Unsupervised Domain Adaptation using Feature-Whitening and Consensus Loss: Supplementary Material

Subhankar Roy[1,2], Aliaksandr Siarohin[1], Enver Sangineto[1], Samuel Rota Bulò[3], Nicu Sebe[1] and Elisa Ricci[1,2]
[1]DISI, University of Trento, Italy, [2]Fondazione Bruno Kessler, Trento, Italy, [3]Mapillary Research
{subhankar.roy, aliaksandr.siarohin, enver.sangineto, niculae.sebe, e.ricci}@unitn.it
samuel@mapillary.com

1. Computing the whitening matrix

The whitening matrix W_B in Eq. (3) of the main paper can be computed in different ways. For instance, Huang et al. [6] use the ZCA whitening [7], while Siarohin et al. [11] use the Cholesky decomposition [2]. Both tecniques are unique (given a covariance matrix) and differentiable, however we adopted the method proposed in [11] because it is faster [13] and more stable [11] than the ZCA-based whitening. Moreover, many modern platforms for deep-network developing include tools for computing the Cholesky decomposition, thus this solution makes our approach easier to be reproduced.

We describe below the main steps we used to compute W_B. Since W_B^s and W_B^t, respectively used in Eq. (4) and in Eq. (5) of the main paper, and depending on B^s and B^t, are computed exacly in the same way, in the following we refer to the generic matrix W_B in Eq. (3) which depends on the batch statistics $\Omega = (\boldsymbol{\mu}_B, \Sigma_B)$.

The first step consists in computing the covariance matrix Σ_B. To avoid instability issues, we blend the empirical covariance matrix $\hat{\Sigma}_B$ with E, the identity matrix [10]:

$$\Sigma_B = (1-\epsilon)\hat{\Sigma}_B + \epsilon E, \tag{1}$$

where:

$$\hat{\Sigma}_B = \frac{1}{m-1}\sum_{i=1}^{m}(\mathbf{x}_i - \boldsymbol{\mu}_B)(\mathbf{x}_i - \boldsymbol{\mu}_B)^\top. \tag{2}$$

Once Σ_B is computed, we use the approach proposed in [11] to compute W_B such that $W_B^\top W_B = \Sigma_B^{-1}$:

1. Let $TT^\top = \Sigma_B$, where T is a lower triangular matrix.
2. Using the Cholesky decomposition we compute T and $T^\top$ from Σ_B.
3. We invert T and we obtain: $W_B = T^{-1}$.

For more details, we refer to [11].

2. Relation between the MEC loss and the Entropy and the Consistency losses

We show below a formal relation between our MEC loss and the Entropy and the Consistency losses.

Proposition 1. *Let $\mathcal{F} \subset \mathcal{X} \to \mathcal{Y}$ be an hypothesis space of predictors of infinite capacity. Then the minimization of the consensus loss L^t yields a predictor that is consistent,* i.e. *$p(\cdot|\mathbf{x}_i^{t_1}) = p(\cdot|\mathbf{x}_i^{t_2})$ for any pairs of perturbed datapoints $(\mathbf{x}_i^{t_1}, \mathbf{x}_i^{t_2})$ and confident,* i.e. *$p(y|\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{X}$ and some $y \in \mathcal{Y}$ depending on $\mathbf{x}$.*

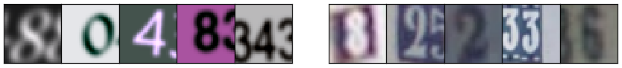
Proof. The pointwise loss $\ell^t(\mathbf{x}_i^{t_1}, \mathbf{x}_i^{t_2})$ is lower bounded by 0 and it attains 0 if and only if the conditions on p listed in the theorem are satisfied. The result follows noting that predictors of infinite capacity can always attain 0 loss. □

3. Additional experiments using synthetic-to-real adaptation settings

In this section we report results of additional UDA experiments using synthetic *source* images and *real* target images and we compare our method with the state-of-the-art approaches in these settings.

3.1. Datasets and experimental setup

Synthetic numbers → **SVHN**. It is a common practice in UDA to train a predictor on annotated synthetic images and then test on real images. In this setting we use the SYN NUMBERS [4] as the source dataset and SVHN [8] as the target dataset. The former (SYN NUMBERS) is composed of images which are software-generated (e.g., using different orientations, stroke colors, etc.), in order to simulate the latter (SVHN). Despite some geometric similarities between the two datasets, there exists a significant domain shift between them because, for instance, the cluttered background in SVHN, which is absent in SYN NUMBERS images (see Fig. 1 (a)). There are approximately 500,000 annotated images in the SYN NUMBERS dataset.



(a) SYN NUMBERS → SVHN

(b) SYN SIGNS → GTSRB

Figure 1: Samples from Synthetic Images dataset (source) and Real Image dataset (target)

Synthetic Signs → **GSTRB**. In this setting, which is analogous to the SYN NUMBERS → SVHN experiment, the source dataset (SYN SIGNS [4]) is composed of synthetic traffic signs, while the target dataset is the German Traffic Sign Recognition Benchmark (GTSRB [12]). The SYN SIGNS dataset is composed of 100,000 synthetic images belonging to 43 different traffic signs categories, while the GTSRB dataset is composed of 39,209 real images, partitioned using the same 43 categories. As shown in Fig. 1 (b), the real target domain exhibits a domain shift because of different illumination conditions, background clutter, etc.

In the experiments conducted on both settings we adopt the standard evaluation protocols and the corresponding training/testing splits [4], using identical experimental setups as reported in Sec. 4.2 of the main paper.

3.2. Comparison with state-of-the-art methods

In Tab 1 we report the results of our method compared with other UDA methods. We compare with the following baselines: Domain-Adversarial Training of Neural Networks (DANN) [4], Asymmetric tri-training (ATT) [9], Associative Domain Adaptation (ADA) [5], AutoDIAL [1] and Self-Ensembling (SE) [3]. The results of most of the methods reported in Tab. 1 are taken from the original papers. In the same table we also show SE and AutoDIAL results obtained using comparable base-network architectures as those used by our method. Moreover, similarly to the main paper, and for a fair comparison, we split Tab. 1 into two sections in order to differentiate the methods which use data augmentation from those methods which do not exploit data augmentation.

When DWT is compared with the methods using no-data augmentation, it outperforms all the baselines in both the SYN NUMBERS → SVHN and the SYN DIGITS → GTSRB setting. When data augmentation is considered, DWT-MEC outperforms all the other approaches in the second setting but performs worse by 1% when compared with SE [3] in the first setting. The superior performance of SE in SYN NUMBERS → SVHN can be attributed to the use of a very conservative threshold on the target predictions, which helps to filter-out noisy predictions during training. However, as demonstrated in Sec 4.3.1 of the main paper (Tab. 2), the absence of a confidence threshold, tuned on the specific setting, might lead SE to a drastic performance degradation.

Method	Source Target	Syn Numbers SVHN	Syn Signs GTSRB
Source Only		86.7±0.8	80.6±0.6
w/o augmentation			
DANN [4]		91.0	88.6
ATT [9]		92.9	96.2
ADA [5]		91.8	97.6
AutoDIAL † [1]		87.9	97.8
DWT		**93.70**±0.21	**98.11**±0.13
Target Only		95.62	98.49
w/ augmentation			
SE † a [3]		91.92±0.09	97.73±0.10
SE † b [3]		**95.62**±0.12	99.01±0.04
DWT-MEC		94.62±0.13	**99.30**±0.07
DWT-MEC (MT)		94.10±0.21	99.22±0.16

Table 1: Accuracy (%) using Synthetic image → Real image settings.[*] denotes values extracted from [3]; [a] means minimal augmentation; [b] means full augmentation of both the source and the target data; and † denotes methods using base networks which are identical to our proposed method.

4. CNN Architectures

In this section we report the network architectures used in all the small-image experiments shown in both the main paper and in this Supplementary Material (Tab. 2, 3, 4, 5).

Description
Input: 28×28
Conv $5 \times 5 \times 32$, pad 2
Max-pool 2×2, stride 2
Conv $5 \times 5 \times 48$, pad 2
Max-pool 2×2, stride 2
Fully connected, 100 units
Fully connected, 100 units
Fully connected, 10 units, softmax

Table 2: MNIST ↔ USPS base architecture as used in [4].

References

[1] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò. Autodial: Automatic domain alignment layers. In *ICCV*, 2017.

[2] D. Dereniowski and K. Marek. Cholesky factorization of matrices in parallel and ranking of graphs. In *5th Int. Conference on Parallel Processing and Applied Mathematics*, 2004.

Description
Input: 32 × 32 × 3
Conv 5 × 5 × 64, pad 2
Max-pool 3 × 3, stride 2
Conv 5 × 5 × 64, pad 2
Max-pool 3 × 3, stride 2
Conv 5 × 5 × 128, pad 2
Fully connected, 3072 units
Dropout, 50%
Fully connected, 2048 units
Dropout, 50%
Fully connected, 10 units, softmax

Table 3: SVHN ↔ MNIST and SYN NUMBERS ↔ SVHN base architecture as used in [4].

Description
Input: 32 × 32 × 3
Conv 3 × 3 × 128, pad 1
Conv 3 × 3 × 128, pad 1
Conv 3 × 3 × 128, pad 1
Max-pool 2 × 2, stride 2
Dropout, 50%
Conv 3 × 3 × 256, pad 1
Conv 3 × 3 × 256, pad 1
Conv 3 × 3 × 256, pad 1
Max-pool 2 × 2, stride 2
Dropout, 50%
Conv 3 × 3 × 512, pad 0
Conv 1 × 1 × 256, pad 0
Conv 1 × 1 × 128, pad 0
Global Average Pooling
Fully connected, 9 units, softmax

Table 4: CIFAR-10 ↔ STL base architecture as used in [3].

Description
Input: 40 × 40 × 3
Conv 5 × 5 × 96, pad 2
Max-pool 2 × 2, stride 2
Conv 3 × 3 × 144, pad 1
Max-pool 2 × 2, stride 2
Conv 5 × 5 × 256, pad 2
Max-pool 2 × 2, stride 2
Fully connected, 512 units
Dropout, 50%
Fully connected, 43 units, softmax

Table 5: SYN SIGNS ↔ GTSRB base architecture as used in [4].

[3] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for visual domain adaptation. *ICLR*, 2018.

[4] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[5] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers. Associative domain adaptation. In *ICCV*, 2017.

[6] L. Huang, D. Yang, B. Lang, and J. Deng. Decorrelated batch normalization. In *CVPR*, 2018.

[7] A. Kessy, A. Lewin, and K. Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 2017.

[8] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, 2011.

[9] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. *arXiv:1702.08400*, 2017.

[10] J. Schfer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1), 2005.

[11] A. Siarohin, E. Sangineto, and N. Sebe. Whitening and Coloring transform for GANs. *arXiv:1806.00420*, 2018.

[12] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

[13] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.