Supplementary : Learning to Detect Open Classes for Universal Domain Adaptation

Bo Fu *, Zhangjie Cao *, Mingsheng Long, and Jianmin Wang

Paper ID 2396

1 Overview

In the supplementary material, we provide proofs, experiment details and more comprehensive experimental results.

2 Proofs

Confidence has contour lines where all class distributions on the line share the same confidence. In the paper, we claim that confidence has high discriminability for extremely confident and uncertain predictions, meaning that the contour line for extremely high and low confidence should be short.

For each confidence, the length of contour line is proportional to the number of possible value combinations of probabilities, i.e. the size of solution space, satisfying the confidence. If the confidence is larger than 0.5, meaning that the highest probability is higher than 0.5, and the sum of the rest probabilities is smaller than 0.5. Let the three probabilities $p_i, (i = 1, ..., n)$ without loss of generality, we let p_1 be the largest probabilities. Then we have

$$\begin{aligned} & p_1 \geq p_i \geq 0, (i = 1, ..., n) \\ & \sum_{i=1}^{n} p_i = 1. \end{aligned} \tag{1}$$

The solution is

$$\begin{aligned} & \sum_{i=2}^{n} p_i = 1 - p_1, \\ & p_i \geq 0.(i = 1, ..., n) \end{aligned} \tag{2}$$

The solution space is proportional to $1 - p_1$. So the solution space increases from confidence 1.0 to 0.5. The length of the contour line increase from confidence 1.0 to 0.5.

For the most uncertain prediction, i.e. every class has the same probability, the number of class distributions is 1. Therefore, the contour line length roughly first increases and then decreases when the confidence decrease from 1.0 to $\frac{1}{|\mathcal{C}^s|}$, where $|\mathcal{C}^s|$ is the number of classes. So confidence has a shorter contour line for extremely high and low confidence and is complementary to entropy.

3 Experiment Details

3.1 Label Set Separation

Office-31 [10] is a visual domain adaptation dataset with 31 categories in 3 visually distinct domains: Amazon(A), Dslr(D), Webcam(W). Following UAN [13], we use the 10 classes shared by Office-31 and Caltech-256 [3] as the common label set $\mathcal{C}$. In alphabetical order, the next 10 classes are used as the $\overline{\mathcal{C}}_s$, and the rests are used as the $\overline{\mathcal{C}}_t$. The class names in the common label set, the source private label set and the target private label set are shown in the "Office-31" row in Table 1.

Office-Home [12] is a more diverse dataset consisting of 15,500 images in 65 classes in office or home settings. It consists of 4 domains: Artistic images (A), Clip-Art images (C), Product images (P) and Real-World images (R). Following UAN [13], in alphabet order, we use the first 10 classes as $\mathcal{C}$, the next 5 classes as $\overline{\mathcal{C}}_s$ and the rest as $\overline{\mathcal{C}}_t$. The class names in the common label set, the source private label set, and the target private label set are shown in the "Office-Home" row in the Table 1. Since there are too many classes in the target private label set, we show part of all classes.

VisDA [9] is a simulation-to-real dataset containing over 280K images across 12 classes. VisDA includes two domains: Synthetic and Real. Following UAN [13], in alphabet order, we use the first 6 classes as $\mathcal{C}$, the next 3 classes as $\overline{\mathcal{C}}_s$ and the rest as $\overline{\mathcal{C}}_t$. The class names in the common label set, the source private label set and the target private label set are shown in the "VisDA" row in Table 1.

DomainNet [8] is by far the largest domain adaptation dataset, consists of six distinct domains: Clipart(C), Infograph(I), Painting(P), Quickdraw(Q), Real(R) and Sketch(S). It contains over 0.6 million images across 345 classes. In alphabet order, we use the first 150 classes as $\mathcal{C}$, the next 50 classes as $\overline{\mathcal{C}}_s$ and the rest as $\overline{\mathcal{C}}_t$. The class names in the common label set, the source private label set, and the target private label set are shown in the "DomainNet" row in the Table 1. Since there are too many classes in this dataset, we only show part of the classes in the three label sets.

3.2 Data Preprocessing

We employ 5 extra classifiers ($m = 5$) in all the experiments. To make the multiple classifiers more diverse, we apply different data preprocessing methods to the data input to different classifiers. In detail, we use the following five data preprocessing methods: (**1**) Random Affine and Random Grayscale ; (**2**) Random Perspective and Color Jitter ; (**3**) Random Affine and Color Jitter ; (**4**) Random Affine and Random Perspective ; (**5**) Random Perspective and Random Grayscale. The classifier C for prediction use no extra data preprocessing but only resizing to fix size 256x256 and random cropping.

Table 1. The specific class for each experiment. And due to a large number of classes, some datasets only show some examples. **All settings follow UAN [13] if the dataset is used in UAN.**

Dataset	common label set	source private label set	target private label set
Office-31	back_pack, bike, calculator, headphones, keyboard, laptop_computer, monitor, mouse, mug, projector	bike_helmet, bookcase, bottle, desk_chair, desk_lamp, desktop_computer, file_cabinet, letter_tray, mobile_phone, paper_notebook	pen, phone, printer, punchers, ring_binder, ruler, scissors, speaker, stapler, tape_dispenser, trash_can
VisDA	aeroplane, bus, horse, knife, person, skateboard	truck, bicycle, car	motorcycle, plant, train
Office-Home	alarm_clock, backpack, batteries, bed, bike, bottle, bucket, calculator, calendar, candles	chair, clipboards, computer, couch, curtains	eg: desk_lamp, drill, eraser, exit_sign, fan, file_cabinet, flipflops, flowers, folder, fork, glasses, hammer, helmet, kettle, keyboard, knives
DomainNet	eg: aircraft_carrier, backpack, banana, bandage, butterfly, cactus, cake, calculator, duck, dumbbell, ear, elbow	eg: hot_dog, hot_tub, ice_cream, jacket, jail, key, knife, lion, lipstick, lobster, lollipop, mailbox	eg: nose, ocean, owl, paintbrush, potato, purse, rabbit, river, rollerskates, sailboat, school_bus, scissors

3.3 Network Initialization

For the 5 different classifiers, we initialize them with different random initializations. We use ResNet-50 [4] pre-trained on ImageNet [1] as our backbone network, we remove the original classifier and add multiple classifiers $G_i|_{i=1}^m$ and G on the layer before the classifier.

3.4 Hyperparameter

For optimizer parameters such as learning rate, we use cross-validation on source data. For other hyperparameters, we use the reverse validation risk proposed in DANN [2] to select hyper-parameters. In detail, we first randomly split the labeled source and unlabeled target data into training sets (S', T') respectively containing 90% of the original examples and validation sets (S_v, T_v). We learn a model η (consisting of F, $G_i|_{i=1}^m$, and G) with S' and T' based on our approach. Then we learn a reverse model η_r using the self-labeled set $(x, \eta(x))$ by labeling

T' with η, where the classifier has one entry for each common class and one entry for all the data labeled as open class by η. Finally, we evaluate the H-score of η_r on source validation set S_v, where all the source classes not existing in the self-labeled set $(x, \eta(x))$ are regarded as open class. This validation error is the reverse validation risk and we select hyper-parameters to induce lower reverse validation risk.

The base learning rate is 0.001 for VisDA, 0.01 for Office-31, Office-Home and DomainNet. The decay strategy of learning rate is the same as DANN. The batch size is 32. As for the threshold w_0, we set it 0.5 on Office-31, VisDA and Office-Home, 0.55 on DomainNet. The experiments run in 3 times.

4 Additional Experiment Results

4.1 Significance of Results

We show per-class accuracy on Office-Home for a fair comparison, which is not present in the paper due to the space limit. We can observe that CMU consistently outperforms previous methods. To demonstrate that CMU improves previous methods including UAN significantly, we show the oracle results for per-class accuracy on Office-31, VisDA, and Office-Home datasets. To obtain the oracle results, we select out all the source and target data in the common label set and perform closed set domain adaptation method DANN [2] on it, which is used in both CMU and UAN and avoids the influence of other factors. The accuracy of the open class is computed as 100% for the oracle. As shown in Table 2 and 3, on the Office-31 dataset, the performance gap between CMU and Oracle is smaller than that between CMU and UAN, the state-of-the-art method for UDA, while on the Office-Home and VisDA datasets, the gaps are comparable. The results demonstrate that the improvement of CMU over UAN is significant. The per-class accuracy is bounded by the based closed set domain adaptation method and can be improved with better closed set domain adaptation, which is not the focus of the paper.

Table 2. Tasks on **Office-31** and **VisDA2017** dataset

Method	Office-31 (Acc)							VisDA	
	A → W	D → W	W → D	A → D	D → A	W → A	Avg	Acc	H-score
ResNet [4]	75.94	89.60	90.91	80.45	78.83	81.42	82.86	52.80	25.44
DANN [2]	80.65	80.94	88.07	82.67	74.82	83.54	81.78	52.94	25.65
RTN [6]	85.70	87.80	88.91	82.69	74.64	83.26	84.18	53.92	26.02
IWAN [14]	85.25	90.09	90.00	84.27	84.22	86.25	86.68	58.72	27.65
PADA [14]	85.37	79.26	90.91	81.68	55.32	82.61	79.19	44.98	23.05
ATI [7]	79.38	92.60	90.08	84.40	78.85	81.57	84.48	54.81	26.34
OSBP [11]	66.13	73.57	85.62	72.92	47.35	60.48	67.68	30.26	27.31
UAN [11]	85.62	94.77	97.99	86.50	85.45	85.12	89.24	60.83	30.47
CMU	86.86	95.72	98.01	89.11	88.35	88.61	91.11	61.42	34.64
Oracle	89.32	96.65	98.77	90.22	89.47	89.27	92.28	66.42	-

Table 3. Average class accuracy (%) on **Office-Home** dataset

Method	Office-Home												
	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R →C	R→P	Avg
ResNet [4]	59.37	76.58	87.48	69.86	71.11	81.66	73.72	56.30	86.07	78.68	59.22	78.59	73.22
DANN [2]	56.17	81.72	86.87	68.67	73.38	83.76	69.92	56.84	85.80	79.41	57.26	78.26	73.17
RTN [6]	50.46	77.80	86.90	65.12	73.40	85.07	67.86	45.23	85.50	79.20	55.55	78.79	70.91
IWAN [14]	52.55	81.40	86.51	70.58	70.99	85.29	74.88	57.33	85.07	77.48	59.65	78.91	73.39
PADA [14]	39.58	69.37	76.26	62.57	67.39	77.47	48.39	35.79	79.60	75.94	44.50	78.10	62.91
ATI [7]	52.90	80.37	85.91	71.08	72.41	84.39	74.28	57.84	85.61	76.06	60.17	78.42	73.29
OSBP [11]	47.75	60.90	76.78	59.23	61.58	74.33	61.67	44.50	79.31	70.59	54.95	75.18	63.90
UAN [11]	63.00	82.83	87.85	76.88	78.70	85.36	78.22	58.59	86.80	83.37	63.17	79.43	77.02
CMU	63.52	83.81	88.94	77.72	79.37	86.85	78.61	59.27	88.25	84.06	64.57	81.36	78.03
Oracle	66.46	85.59	89.43	80.52	81.56	87.26	81.20	64.35	89.25	85.27	66.78	82.42	80.01

As for H-score, the oracle result for the open class accuracy is 100%, and then the H-score for the oracle only changes with per-class accuracy. Thus, we do not report the result since it provides no additional information.

4.2 Supplemental Ablation Study

To compensate for the ablation study in the main text for the performance of each individual component, we use the three components of the criterion but still use domain adversarial learning on Office-31. As shown in Table 4, 'CMU' uses all the criteria and the deep ensemble. 'Ent' only uses calibrated entropy; 'Conf' only uses calibrated confidence; 'Cons' only uses consistency.

Table 4. Supplemental Ablation Study tasks on **Office-31** dataset

Method	D → W		A → D		W → A		Avg (6 task)	
	Acc	H-score	Acc	H-score	Acc	H-score	Acc	H-score
CMU	95.72	79.32	89.11	68.11	88.61	72.23	91.11	73.14
Ent	93.11	76.45	87.44	65.45	86.72	69.34	89.36	69.68
Conf	92.43	74.74	86.72	62.38	85.64	67.73	88.82	68.53
Cons	92.21	74.58	86.41	62.07	85.24	67.37	88.43	68.10

The deep ensemble of entropy is an out-of-distribution detection algorithm in [5], without the domain adversarial learning. For Office-31, the mean accuracy is 88.33; the H-score is 68.30.

4.3 Variance

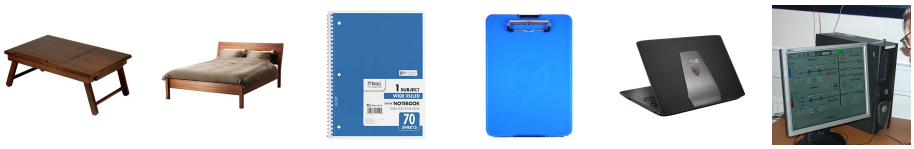
The average of the variances of six Office-31 tasks is ± 0.5 for Acc and ± 0.9 for H-score. The average of the variances of twelve Office-Home tasks is ± 0.5 for Acc and ± 1.0 for H-score. The small variance further demonstrates the significance of the results.

4.4 Examples of Classification Results

We show examples of classification results of CMU and UAN.

(a) bed T (b) bed S (c) bottle T (d) bottle S (e) alarm T (f) alarm S

Fig. 1. Images that are classified correctly by both methods. Which domain the image belongs to is noted in the caption (S: source T: target). The source and target images are quite similar when it can be classified correctly by both methods.



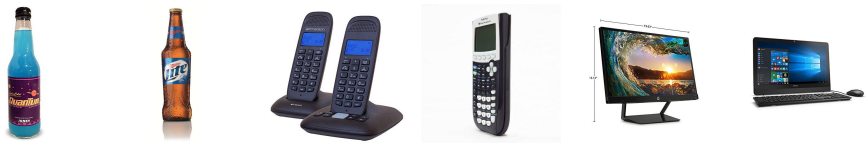
(a) table T (b) bed S (c) notebook T (d) clipboard S (e) laptop T (f) computer S

Fig. 2. Images that are classified correctly by CMU but incorrectly by UAN. Fig. (a) shows a table in the target domain is mis-classified as bed by UAN. Fig. (c) shows a notebook in the target domain is mis-classified as clipboards. Fig. (e) shows a laptop in the target domain is mis-classified as computer. CMU can detect all these images as open classes.

In Fig. 1, we show images that are classified correctly by both CMU and UAN. In general, images classified correctly by both domains are target domain samples of the common class. Although they are different in background and some details, they are overall very similar. Generally, such images can be easily classified correctly by both methods.

In Fig. 2, we show images that are classified correctly by CMU but wrongly by UAN. They are mostly from the open class images, but similar to some common classes. The tables were misclassified into beds, as their structure is homologous. The notebook was misclassified into clipboards, which are both the blue board but have different elements on the surface. Such images need a transferability measurement with stronger discriminability.

In Fig. 3, we show images that are classified wrongly by both CMU and UAN. This part mostly composes of images from open classes that are really like source classes but have different labels from source classes. For example, the soda image



(a) soda T (b) bottle S (c) telephone T (d) calculator S (e) monitor T (f) computer S

Fig. 3. Images classified incorrectly by both methods. Fig. (a) shows a soda in the target domain is mis-classified as bottle. Fig. (c) shows a telephone in the target domain is mis-classified as calculator. Fig. (e) shows a monitor in the target domain is mis-classified as computer. These open class images are quite similar to common classes so they are difficult to classify even by supervised learning.

can be really regarded as a bottle. A monitor is part of a computer. Such images are almost indistinguishable even by human beings.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
2. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.S.: Domain-adversarial training of neural networks. JMLR **17**, 59:1–59:35 (2016)
3. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Tech. rep., California Institute of Technology (2007)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
5. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems. pp. 6402–6413 (2017)
6. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. In: NeurIPS. pp. 136–144 (2016)
7. Panareda Busto, P., Gall, J.: Open set domain adaptation. In: ICCV (Oct 2017)
8. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1406–1415 (2019)
9. Peng, X., Usman, B., Kaushik, N., Wang, D., Hoffman, J., Saenko, K., Roynard, X., Deschaud, J.E., Goulette, F., Hayes, T.L.: VisDA: A synthetic-to-real benchmark for visual domain adaptation. In: CVPR Workshops. pp. 2021–2026 (2018)
10. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV (2010)
11. Saito, K., Yamamoto, S., Ushiku, Y., Harada, T.: Open set domain adaptation by backpropagation. In: ECCV (September 2018)
12. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: CVPR (2017)
13. You, K., Long, M., Cao, Z., Wang, J., Jordan, M.I.: Universal domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
14. Zhang, J., Ding, Z., Li, W., Ogunbona, P.: Importance weighted adversarial nets for partial domain adaptation. In: CVPR (June 2018)