\mathcal{X}^{wk}$ can achieve a good balance between them, and consequently, such fusion achieves smaller domain divergence and higher discriminability. Therefore, we adopt a random fusion strategy to generate the fused strong-weak representative set $\mathcal{X}^{sw}$:

$$x_j^{sw} = r x_j^{st} + (1-r) x_j^{wk},\ \text{s.t. } r \in (0,1), \tag{5}$$

where x_j^{sw} is the fused representative sample for class j, r is a random number in (0, 1). The reason for using a random value instead of a constant one for r is to increase the diversity of the strong-weak representative set $\mathcal{X}^{sw}$. In practice, we find that adding the full set of $\mathcal{X}^{sw}$ along with the corresponding pseudo-label as supervision may override the target domain itself when the number of classes k is much greater than the batch size, e.g., k=345 in DomainNet. Therefore, we only select a small batch X^{sw} from $\mathcal{X}^{sw}$ according to the data distribution of the input batch. Namely, given the input batch X and its most-likely class label $\hat{Y}$ which is obtained from the estimated probability, we have $X^{sw} = \{x_j^{sw}\}_{j\in\hat{Y}}$. Finally, the selected batch of images along with the corresponding label $(X^{sw}, \hat{Y}^{sw})$ are utilized to train the baseline network.

2.3. Loss Functions

For the unlabeled target domain, we adopt the mutual information maximization loss L_{IM} [15, 12] as the baseline and propose a new adversarial logit loss as an improvement. The L_{IM} is based on the mutual information $I(X;\hat{Y})$ between the input X and the output $\hat{Y}$, namely, $I(X;\hat{Y}) = H(\hat{Y}) - H(\hat{Y}|X)$. L_{IM} is formulated as:

$$L_{IM} = \sum_{j=1}^{k} \hat{p_j} log(\hat{p_j}) + \frac{1}{b_s}\sum_{i=1}^{b_s}\sum_{j=1}^{k} -p_{ij} log(p_{ij}), \tag{6}$$

where b_s is batch size, k is the number of classes, p_{ij} is the probability of a sample i belonging to class j, $\hat{p_j} = \frac{1}{b_s}\sum_{i=1}^{b_s} p_{ij}$ is the average probabilities of samples belonging to class j in a training batch with b_s samples. Note that the mutual information between X and $\hat{Y}$ can also be reformulated as: $I(X;\hat{Y}) = H(X) - H(X|\hat{Y})$, which means that minimizing $H(X|\hat{Y})$ can also improve the performance. Given that $H(X|\hat{Y}) = \sum_{j=1}^{k} p_j H(X|\hat{Y}=j)$, we can see that the final goal is to minimize the conditional entropy of X given that it belongs to class j.

Recall that in our baseline network, the feature vector v fed into the classifier $\mathbf{F}$ is normalized by a constant temperature T. Therefore, the equation for the logit l_j between v and the j_{th} prototype can be reformulated as: $l_j = \mathbf{w}_j^{\mathrm{T}} \cdot v = T||\mathbf{w}_j^{\mathrm{T}}||cos(\theta)$ where $\mathbf{w}_j$ is the j_{th} prototype, which is also the j_{th} weight vector in the classifier $\mathbf{F}$, θ is the angle between v and $\mathbf{w}_j$. We can see from the equation above that for a given class j, the prototype $\mathbf{w}_j$ is constant for all the samples belonging to this class, therefore, a feasible way to reduce the intra-class variance of feature vectors is to minimize θ for those samples. In order to reduce the angle θ for target domain samples, we propose a novel adversarial logit loss which tries to enlarge $cos(\theta)$ by reducing $\mathbf{w}_j$. More specifically, we formulate the adversarial logit loss L_{ALL} as:

$$L_{ALL} = \frac{1}{b_s}\sum_{i=1}^{b_s} \begin{cases} \max\limits_{j\in[1,k]} l_{ij}, \text{if } \max\limits_{j\in[1,k]} p_{ij} > \lambda \\ 0, \text{otherwise} \end{cases} \tag{7}$$

where b_s is the batch size, l_{ij} is the logit value between sample i and class j, p_{ij} is the corresponding prediction probability. Eq. (7) implies that for a given input target domain sample, we firstly calculate its logits to each class in the forwarding path, then select the largest one as its contribution toward the overall value of the proposed adversarial loss L_{ALL}. Subsequently, L_{ALL} is back-propagated with a gradient reverse layer between the classifier $\mathbf{F}$ and the rest of the network such that $\mathbf{F}$ will try to minimize L_{ALL} by reducing the norm of prototypes, while the bottleneck layers and $\mathbf{G}$ focus on maximizing L_{ALL} by decreasing the angle θ.

For a labeled source batch, we utilize the standard Cross Entropy loss L_{CE} to train the network to minimize the classification error as follows:

$$L_{CE} = \frac{1}{b_s}\sum_{i=1}^{b_s}\sum_{j=1}^{k} -y_{ij} log(p_{ij}) \tag{8}$$

where b_s is batch size, k is the number of classes, $y_{ij} = 1$ if the (ground truth) label of the i_{th} sample is j, otherwise $y_{ij} = 0$.

For the strong-weak semi-supervision batch $(X^{sw}, \hat{Y}^{sw})$, one straightforward approach is to use the Cross Entropy as the loss function. However, we found that the derivative of the Cross Entropy is relatively small when the probability is approaching 1.0, which means that it's hard for the Cross Entropy loss to optimize the network when the pseudo-label is incorrectly estimated with a high confidence. Therefore, we design the following function for the strong-weak supervision loss L_{SW}:

$$L_{SW} = \frac{1}{b_s}\sum_{i=1}^{b_s}\sum_{j=1}^{k} y_{ij}(1.0 - p_{ij}) \tag{9}$$

which replace $-log(p_{ij})$ in Eq. (8) with $1.0 - p_{ij}$ such that the derivative is constant regardless of the probability.

As a summary, the overall optimization goal is:

$$minimizing \begin{cases} L_{CE}\ for\ S \\ k1 L_{IM} + k2 L_{ALL} + k3 L_{SW}\ for\ T \end{cases} \tag{10}$$

Table 1. Classification accuracies (%) on medium-size Office-Home dataset with backbone ResNet-50.

Method (S→ T)	Ar→ Cl	Ar→ Pr	Ar→ Re	Cl→ Ar	Cl→ Pr	Cl→ Re	Pr→ Ar	Pr→ Cl	Pr→ Re	Re→ Ar	Re→ Cl	Re→ Pr	**Avg.**
CDAN+BSP [16]	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
SHOT-IM [12]	55.4	76.6	80.4	66.9	74.3	75.4	65.6	54.8	80.7	73.7	58.4	83.4	70.5
DCAN [17]	57.9	76.2	79.3	67.3	76.1	75.6	65.4	56.0	80.7	74.2	61.2	84.2	71.2
FixBi [18]	58.1	77.3	80.4	67.7	79.5	78.1	65.8	57.9	81.7	76.4	62.9	86.7	72.7
AANet [4]	58.4	79.0	82.4	67.5	79.3	78.9	68.0	56.2	82.9	74.1	60.5	85.0	72.8
SCDA [19]	60.7	76.4	82.8	69.8	77.5	78.4	68.9	59.0	82.7	74.9	61.8	84.5	73.1
SWISS(our)	60.7	78.2	81.9	69.3	77.9	79.4	69.5	57.4	82.6	76.4	59.8	85.0	73.2

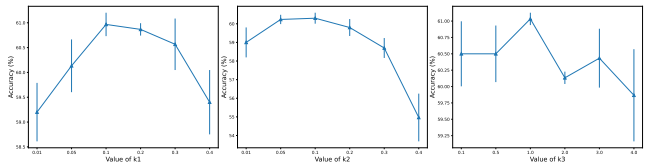

Fig. 3. Parameter sensitivity analysis for k1, k2, k3 in task Ar→ Cl.

Table 2. Classification accuracies (%) on large-size DomainNet dataset with backbone ResNet-101.

Model	DomainNet						
	clp	inf	pnt	qdr	rel	skt	Avg.
Source only	25.6	16.8	25.8	9.2	20.6	22.3	20.1
DADA [20]	26.1	20.0	26.5	12.9	20.7	22.8	21.5
CDAN [21]	31.6	27.1	31.8	12.5	33.2	35.8	28.7
MCC [22]	33.6	30.0	32.4	13.5	28.0	35.3	28.8
SCDA [19]	36.9	30.1	35.2	21.3	37.4	38.8	33.3
SWISS (our)	**39.0**	**34.5**	36.4	**23.4**	37.8	**41.5**	**35.4**

Table 3. Classification accuracies (%) on Office-Home with backbone ResNet-50 under different configurations of loss functions.

Method	Ar	Cl	Pr	Re	**Avg.**
ResNet50 [13]	47.6	41.8	43.4	51.7	46.1
$+L_{IM}$	71.5	73.4	66.9	71.5	70.8
$+L_{IM}+L_{ALL}$	73.1	73.7	68.5	72.5	72.0
$+L_{IM}+L_{SW}$	72.3	74.4	67.7	73.2	71.9
$+L_{ALL}+L_{SW}$	72.6	73.7	68.6	73.5	72.1
$+L_{IM}+L_{ALL}+L_{SW}$	73.6	75.5	69.8	73.7	73.2

Details about the hyper-parameters $k1$, $k2$, $k3$ will be discussed in the experimental results section.

3. EXPERIMENTS

For quantitative assessment of the proposed strong-weak integrated semi-supervision (SWISS) method, we evaluate it on two popular benchmarks Office-Home [23], DomainNet [2]; and, we compare our proposed method with state-of-the-art methods, namely, DADA [20], CDAN+BSP [16], MCC [22], DCAN [17], SHOT-IM [12], and the most recent FixBi [18], SCDA [19], AANet [4].

We employ the pre-trained ResNet-50 or ResNet-101 as the feature generator **G** as was done in [12, 24, 17, 21]. For the classifier **F**, we use a linear layer with the input and output dimensions being $(1024, k)$, where k is the number of classes. The network is trained with Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9. The learning rate is scheduled with the strategy in [25]. The up-limit of training iteration for Office-Home and DomainNet are 5000 and 10000, respectively. The value of temperature T in feature normalization is set to 20 following the results of [26]. Similar to the setting in [12], we randomly run our method three times via PyTorch and report the average accuracy.

Parameter sensitivity. There are four hyper-parameters in our method, namely, k1, k2, k3, and λ. For λ, which is the prediction probability threshold, we set it to 0.8 empirically. For k1, k2, k3, we test k1=$\{0.01, 0.05, 0.1, 0.2, 0.3, 0.4\}$, k2=$\{0.01, 0.05, 0.1, 0.2, 0.3, 0.4\}$, k3=$\{0.1, 0.5, 1.0, 2.0, 3.0, 4.0\}$ in the task Ar→ Cl in Office-Home dataset. Fig. 3(a)-(c), we can see that k1=0.1, k2=0.05, k3=1.0 is the best combination that leads to highest accuracy and relatively smaller sensitivity. Therefore, we use k1=0.1, k2=0.05, k3=1.0, $\lambda = 0.8$ for all of our experiments.

Results on Office-Home For the challenging medium-size Office-Home dataset, we can see from Tab. 1 that our SWISS achieve state-of-the-art performance, which is 0.1% higher than the second-place method SCDA [19], and 0.4% higher than the third-level methods AANet [4] and FixBi [18]. Comparing our approach to other methods, we can see that our method can outperform other methods especially in several challenging tasks such as Ar→ Cl, Pr→ Ar.

Results on DomainNet For the most challenging large-size DomainNet dataset, our SWISS outperforms state-of-the-art method SCDA [19] by a large margin 2.1%. We can see from Tab. 2 that in all six sub-domains, our method accomplishes 4 best and 2 second-place scores among all the methods. Comparing the performance of our SWISS approach in Office-Home and DomainNet, we can see that the proposed method performs better in the challenging cases. This is consistent with the conclusion drawn from the Office-Home dataset that our method achieves better performance in the challenging sub-tasks.

Ablation study on different losses. Tab. 3 shows the classification accuracies on Office-Home with difference configurations of loss functions. We can observe clear improvement of the proposed adversarial logit loss L_{ALL} and strong-weak semi-supervision loss L_{SW} by comparing $+L_{IM}+L_{ALL}$ and $+L_{IM}+L_{SW}$ with $+L_{IM}$, which is 1.2% and 1.1% for L_{ALL} and L_{SW} respectively. The combination of all these three loss functions accomplishes the best performance among all the configurations with an average improvement of 2.4% over L_{IM}, which clearly shows the advantages of the proposed SWISS framework.

4. CONCLUSION

This paper presents a novel strong-weak semi-supervision strategy for unsupervised domain adaptation. A strong representative set with high prediction confidence but low diversity and a weak representative set with low prediction confidence but high diversity are maintained and updated during the training process. The fusion of the two sets generates augmented training samples with pseudo-labels, which are utilized as supervision for training the network. Moreover, a novel adversarially optimized loss based on logit is developed to further reduce the intra-class domain divergence. Comprehensive experiments on several popular benchmarks have demonstrated the effectiveness of the proposed method.

Acknowledgement: This work has been supported in part by a grant from the Semiconductor Research Corporation (SRC), an Amazon Research Award (ARA), and the MSU Foundation under the Strategic Partnership Grants (SPG) program.

5. REFERENCES

[1] Sicheng Zhao, Xiangyu Yue, Shanghang Zhang, Bo Li, Han Zhao, Bichen Wu, Ravi Krishna, Joseph E Gonzalez, Alberto L Sangiovanni-Vincentelli, Sanjit A Seshia, et al., "A review of single-source deep unsupervised visual domain adaptation," *arXiv preprint arXiv:2009.00155*, 2020.

[2] 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*, 2019, pp. 1406–1415.

[3] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola, "A kernel method for the two-sample-problem," in *Advances in Neural Information Processing Systems*, 2007, pp. 513–520.

[4] Haifeng Xia, Handong Zhao, and Zhengming Ding, "Adaptive adversarial network for source-free domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9010–9019.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[6] Zijian Hu, Zhengyu Yang, Xuefeng Hu, and Ram Nevatia, "Simple: Similar pseudo label exploitation for semi-supervised classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15099–15108.

[7] Zhaowei Cai, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Zhuowen Tu, and Stefano Soatto, "Exponential moving average normalization for self-supervised and semi-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 194–203.

[8] Abulikemu Abuduweili, Xingjian Li, Humphrey Shi, Cheng-Zhong Xu, and Dejing Dou, "Adaptive consistency regularization for semi-supervised transfer learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6923–6932.

[9] Dong-Hyun Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, 2013, vol. 3.

[10] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama, "Learning discrete representations via information maximizing self-augmented training," *arXiv preprint arXiv:1702.08720*, 2017.

[11] Jaehoon Choi, Minki Jeong, Taekyung Kim, and Changick Kim, "Pseudo-labeling curriculum for unsupervised domain adaptation," in *British Machine Vision Conference (BMVC)*. Springer, 2019.

[12] Jian Liang, Dapeng Hu, and Jiashi Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," *arXiv preprint arXiv:2002.08546*, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[14] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu, "Cross-domain gradient discrepancy minimization for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3937–3946.

[15] Shuang Li, Mixue Xie, Kaixiong Gong, Chi Harold Liu, Yulin Wang, and Wei Li, "Transferable semantic augmentation for domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11516–11525.

[16] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in *International Conference on Machine Learning*, 2019, pp. 1081–1090.

[17] Pengfei Ge, Chuan-Xian Ren, Dao-Qing Dai, and Hong Yan, "Domain adaptation and image classification via deep conditional adaptation network," *arXiv preprint arXiv:2006.07776*, 2020.

[18] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang, "Fixbi: Bridging domain spaces for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1094–1103.

[19] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li, "Semantic concentration for domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9102–9111.

[20] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko, "Domain agnostic learning with disentangled representations," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5102–5112.

[21] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan, "Conditional adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, 2018, pp. 1640–1650.

[22] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang, "Minimum class confusion for versatile domain adaptation," in *European Conference on Computer Vision*. Springer, 2020, pp. 464–480.

[23] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5018–5027.

[24] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai, "Adversarial-learned loss for domain adaptation," *arXiv*, vol. abs/2001.01046, 2020.

[25] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[26] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8050–8058.