Supplementary Material for:

Unsupervised Visual Domain Adaptation:
A Deep Max-Margin Gaussian Process Approach

Minyoung Kim[1,2,3], Pritish Sahu[1], Behnam Gholami[1], and Vladimir Pavlovic[1,3]

[1]Dept. of Computer Science, Rutgers University, NJ, USA
[2]Dept. of Electronic Engineering, Seoul National University of Science & Technology, South Korea
[3]Samsung AI Center, Cambridge, UK
mikim21@gmail.com, {ps851,bb510,vladimir}@cs.rutgers.edu, v.pavlovic@samsung.com

1. Overview

In this Supplement, we present additional analyses highlighting the ability of our model, GPDA, to leverage its inherent measure of uncertainty to both produce increasingly accurate predictions as well as provide a measure of its own trustworthiness. These new results are summarized in Sec. 2. Sec. 3 provides further analysis showing the key connection between GPDA and the max-margin Gaussian Process classification in the original space $\mathcal{X}$, surpassing the explicit need for a shared space $\mathcal{Z}$ of traditional domain adaptation approaches. We then present specific details of all datasets used in our experiments as well as the particulars of relevant experimental setups in Sec. 4. Additional experimental results on the Office-31 dataset [14] are in Sec. 5. We discuss computational complexity of GPDA compared to MCDA in Sec. 6. Finally, we provide a brief overview of Gaussian Process models in Sec. 7 and another related state-of-the-art domain adaptation approach, the MCDA, in Sec. 8.

2. Additional Analysis: Prediction Uncertainty vs. Prediction Quality

A key benefit of our GPDA algorithm, inherited from Bayesian modeling, is that it provides a quantified measure of prediction uncertainty. In the multi-class setup, for an input $\mathbf{x}$ we measure the uncertainty as the degree of overlap between the two largest mean posteriors, $p(f_{j^*}(\mathbf{z})|\mathcal{D}_S)$ and $p(f_{j^\dagger}(\mathbf{z})|\mathcal{D}_S)$, where $\mathbf{z} = \mathbf{G}(\mathbf{x})$, j^* and $j^\dagger$ are the indices of the largest and the second largest among the posterior means $\{\mu_j(\mathbf{z})\}_{j=1}^K$, respectively, If the two overlap significantly, our model's decision is less certain, signifying that we anticipate the class prediction not to be trustworthy. On the other hand, if the two are well separated, we expect high prediction quality.

Bhattacharyya distance. In the main paper (Sec. 5.4 and Fig. 4), we have verified this hypothesis by evaluating the Bhattacharyya distances (BD) between two posteriors (i.e., a measure of *certainty* in prediction) for two different cohorts: correctly classified test target samples by our model and incorrectly predicted ones, for the **SVHN** to **MNIST** adaptation task. Since we use variational Gaussian approximation of the posteriors $p(f_j(\mathbf{z})|\mathcal{D}_S) \approx \mathcal{N}(\mu_j(\mathbf{z}), \sigma_j(\mathbf{z})^2)$, where $\mu_j(\mathbf{z})$ and $\sigma_j(\mathbf{z})$ are determined by Eq. (22) in the main paper, the Bhattacharyya distance can be computed in closed form:

$$\text{BD} = \frac{1}{4}\log\left(\frac{1}{4}\left(\frac{\sigma_{j^*}^2}{\sigma_{j^\dagger}^2} + \frac{\sigma_{j^\dagger}^2}{\sigma_{j^*}^2} + 2\right)\right) + \frac{1}{4}\left(\frac{(\mu_{j^*} - \mu_{j^\dagger})^2}{\sigma_{j^*}^2 + \sigma_{j^\dagger}^2}\right). \tag{1}$$

Pritish Sahu and Behnam Gholami contributed equally to this work.

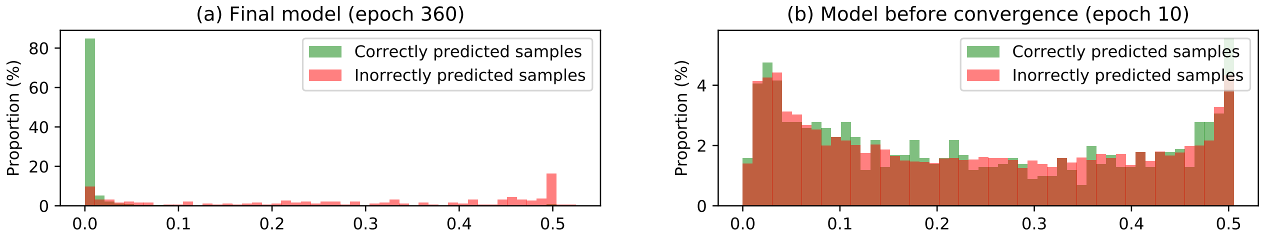


Figure 1: (For our GPDA) Histograms of Bayes error rates (prediction uncertainty) for our two models: (a) after convergence, (b) at an early stage of training. The X-axis is the Bayes optimal error rate (2) b/w two largest mean posteriors, an indication of *prediction uncertainty*; the higher the error rate, the more uncertain the prediction is. For each model, we compute histograms of correctly and incorrectly predicted samples separately (by color). In our final model (a), there is a strong correlation between prediction uncertainty (horizontal axis) and prediction correctness (color).

Bayes Optimal Error Rate. An alternative metric to measure the prediction uncertainty, perhaps more principled in the Bayesian sense, is the Bayes optimal error rate between the two largest mean posteriors, which can be computed as

$$\text{Bayes error} = \frac{1}{2}\int_{D}^{\infty} \mathcal{N}(x; \mu_{j^\dagger}, \sigma_{j^\dagger}^2)\, dx + \frac{1}{2}\int_{-\infty}^{D} \mathcal{N}(x; \mu_{j^*}, \sigma_{j^*}^2)\, dx = \frac{1}{2}\left(\Phi\left(\frac{\mu_{j^\dagger} - D}{\sigma_{j^\dagger}}\right) + \Phi\left(\frac{D - \mu_{j^*}}{\sigma_{j^*}}\right)\right), \quad (2)$$

where Φ is the CDF of $\mathcal{N}(0,1)$ and D is the Bayes optimal decision threshold, $D = (\mu_{j^*} - \mu_{j^\dagger})/\sqrt{(\sigma_{j^*}^2 + \sigma_{j^\dagger}^2)/2}$. The interpretation is: the smaller the Bayes error rate, the more certain our prediction is, and vice versa. We depict the histograms of the Bayes error rates for two cohorts in Fig. 1. As shown, the conclusion is very similar to our earlier analysis based on Bhattacharyya distances: Our final model in Fig. 1(a) exhibits low error rates for most of the samples in the correctly predicted cohort (green), implying well separated posteriors or high certainty of prediction. For the incorrectly predicted samples (red), the error rates are mostly high implying significant overlap between the two posteriors, i.e., high uncertainty of prediction.

Uncertainty in GPDA vs MCDA. Lastly, to demonstrate that it is the unique property of our GPDA model that the uncertainty measure can be used to credibly gauge the quality of prediction at test time, we contrast our model with other non-Bayesian approaches. Specifically, we consider MCDA, as the second-best competing method. The MCDA is a non-Bayesian method that yields *point estimate* class prediction, namely $p(y|\mathbf{x})$. By point estimate, we mean that the MCDA prediction places all its probability mass on a single (softmax) probability (score) value $p(y = j|\mathbf{x})$ for each class j, unable to provide a degree of uncertainty in its prediction (e.g., σ_j in our GPDA model).

However, one can define a *heuristic* notion of uncertainty for the MCDA by measuring how distant the two largest score predictions are from each other. More specifically, we compute the following quantity, dubbed Bhattacharyya *pseudo* distance (BPD), as a measure of uncertainty in the MCDA:

$$\text{BPD} := \log p(y = j^*|\mathbf{x}) - \log p(y = j^\dagger|\mathbf{x}) \quad (3)$$

where j^* and $j^\dagger$ are the indices of the largest and the second largest among the scores $\log p(y = j|\mathbf{x})$, respectively. Note that (3) is the log-ratio between the largest two class prediction scores. We name it the *pseudo* distance as it reduces to the Bhattacharyya distance if we form Gaussians with the mean equal to $\log p(y = j|\mathbf{x})$ and the same variances for both j^* and $j^\dagger$.

We depict the histograms of the pseudo distances for MCDA's two cohorts in Fig. 2(b), where the Bhattacharyya histograms for our GPDA are also shown in Fig. 2(a) for comparison. Unlike the more clear separation attained in our GPDA model, the MCDA exhibits two issues: i) For the correctly predicted samples (green), a considerable number of points have large overlap between j^* and $j^\dagger$ (i.e., low BPDs). ii) For the incorrectly predicted samples (red), the number of cases where the two largest scores are relatively well separated[1] exceeds that of our GPDA model, suggesting higher prediction uncertainty. This signifies the unique benefit of our Bayesian domain adaptation approach, that is, the *capability to utilize the prediction uncertainty as a gauge of prediction quality*.

[1] E.g., those with BPD > 1.0, namely, certain predictions.

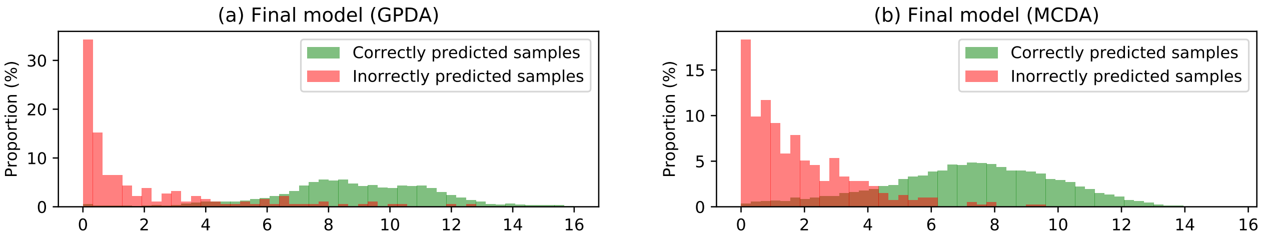
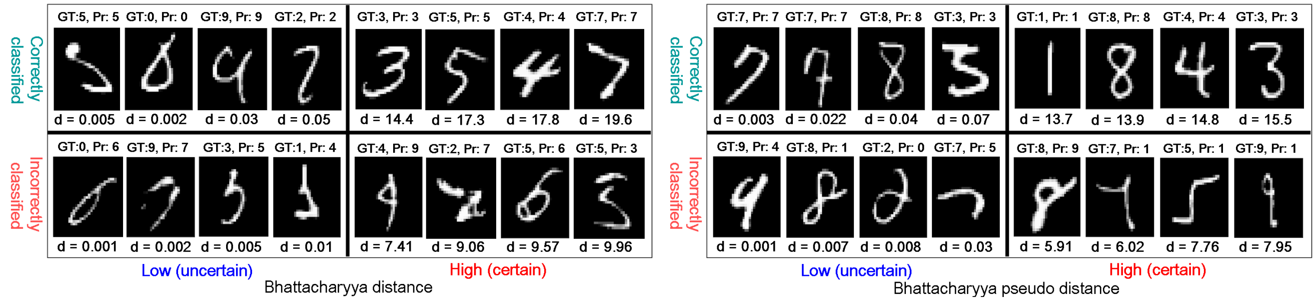


Figure 2: (GPDA vs. MCDA) Histograms of Bhattacharyya distances between two largest mean posteriors (prediction certainty) for (a) GPDA and (b) MCDA. The X-axis is the Bhattacharyya distance, an indication of *prediction certainty*; the higher the distance, the more certain the prediction is. For the non-Bayesian point-estimate-based MCDA, we compute the Bhattacharyya pseudo distance instead, as described in the text. Qualitatively, our GPDA model exhibits stronger correlation (histograms less overlapped) between prediction uncertainty (horizontal axis) and prediction correctness (color).



(a) GPDA

(b) MCDA

Figure 3: Selected test (**MNIST**) images according to the Bhattacharyya (pseudo) distances estimated by (a) GPDA and (b) MCDA. For each figure, Left: samples with low distances (uncertain prediction). Right: high distances (certain prediction). Top: correctly classified by the model. Bottom: incorrectly classified. For each image, GT, Pr, and d stand for ground-truth label, predicted label, and the (pseudo) distance, respectively.

GPDA vs. MCDA – Hard vs. Easy Instances. As a counterpart to Fig. 5 in the main paper, we also depict in Fig. 3(b) some sample target test images that are correctly/incorrectly predicted by the MCDA with low/high certainty according to the BPD. For ease of comparison, we also show the samples for our GPDA from the main paper, Fig. 5, in Fig. 3(a). Unlike the GPDA, the uncertainty prediction made by the MCDA shows less agreement with the human assessment. Images whose BPDs are low (i.e., uncertain prediction judged by the MCDA shown in the left panel of Fig. 3(b)), appear to be visually easy to classify by a human, with no ambiguity, with a possible exception in few cases: e.g., the last example in the correct/low quadrant that may look like ”five”, while the first example in the incorrect/low quadrant may be interpreted as ”four”. Furthermore, the sample images in the incorrect/high quadrant of Fig. 3(b), i.e., those predicted by the MCDA with high certainty but misclassified, are relatively easy-to-classify examples for a human, other than the second example that may be confused as ”one”.

This empirical analysis verifies that the measure of prediction uncertainty provided by our GPDA model can be used as a more accurate indicator of prediction quality than that implied by the MCDA, our top competitor. That is, our model's quantitative uncertainty measure can determine, with high precision, whether the prediction made by the model is trustworthy or not.

3. A Remark on Proposed GPDA Algorithm

In this section we discuss the strong connection between the GPDA algorithm and the max-margin confident prediction (or the entropy minimization) framework in classical semi-supervised learning [6, 18]. More specifically, we show that our GPDA algorithm, in the algorithmic point of view, can be viewed as a *max-margin Gaussian process classifier* on the original input space $\mathcal{X}$ without explicitly considering a shared space $\mathcal{Z}$.

Recall that the GPDA algorithm can be summarized as the following two alternating optimizations:

- $\min_{\{\mathbf{m}_j, \mathbf{S}_j\}}\ -\mathrm{LL} + \mathrm{KL}$ (variational inference)
- $\min_{\mathbf{G}, k}\ -\mathrm{LL} + \mathrm{KL} + \lambda \cdot \mathrm{MS}$ (model selection)

where the key terms in these objectives are defined as follows:

$$\mathrm{KL} = \frac{1}{2} \sum_{j=1}^{K} \big(\, \mathrm{Tr}(\mathbf{S}_j) + ||\mathbf{m}_j||_2^2 - \log\det(\mathbf{S}_j) - d \,\big), \tag{4}$$

$$\mathrm{LL} = \frac{1}{M} \sum_{m=1}^{M} \frac{N_S}{|B_S|} \sum_{i \in B_S} \log P(y_i^S | \mathbf{W}^{(m)} \boldsymbol{\phi}(\mathbf{z}_i^S)), \tag{5}$$

and

$$\mathrm{MS} := \frac{1}{|B_T|} \sum_{i \in B_T} \left(\max_{j \neq j^*} \mathbf{m}_j^\top \boldsymbol{\phi}(\mathbf{z}_i^T) - \max_{1 \leq j \leq K} \mathbf{m}_j^\top \boldsymbol{\phi}(\mathbf{z}_i^T) \ + \ 1 \ + \ \alpha \max_{1 \leq j \leq K} \big(\boldsymbol{\phi}(\mathbf{z}_i^T)^\top \mathbf{S}_j \boldsymbol{\phi}(\mathbf{z}_i^T) \big)^{1/2} \right)_+ . \tag{6}$$

Note that $\mathbf{z} = \mathbf{G}(\mathbf{x})$. Although we have built a **GP** classification model on top of the shared space $\mathcal{Z}$, leading to the algorithm above, in our learning objective terms (4–6), the deep kernel feature mapping $\boldsymbol{\phi}(\cdot)$ and the embedding function $\mathbf{G}(\cdot)$ always appear together in the composite form $\boldsymbol{\phi}(\mathbf{G}(\cdot))$.

Thus, our approach is functionally equivalent to building a **GP** classification model on top of the original $\mathcal{X}$ space, where the explicit feature mapping is $\mathbf{x} \to (\boldsymbol{\phi} \circ \mathbf{G})(\mathbf{x})$. More formally, our classifier can be written as $\mathbf{f}(\mathbf{x}) = \mathbf{W}\boldsymbol{\phi}(\mathbf{G}(\mathbf{x}))$, a function of $\mathbf{x}$. Consequently, our approach can be viewed as a *max-margin Gaussian process classifier*, without explicitly considering the shared space, where we push the posterior inferred from the source domain data to meet the large margin criterion on the (unlabeled) target domain data. This is clearly in line with *entropy minimization* or *max-margin confident prediction* principles in classical semi-supervised learning [6, 18].

4. Details of Datasets and Experimental Setups

We now present additional details of experiments on the three datasets used in the main paper. For all experiments, we set $M = 50$, the number of posterior samples from the variational density $q(\mathbf{W})$ (Sec. 3.2 in the main paper for more details).

4.1. Digit and Traffic Signs Datasets

We followed the experimental setup used in [4] in the following three adaptation scenarios. For this experiment, we compare our GPDA model with various state-of-the-art unsupervised domain adaptation approaces, namely: **MMD** [11], **DANN** [4], **DSN** [2], **ADDA** [17], **CoGAN** [10], **PixelDA** [1], **ATDA** [15], **ASSC** [7], **DRCN** [5], and, **MCDA** [16].

- **SVHN→MNIST.** In this adaptation scenario, we used the standard training set as our training samples, and the standard testing set as our testing samples both for source and target samples.
- **SYN SIGNS→GTSRB.** Following **MCDA** [16], we randomly selected 31367 samples for the target training set and evaluated the accuracy on the remaining samples.
- **MNIST↔USPS.** For this experiment, we followed the protocols used in ADDA [17] and PixelDA [1]. ADDA provides the setting where a part of training samples are utilized during training. 2,000 training samples are picked up for MNIST and 1,800 samples are used for USPS. PixelDA allows one to utilize all of the standard training samples during learning.

4.2. VisDA Dataset

We used VisDA dataset [13] to evaluate adaptation from synthetic to real-object images. The dataset is an instance of cross-domain object classification, with over 280K images across 12 categories in the combined training, validation, and testing domains. The source images, 152,397 synthetic images, were generated by rendering 3D models of the same object categories as in the real data from different angles and under different lighting conditions. The validation set of 55,388 images was collected from MSCOCO [9]. In our experiment, we considered the images of validation splits as the target domain and trained models in the unsupervised domain adaptation settings. We evaluate the performance of ResNet101 [8] model pre-trained on Imagenet [3]. For this experiment, we compare our model with **MMD** [11], **DANN** [4], and **MCDA** [16].

Table 1: Results on the Office-31 dataset with the evaluation setup of [12] (pre-trained ResNet-50 as the encoder network).

METHOD	A → W	A → D	W → A	W → D	D → A	D → W
JAN [12]	85.4	84.7	70.0	99.8	68.6	96.7
MCDA	77.36	84.14	67.24	98.44	71.03	96.10
GPDA	**83.93**	**85.54**	**68.84**	**100.00**	**72.31**	**97.26**
Src. only	76.15	78.34	63.64	99.21	63.78	92.69

5. Results on Office-31 dataset

Although relatively small and not much considered in recent DA work, we test our approach on the Office-31 dataset [14], which is comprised of $4,652$ images of objects belonging to 31 categories collected from three distinct domains: Amazon (A), Webcam (W), and DSLR (D). Following the standard settings [12], we evaluate DA tasks of: A → W, A → D, D → W, and all the other directions. We followed the evaluation setups of [12], employing the pre-trained ResNet-50 model as the encoder network. The results of our GPDA, the competing MCDA and JAN [12], and the baseline source-only trained model are summarized in Tab. 1. As shown, our approach consistently outperforms the competing models for all adaptation tasks.

6. Complexity of GPDA Training

In the sense of model complexity, both GPDA and MCDA are *functionally and structurally* equivalent: their encoder networks are identical and the GP classifier networks in our GPDA correspond to the two adversarial classifier networks F_1 and F_2 in MCDA. In training complexity, GPDA has only a slight overhead over MCDA, a constant M times due to Monte-Carlo sampling $\mathbf{W}$ from $q(\mathbf{W})$, where $\mathbf{W}$ is the last layer of the GP classifier network. That is,

$$\mathbf{w}_j^{(m)} = \mathbf{m}_j + \mathbf{S}_j^{1/2}\boldsymbol{\epsilon}_j^{(m)}, \qquad \boldsymbol{\epsilon}_j^{(m)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{7}$$

This typically translates into the factor of no more than two.

7. Background – Gaussian Process

A Gaussian Process (GP) is an infinite collection of random variables $\{f(\mathbf{x})|\mathbf{x} \in X\}$, such that any finite number of samples have a joint Gaussian distribution. A GP is fully specified by the mean function $\mu(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$, typically user-defined. GPs can also be interpreted as a distribution over functions $f(\mathbf{x}) \sim \mathcal{GP}(\mu(x), k(x, x))$ such that any finite collection of function values $[f(\mathbf{x_1}), \ldots, f(\mathbf{x_N})]$ have a joint Gaussian distribution:

$$[f(\mathbf{x_1}), \ldots, f(\mathbf{x_N})] \sim \mathcal{N}(\boldsymbol{\mu}, K), \tag{8}$$

where $\boldsymbol{\mu}$ is the $N \times 1$ vector $\mu_i = \mu(\mathbf{x_i})$ and K is the $N \times N$ covariance matrix with $K_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$.

A training dataset consists of N pairs of data $(\mathbf{x_i}, y_i)_{i=1}^N$, where y_i are noisy observations of some latent function f with Gaussian noise $y_i = f(\mathbf{x}_i) + \epsilon_i$, $\epsilon_i \in \mathcal{N}(0, \sigma^2)$. The likelihood of the data $\mathbf{y}|\mathbf{f} \sim \mathcal{N}(f, \sigma^2 I)$ and the prior $\mathbf{f} \sim \mathcal{N}(0, K)$ give the joint probability model $p(\mathbf{f}, \mathbf{y}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f})$, where $\mathbf{y}$ denotes the noisy targets and $\mathbf{f}$ denotes the vector of underlying latent function values. The predictive distribution at a set of test points X_* is given in closed form using the properties of conditional Gaussians,

$$\begin{aligned} \mathbf{f}_*|\mathbf{y}, X, X_*, \boldsymbol{\theta}, \sigma^2 &\sim \mathcal{N}(\mathbf{f}_*, cov(\mathbf{f}_*)) \\ \mathbf{f}_* &= K_*(K + \sigma^2 I)^{-1}\mathbf{y} \\ cov(\mathbf{f}_*) &= K_{**} - K_*(K + \sigma^2 I)^{-1}K_*^T, \end{aligned} \tag{9}$$

where K_{**} denotes the covariance matrix evaluated among the test inputs X_* and K_* denotes the covariance matrix evaluated between the test points X_* and the training set X. If there are N_* test points, the covariance matrix K_{**} is of size $N_* \times N_*$ and K_* is of size $N_* \times N$.

7.1. Gaussian Process Classification

In Gaussian Process Classification (GPC), the target values are discrete class labels, hence it is not appropriate to model them via a multivariate Gaussian density. Instead, we use the Gaussian process as a latent function whose sign determines the class label for binary classification; for multi-class classification one can use multiple GPs or a multivariate GP.

The key difference between the GP regression and GPC is how the output data, $\mathbf{y}$, are connected to the underlying function values, $\mathbf{f}$. Precisely, they are no longer connected via a simple noise process as in the previous section, instead now discrete: for example, for binary classification framework, say $y = 1$ for one class and $y = -1$ for the other. In this case, one could try fitting a GP that produces an output of 1 for some values of x and -1 for others, simulating the discrete nature of the problem. Then, the classification of a new data point x_* involves two steps:

1. Evaluate a 'latent function' f which models qualitatively how the likelihood of one class versus the other changes over the x axis. This is the usual GP.
2. Squeeze the output of this latent function onto $[0, 1]$ using logistic function, $\pi(f) = \sigma(y = 1|f)$.

Writing these two steps schematically,

$$\boxed{\text{data, } x_*} \xrightarrow{\text{GP}} \boxed{\text{latent function, } f_*|x_*} \xrightarrow{\text{sigmoid}} \boxed{\text{class probability, } \pi(f_*)}.$$

8. Background – MCDA [16]

For multi-class classification, the **MCDA** adopts classifier networks that output class prediction probabilities, $h(\mathbf{z}) = [p(y = 1|\mathbf{z}), \ldots, p(y = K|\mathbf{z})]^\top$. The discrepancy between h and h' is defined as the expected normalized L_1 difference, that is, $\mathbb{E}||h(\mathbf{z}) - h'(\mathbf{z})||_1/K$. The learning algorithm is a coordinate descent optimization alternating among three steps:

1. $\min_{G,h,h'} L_S := \mathbb{E}_{(\mathbf{x},y)\sim S}\big[\, CE(y; h(\mathbf{G}(\mathbf{x}))) \;+\; CE(y; h'(\mathbf{G}(\mathbf{x})))\,\big]$
2. (Fix G) $\min_{h,h'} L_S - L_{adv}$, where $L_{adv} := \mathbb{E}_{\mathbf{x}\sim T}\big[\, ||h(\mathbf{G}(\mathbf{x})) - h'(\mathbf{G}(\mathbf{x}))||_1/K \,\big]$
3. (Fix h, h') $\min_G L_{adv}$

Here, $CE(y; p)$ stands for the cross entropy (or log) loss, i.e., $CE(y; p) = -\log p(y)$. All the expectations are approximately estimated on a mini-batch. Optionally, Step-3 can be repeated $2 \sim 4$ times (on the same mini-batch) to boost the convergence of the embedding network $\mathbf{G}(\cdot)$.

References

[1] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 7, 2017. 4

[2] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 343–351, 2016. 4

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009. 4

[4] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *International Conference on Machine Learning (ICML)*, 2015. 4

[5] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Euroupean Conference on Computer Vision (ECCV)*, pages 597–613, 2016. 4

[6] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization, 2004. In Proc. of Advances in Neural Information Processing Systems. 3, 4

[7] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers. Associative domain adaptation. In *International Conference on Computer Vision (ICCV)*, volume 2, page 6, 2017. 4

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[10] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 469–477. Curran Associates, Inc., 2016. 4

[11] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *International Conference on Machine Learning (ICML)*, 2015. 4

[12] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. *ICML*, 2017. 5

[13] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 4

[14] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, pages 213–226, 2010. 1, 5

[15] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. *International Conference on Machine Learning (ICML)*, 2017. 4

[16] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. *Computer Vision and Pattern Recognition*, 2018. 4, 6

[17] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017. 4

[18] X. Zhu and A. B. Goldberg. *Introduction to semi-supervised learning*. Morgan & Claypool, 2009. 3, 4