# Semi-supervised Domain Adaptation via Minimax Entropy Supplementary Material

## 1. Datasets

First, we show the examples of datasets we employ in the experiments in Fig 1. We also attach a list of classes used in our experiments on DomainNet with this material.

## 2. Implementation Detail

We provide details of our implementation. **We will publish our implementation upon acceptance.** The reported performance in the main paper is obtained by one-time training. In this material, we also report both average and variance on multiple runs and results on different dataset splits (i.e., different train/val split).

**Implementation of MME.** For VGG and AlexNet, we replace the last linear layer with randomly initialized linear layer. With regard to ResNet34, we remove the last linear layer and add two fully-connected layers following [6]. We use the momentum optimizer where the initial learning rate is set 0.01 for all fully-connected layers whereas it is set 0.001 for other layers including convolution layers and batch-normalization layers. We employ learning rate annealing strategy proposed in [4]. Each mini-batch consists of labeled source, labeled target and unlabeled target images. Labeled examples and unlabeled examples are separately forwarded. We sample $s$ labeled source and labeled target images and $2s$ unlabeled target images. $s$ is set to be 32 for AlexNet, but 24 for VGG and ResNet due to GPU memory contraints. We use horizontal-flipping and random-cropping based data augmentation for all training images.

### 2.1. Baseline Implementation

Except for CDAN, we implemented all baselines by ourselves. **S+T** [3]. This approach only uses labeled source and target examples with the cross-entropy loss for training.

**DANN** [4]. We train a domain classifier on the output of the feature extractor. It has three fully-connected layers with relu activation. The dimension of the hidden layer is set 512. We use a sigmoid activation only for the final layer. The domain classifier is trained to distinguish source examples and unlabeled target examples.

**ADR** [8]. We put dropout layer with 0.1 dropout rate after l2-normalization layer. For unlabeled target examples, we calculate sensitivity loss and trained $C$ to maximize it whereas trained $F$ to minimize it. We also implemented $C$ with deeper layers, but could not find improvement.

**ENT**. The difference from MME is that the entire network is trained to minimize entropy loss for unlabeled examples in addition to classification loss.

**CDAN** [6]. We used the official implementation of CDAN provided in https://github.com/thuml/CDAN. For brevity, CDAN in our paper denotes CDAN+E in their paper. We changed their implementation so that the model is trained with labeled target examples. Similar to DANN, the domain classifier of CDAN is trained to distinguish source examples and unlabeled target examples.

## 3. Additional Results Analysis

**Results on Office-Home and Office.** In Table 1 and Table 2, we report all results on Office-Home and Office. In almost all settings, our method outperformed baseline methods.

**Sensitivity to hyper-parameter $\lambda$.** In Fig. 2, we show our method's performance when varying the hyper-parameter $\lambda$ which is the trade-off parameter between classification loss on labeled examples and entropy on unlabeled target examples. The best validation result is obtained when $\lambda$ is 0.1. From the result on validation, we set $\lambda$ 0.1 in all experiments.

**Changes in accuracy during training.** We show the learning curve during training in Fig 3. Our method gradually increases the performance whereas others quickly converges.

**Comparison with virtual adversarial training.** Here, we present the comparison with general semi-supervised learning algorithm. We select virtual adversarial training (VAT) [7] as the baseline because the method is one of the state-of-the art algorithms on semi-supervised learning and works well on various settings. The work proposes a loss called virtual adversarial loss. The loss is defined as the robustness of the conditional label distribution around each input data point against local perturbation. We add the virtual adversarial loss for unlabeled target examples in addition to classification loss. We employ hyper-parameters used in the original implementation because we could not see improvement in changing the parameters. We show the results in Table 3. We do not observe the effectiveness of VAT in

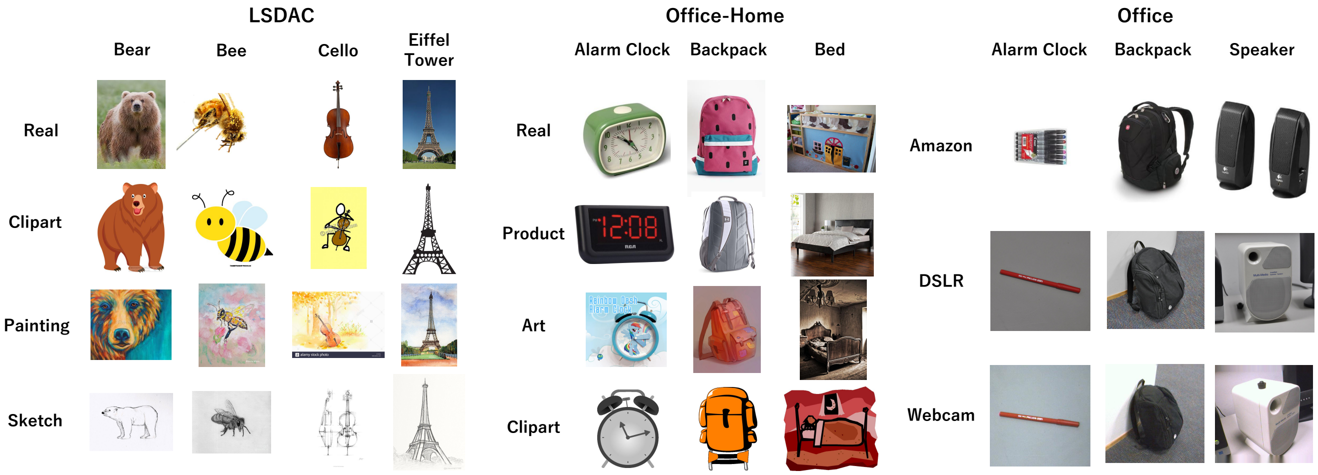


Figure 1: Example images in DomainNet, Office-Home, and Office.

| Network | Method | R to C | R to P | R to A | P to R | P to C | P to A | A to P | A to C | A to R | C to R | C to A | C to P | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | **One-shot** | | | | | | | |
| AlexNet | S+T | 37.5 | 63.1 | 44.8 | 54.3 | 31.7 | 31.5 | 48.8 | 31.1 | 53.3 | 48.5 | 33.9 | 50.8 | 44.1 |
| | DANN | **42.5** | 64.2 | 45.1 | 56.4 | 36.6 | 32.7 | 43.5 | 34.4 | 51.9 | 51.0 | 33.8 | 49.4 | 45.1 |
| | ADR | 37.8 | 63.5 | 45.4 | 53.5 | 32.5 | 32.2 | 49.5 | 31.8 | 53.4 | 49.7 | 34.2 | 50.4 | 44.5 |
| | CDAN | 36.1 | 62.3 | 42.2 | 52.7 | 28.0 | 27.8 | 48.7 | 28.0 | 51.3 | 41.0 | 26.8 | 49.9 | 41.2 |
| | ENT | 26.8 | 65.8 | 45.8 | 56.3 | 23.5 | 21.9 | 47.4 | 22.1 | 53.4 | 30.8 | 18.1 | 53.6 | 38.8 |
| | MME | 42.0 | **69.6** | **48.3** | **58.7** | **37.8** | **34.9** | **52.5** | **36.4** | **57.0** | **54.1** | **39.5** | **59.1** | **49.2** |
| VGG | S+T | 39.5 | 75.3 | 61.2 | 71.6 | 37.0 | 52.0 | 63.6 | 37.5 | 69.5 | 64.5 | 51.4 | 65.9 | 57.4 |
| | DANN | **52.0** | 75.7 | 62.7 | 72.7 | 45.9 | 51.3 | 64.3 | 44.4 | 68.9 | 64.2 | 52.3 | 65.3 | 60.0 |
| | ADR | 39.7 | 76.2 | 60.2 | 71.8 | 37.2 | 51.4 | 63.9 | 39.0 | 68.7 | 64.8 | 50.0 | 65.2 | 57.4 |
| | CDAN | 43.3 | 75.7 | 60.9 | 69.6 | 37.4 | 44.5 | 67.7 | 39.8 | 64.8 | 58.7 | 41.6 | 66.2 | 55.8 |
| | ENT | 23.7 | 77.5 | 64.0 | **74.6** | 21.3 | 44.6 | 66.0 | 22.4 | 70.6 | 62.1 | 25.1 | 67.7 | 51.6 |
| | MME | 49.1 | **78.7** | **65.1** | 74.4 | **46.2** | **56.0** | **68.6** | **45.8** | **72.2** | **68.0** | **57.5** | **71.3** | **62.7** |
| | | | | | | | **Three-shot** | | | | | | | |
| AlexNet | S+T | 44.6 | 66.7 | 47.7 | 57.8 | 44.4 | 36.1 | 57.6 | 38.8 | 57.0 | 54.3 | 37.5 | 57.9 | 50.0 |
| | DANN | 47.2 | 66.7 | 46.6 | 58.1 | 44.4 | 36.1 | 57.2 | 39.8 | 56.6 | 54.3 | 38.6 | 57.9 | 50.3 |
| | ADR | 45.0 | 66.2 | 46.9 | 57.3 | 38.9 | 36.3 | 57.5 | 40.0 | 57.8 | 53.4 | 37.3 | 57.7 | 49.5 |
| | CDAN | 41.8 | 69.9 | 43.2 | 53.6 | 35.8 | 32.0 | 56.3 | 34.5 | 53.5 | 49.3 | 27.9 | 56.2 | 46.2 |
| | ENT | 44.9 | 70.4 | 47.1 | 60.3 | 41.2 | 34.6 | 60.7 | 37.8 | 60.5 | 58.0 | 31.8 | 63.4 | 50.9 |
| | MME | **51.2** | **73.0** | **50.3** | **61.6** | **47.2** | **40.7** | **63.9** | **43.8** | **61.4** | **59.9** | **44.7** | **64.7** | **55.2** |
| VGG | S+T | 49.6 | 78.6 | 63.6 | 72.7 | 47.2 | 55.9 | 69.4 | 47.5 | 73.4 | 69.7 | 56.2 | 70.4 | 62.9 |
| | DANN | 56.1 | 77.9 | 63.7 | 73.6 | 52.4 | 56.3 | 69.5 | 50.0 | 72.3 | 68.7 | 56.4 | 69.8 | 63.9 |
| | ADR | 49.0 | 78.1 | 62.8 | 73.6 | 47.8 | 55.8 | 69.9 | 49.3 | 73.3 | 69.3 | 56.3 | 71.4 | 63.0 |
| | CDAN | 50.2 | 80.9 | 62.1 | 70.8 | 45.1 | 50.3 | 74.7 | 46.0 | 71.4 | 65.9 | 52.9 | 71.2 | 61.8 |
| | ENT | 48.3 | 81.6 | 65.5 | 76.6 | 46.8 | 56.9 | 73.0 | 44.8 | **75.3** | **72.9** | 59.1 | **77.0** | 64.8 |
| | MME | **56.9** | **82.9** | **65.7** | **76.7** | **53.6** | **59.2** | **75.7** | **54.9** | **75.3** | **72.9** | **61.1** | 76.3 | **67.6** |

Table 1: Results on Office-Home. Our method performs better than baselines in most settings.

SSDA. This could be due to the fact that the method does not consider the domain-gap between labeled and unlabeled examples. In order to boost the performance, it should be better to account for the gap.

**Analysis of Batch Normalization.** We investigate the effect of BN and analyze the behavior of entropy minimization and our method with ResNet. When training all models, unlabeled target examples and labeled examples are forwarded separately. Thus, the BN stats are calculated separately between unlabeled target and labeled ones. Some previous work [2, 5] have demonstrated that this operation can reduce domain-gap. We call this batch strategy as a "Separate BN". To analyze the effect of Separate BN, we compared this with a "Joint BN" where we forwarded unlabeled and labeled examples at once. BN stats are calculated jointly and Joint BN will not help to reduce domain-gap. We compare ours with entropy minimization on both Separate BN and Joint BN. Entropy minimization with Joint BN

| Network | Method | W to A |  | D to A |  |
|---|---|---|---|---|---|
|  |  | 1-shot | 3-shot | 1-shot | 3-shot |
| AlexNet | S+T | 50.4 | 61.2 | 50.0 | 62.4 |
|  | DANN | 57.0 | 64.4 | 54.5 | 65.2 |
|  | ADR | 50.2 | 61.2 | 50.9 | 61.4 |
|  | CDAN | 50.4 | 60.3 | 48.5 | 61.4 |
|  | ENT | 50.7 | 64.0 | 50.0 | 66.2 |
|  | MME | **57.2** | **67.3** | **55.8** | **67.8** |
| VGG | S+T | 69.2 | 73.2 | 68.2 | 73.3 |
|  | DANN | 69.3 | 75.4 | 70.4 | 74.6 |
|  | ADR | 69.7 | 73.3 | 69.2 | 74.1 |
|  | CDAN | 65.9 | 74.4 | 64.4 | 71.4 |
|  | ENT | 69.1 | 75.4 | 72.1 | 75.1 |
|  | MME | **73.1** | **76.3** | **73.6** | **77.6** |

Table 2: Results on Office. Our method outperformed other baselines in all settings.

performs much worse than Separate BN as shown in Table 4. This results show that entropy minimization does not reduce domain-gap by itself. On the other hand, our method works well even in case of Joint BN. This is because our training method is designed to reduce domain-gap.

**Comparison with SSDA methods [9, 1]** Since there are no recently proposed SSDA methods using deep learning, we compared with the state-of-the-art unsupervised DA methods modified for the SSDA task. We also compared our method with [9] and [1]. We implemented [9] and also modified it for the SSDA task. To compare with [1], we follow their evaluation protocol and report our and their best accuracy (see Fig. 3 (c)(f) in [1]). As shown in Table 7, we outperform these methods with a significant margin.

**Results on Multiple Runs.** We investigate the stability of our method and several baselines. Table 6 shows results averaged accuracy and standard deviation of three runs. The deviation is not large and we can say that our method is stable.

**Results on Different Splits.** We investigate the stability of our method for labeled target examples. Table 5 shows results on different splits. *sp0* correponds to the split we use in the experiment on our paper. For each split, we randomly picked up labeled training examples and validation examples. Our method consistently performs better than other methods.

| Method | R to C | R to P | P to C | C to P | C to S | S to P | R-S | P to R |
|---|---|---|---|---|---|---|---|---|
| S+T | 47.1 | 45.0 | 44.9 | 35.9 | 36.4 | 38.4 | 33.3 | 58.7 |
| VAT | 46.1 | 43.8 | 44.3 | 35.8 | 35.6 | 38.2 | 31.8 | 57.7 |
| MME | 55.6 | 49.0 | 51.7 | 40.2 | 39.4 | 43.0 | 37.9 | 60.7 |

Table 3: Comparison with VAT [7] using AlexNet on DomainNet. VAT does not perform bettern than S+T

| Method | Joint BN | Separate BN |
|---|---|---|
| ENT | 63.6 | 68.9 |
| MME | 69.5 | 69.6 |

Table 4: Ablation study of batch-normalization. The performance of the ENT method highly depends on the choice of BN while our method shows consistent behavior.

| Method | 1-shot |  |  | 3-shot |  |  |
|---|---|---|---|---|---|---|
|  | sp0 | sp1 | sp2 | sp0 | sp1 | sp2 |
| S+T | 43.3 | 43.8 | 43.8 | 47.1 | 45.9 | 48.8 |
| DANN | 43.3 | 44.0 | 45.4 | 46.1 | 43.1 | 45.3 |
| ENT | 37.0 | 32.9 | 38.2 | 45.5 | 45.4 | 47.8 |
| MME | **48.9** | **51.2** | **51.4** | **55.6** | **55.0** | **55.8** |

Table 5: Results on different training splits on DomainNet, Real to Clipart adaptation scenario using AlexNet.

| Method | 1-shot | 3-shot |
|---|---|---|
| CDAN | 62.9±1.5 | 65.3±0.1 |
| ENT | 59.5±1.5 | 63.6±1.3 |
| MME | **64.3**±0.8 | **66.8**±0.4 |

Table 6: Results on three runs on DomainNet, Sketch to Painting adaptation scenario using ResNet.

| AlexNet | R to C |  | AlexNet | D to A | W to A |
|---|---|---|---|---|---|
|  | 1-shot | 3-shot |  | 1-shot | 1-shot |
| DIRT-T [9] | 45.2 | 48.0 | GDSDA [1] | 51.5 | 48.3 |
| MME | **48.9** | **55.6** | MME | **58.5** | **60.4** |

Table 7: Comparison with [9, 1].

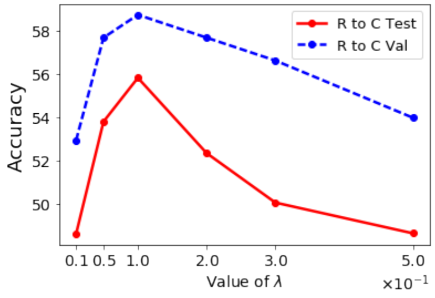
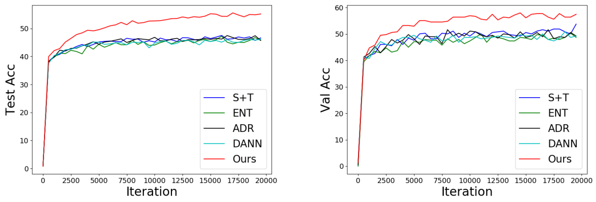


Figure 2: Sensitivity to hyper-parameter $\lambda$. The result is obtained when we use AlexNet on DomainNet, Real to Clipart.

## References

[1] Shuang Ao, Xiang Li, and Charles X Ling. Fast generalized distillation for semi-supervised domain adaptation. In *AAAI*, 2017. 3

[2] Fabio Maria Cariucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulò. Autodial: Automatic domain



(a) Test accuracy (b) Validation accuracy

Figure 3: Test and validation accuracy over iterations. Our method increases performances over iterations while others quickly converges. The result is obtained on Real to Clipart adaptation of DomainNet using AlexNet.

alignment layers. In *ICCV*, 2017. 2

[3] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv*, 2018. 1

[4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2014. 1

[5] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv*, 2016. 2

[6] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NIPS*, 2018. 1

[7] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv*, 2015. 1, 3

[8] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. In *ICLR*, 2018. 1

[9] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *ICLR*, 2018. 3