Domain Conditional Predictors for Domain Adaptation

João Monteiro[1,*] **Xavier Gibert**[2], **Jianqiao Feng**[2], **Vincent Dumoulin**[2], **Dar-Shyang Lee**[2]
[1]INRS-EMT, Université du Québec
[2]Google
joao.monteiro@emt.inrs.ca, {xgibert,jianqiaofeng,vdumoulin,dsl}@google.com

Abstract

Learning guarantees often rely on assumptions of i.i.d. data, which will likely be violated in practice once predictors are deployed to perform *real-world* tasks. Domain adaptation approaches thus appeared as a useful framework yielding extra flexibility in that distinct train and test data distributions are supported, provided that other assumptions are satisfied such as *covariate shift*, which expects the conditional distributions over labels to be independent of the underlying data distribution. Several approaches were introduced in order to induce generalization across varying train and test data sources, and those often rely on the general idea of domain-invariance, in such a way that the data-generating distributions are to be disregarded by the prediction model. In this contribution, we tackle the problem of generalizing across data sources by taking the opposite direction. We consider a conditional modeling approach in which predictions, in addition of being dependent of the input data, use information relative to the underlying data-generating distribution. For instance, the model has an explicit mechanism to adapt to changing environments and/or new data sources. We argue that such an approach is more general than current domain adaptation methods since it does not require extra assumptions such as covariate shift and further yields simpler training algorithms that avoid a common source of training instabilities caused by minimax formulations, often employed in domain-invariant methods.

1 Introduction

Common generalization guarantees used to motivate supervised learning approaches under the empirical risk minimization framework rely on the assumption that data is collected independently from a fixed underlying distribution. Such an assumption, however, is not without shortcomings; for instance: (**i**)-i.i.d. requirements are *unverifiable* [1] in the sense that, given a data sample and no access to the distribution it was observed from, one cannot tell whether such a sample was collected independently, and (**ii**)-the i.i.d. assumption is *unpractical* since in several scenarios the conditions under which data is collected will likely change relative to when training samples were observed and, as such, generalization cannot be expected. Yet another practical limitation given by the lack of robustness against distribution shifts in common predictors is the fact that one cannot benefit from data sources that differ from those against which such predictors will be tested. In some situations, for example, data can be collected from inexpensive simulations, but generalization to real data cannot be expected depending on how far it is from available synthetic data.

Several approaches have been consequently introduced with the goal of relaxing requirements of i.i.d. data to some extent. For instance, *domain adaptation approaches* [2] assume the existence of two

[*]Work performed while João Monteiro was interning at Google.

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

distributions: the source distribution – which contains the bulk of the training data – and the target distribution – which corresponds to the test-time data distribution. While the domain adaptation setting enlarged the scope of the standard empirical risk minimization framework by enabling the use of predictors even when a distribution other than the one used for training is considered, a particular *target* is expected to be defined at training time, often with unlabeled examples, and nothing can be guaranteed for distributions other than that particular *target*, which renders such setting still unpractical since unseen variations in the data are possible during test. More general settings were introduced considering a larger set of supported target distributions while not requiring access to any target data during training [3]. However, such approaches, including domain adaptation techniques discussed so far, despite of relaxing the i.i.d. requirement, still require other assumptions to be met such as *covariate shift* (c.f. Sec. 2 for a definition).

As will be further discussed in Section 2, a common feature across a number of approaches enabling some kind of out-of-distribution generalization rely on some notion of invariance, be it either in the feature level [3, 4], in the sense that the underlying distributions or domains cannot be discriminated after mapped by a feature extractor, or in the predictor level [5, 6] in which case one expects distribution shifts will have little effect over predictions. In this contribution, the research question we pose to ourselves is as follows: *can one leverage contextual information to induce generalization to novel data sources?* We thus take an alternative approach relative to previous work and propose a framework where the opposite direction is considered in order to tackle the limitations discussed above, i.e. instead of filtering out the domain influence over predictions, we explore approaches where such information is leveraged as a source of context on which predictions can be conditioned. We refer to predictors resulting of such an approach as *domain conditional predictors*. We argue such a method includes the following advantages compared to settings seeking invariance:

1. Training strategies yielding domain conditional predictors do not rely on minimax formulations, often employed in domain invariant approaches where a domain discriminator is trained against a feature extractor. Such formulations are often observed to be source of training instabilities which do not appear in the setting considered herein.
2. The proposed setting does not rely on the covariate assumption since it considers multiple inputs, i.e. for a fixed input data instance, any prediction can be obtained through variations of the conditioning variable.
3. The proposed setting has a larger scope when compared to domain-invariant approaches in the sense that it can be used to perform inferences regarding the domains it observed during training.

The remainder of this paper is organized as follows: related literature is discussed in Section 2 along with background information and notation, while the proposed approach is presented in Section 3. The planned experimental setup is discussed in Section 4 and proof-of-concept results are reported in Section 5. Finally, conclusions as well as anticipated challenges are drawn in Section 6.

2 Background and Related Work

2.1 Domain adaptation guarantees and domain invariant approaches

Assume (x, y) represents instances from $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^D$ is the data space while $\mathcal{Y}$ is the space of labels, and $\mathcal{Y}$ will be a discrete set in the cases we consider. Furthermore, consider a deterministic labeling function, denoted $f : \mathcal{X} \mapsto \mathcal{Y}$, is such that $y = f(x)$. We will refer to domains as the pairs given by a marginal distribution over $\mathcal{X}$, denoted by $\mathcal{D}$, and a labeling function f. We further consider a family of candidate predictors $\mathcal{H}$ where $h \in \mathcal{H} : \mathcal{X} \mapsto \mathcal{Y}$. For a particular predictor h, we use the standard definition of risk R, which is its expected loss:

$$R_{\mathcal{D}}[h] = \mathbb{E}_{x \sim \mathcal{D}} \ell[h(x), f(x)], \tag{1}$$

where the loss $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ indicates how different $h(x)$ and $f(x)$ are (e.g. the 0-1 loss for $\mathcal{Y} = \{0, 1\}$).

In [2], Ben-David et al. showed that the following bound holds for the risk on a target domain $\mathcal{D}_T$ depending on the risk measured on the source domain $\mathcal{D}_S$:

$$R_{\mathcal{D}_T}[h] \leq R_{\mathcal{D}_S}[h] + d_{\mathcal{H}}[\mathcal{D}_S, \mathcal{D}_T] + \lambda, \tag{2}$$

and the following details are worth highlighting regarding such result: (**i**)-the term λ, as discussed in [7], accounts for differences between the labeling functions in the source and target domains, i.e. in the more general case the label y' of a particular data point x' depends on the underlying domain it was observed from. The *covariate shift* assumptions thus considers the more restrictive case where labeling functions match across domains, zeroing out λ and tightening the bound shown above. (**ii**)-the term $d_{\mathcal{H}}[\mathcal{D}_S, \mathcal{D}_T]$ corresponds to the discrepancy measure in terms of the $\mathcal{H}$-divergence (c.f. definition in [2]) as measured across the two considered domains.

The covariate shift assumption thus induces a setting where generalization can be expected if the considered domains lie close to each other in the $\mathcal{H}$-divergence sense. Such setting motivated the domain invariant approaches appearing across a number of recent domain adaptation methods [3, 4, 8, 9] where a feature extractor is forced to ignore domain-specific cues from the data and induce a low discrepancy across domains, enabling generalization. A similar direction was recently proposed to define invariant predictors instead of invariant representations in [5, 6]. In such setting, while data representations might still be domain-dependent, one seeks predictors that disregard domain factors in the sense that their predictions are not affected by changes in the underlying domains.

2.2 Conditional modeling

The problem of conditional modeling, i.e. that of defining models of a conditional distribution where some kind of contextual information is considered, appears across several areas. A conditioning approach was introduced in [10], for instance, where multi-modal visual reasoning tasks were considered. The conditioning layers, which authors referred to as FiLM, consist of an affine transformation where its parameters are themselves a function of the data. FiLM layers are defined by:

$$\text{FiLM}(x, z) = \gamma(z)F(x) + \beta(z), \tag{3}$$

where $F(x)$ represents features extracted from an input denoted x, while $\gamma(z)$ and $\beta(z)$ are arbitrary functions of some available conditioning information from the data and represented by z (e.g. x corresponded to images and z represented text in the original paper). γ and β were thus parameterized by neural networks and trained jointly with the main model, function of x.

An approach similar to FiLM was employed in [11] for the case of generative modeling. A conditioning layer consisting of adaptive instance normalization layers [12] was used to perform conditional generation of images providing control over the style of the generated data. Moreover, in [13] FiLM layers were applied in order to condition a predictor on entire classification tasks. The goal in that case was to enable adaptable predictors for few-shot classification of novel classes.

Other applications of such a framework include, for instance, conditional language modeling. In [14] for example, deterministic codes are given at train time indicating the style of a particular corpus. At test time, one can control the sampling process by giving each such code as an additional input and generate outputs corresponding to a particular style. In the case of applications to speech recognition, a common approach for acoustic modelling is to augment acoustic features with speaker-dependent representations so that the model can account for factors that are specific to the underlying speaker such as accent and speaking speed [15]. Representations such as i-vectors [16] are then combined with spectral features at every frame, and the combined representations are projected onto a mixed subspace, learned prior to training the acoustic model.

3 Domain conditional predictors

The setting we consider consists in designing models that parameterize a conditional categorical distribution which simultaneously relies on data as well as on domain information. We then assume training data comes from multiple source domains. Each example has a label that indicates which domain it belongs to. Additional notation is introduced to account for that, in which case we denote the set of domain labels by $\mathcal{Y}_{\mathcal{D}}$. We then consider two models given by:

1. $M_{domain} : \mathcal{X} \mapsto \Delta^{|\mathcal{Y}_\mathcal{D}|-1}$ maps a data instance x onto the $|\mathcal{Y}_\mathcal{D}| - 1$ probability simplex that defines the following categorical conditional distribution: $P(\mathcal{Y}_\mathcal{D}|x) = M_{domain}(x)$.
2. $M_{task} : \mathcal{X} \times \mathbb{R}^d \mapsto \Delta^{|\mathcal{Y}|-1}$, where an extra input represented by $z \in \mathbb{R}^d$ is a conditioning variable expected to carry domain information. M_{task} maps a data instance x and its corresponding z onto the $|\mathcal{Y}| - 1$ probability simplex, thus defining the following categorical conditional distribution: $P(\mathcal{Y}|x, z) = M_{task}(x, z)$.

We implement both M_{domain} and M_{task} using neural networks, and training is carried out with simultaneous maximum likelihood estimation over $P(\mathcal{Y}_\mathcal{D}|x)$ and $P(\mathcal{Y}|x, z)$ so that the training objective $\mathcal{L} = (1-\lambda)\mathcal{L}_{task} + \lambda\mathcal{L}_{domain}$ is defined by the sum of the multi-class cross-entropy losses defined over the set of task and domains labels, respectively, and $\lambda \in [0, 1]$ is a hyperparameter that controls the importance of each loss term during training. Moreover, z is given by the output of some inner layer of M_{domain}, since z is expected to contain domain-dependent information. A training procedure is depicted in Algorithm 1.

In order for M_{task} to be able to use the domain conditioning information made available through z, we make use of FiLM layers represented by:

$$FiLM^k(x^{k-1}) = (W_1^k z + b_1^k)x^{k-1} + (W_2^k z + b_2^k), \tag{4}$$

where k indicates a particular layer within M_{task}, and x^{k-1} corresponds to the output of the last layer. W_1^k, b_1^k, W_2^k, and b_2^k correspond to the conditioning parameters trained along with the complete model.

Algorithm 1 Training procedure.

$M_{task}, M_{domain} = InitializeModels()$
repeat
  $x, y, y_\mathcal{D} = SampleMinibatch()$
  $y'_\mathcal{D}, z = M_{domain}(x)$
  $y' = M_{task}(x, z)$
  $\mathcal{L} = \mathcal{L}_{task}(y', y) + \mathcal{L}_{domain}(y'_\mathcal{D}, y_\mathcal{D})$
  $M_{task}, M_{domain} = UpdateRule(M_{task}, M_{domain}, \mathcal{L})$
until Maximum number of iterations reached
return M_{task}, M_{domain}

At test time, two distinct classifiers can be defined such as the task predictor given by:

$$\underset{i \in [|\mathcal{Y}|]}{\arg\max}\, M_{task}(x, z)_i, \tag{5}$$

or the domain predictor defined by:

$$\underset{j \in [|\mathcal{Y}_\mathcal{D}|]}{\arg\max}\, M_{domain}(x)_j, \tag{6}$$

thus enabling extra prediction mechanisms compared to methods that remove domain information.

4 Planned evaluation and results

In this section, we list the datasets we consider for the evaluation along with baseline methods, ablations, and the considered variations of conditioning approaches.

4.1 Datasets

Evaluations are to be performed on a subset of the following well-known domain generalization benchmarks:

- PACS [17]: It consists of 224x224 RGB images distributed into 7 classes and originated from four different domains: Photo (P), Art painting (A), Cartoon (C), and Sketch (S).

- VLCS [18]: VLCS is composed by 5 overlapping classes of objects obtained from the following datasets: VOC2007 [19], LabelMe [20], Caltech-101 [21], and SUN [22].
- OfficeHome [23]: This dataset consists of images from the following domains: artistic images, clip art, product images and natural images. Each domain contains images of 65 classes.
- DomainNet [24]: DomainNet contains examples of 224x224 RGB images corresponding to 345 classes of objects across 6 distinct domains.

Evaluation metric Across all mentioned datasets, we follow the *leave-one-domain-out* evaluation protocol such that data from $|\mathcal{Y}_{\mathcal{D}}| - 1$ out of the $|\mathcal{Y}_{\mathcal{D}}|$ available domains are used for training, while evaluation is carried out on the data from the left out domain. This procedure is repeated for all available domains, and once each domain is left out, the *average top-1 accuracy* is the evaluation metric under consideration. Moreover, in order to provide comparisons with significance, performance is to be reported in terms of confidence intervals obtained from independent models trained with different random seeds.

4.2 Baselines

The main aspect under investigation within this work is whether one can leverage domain information rather than removing or disregarding it such as in typical settings. Our main baselines then correspond to two settings where some kind of domain invariance is enforced: *domain-adversarial approaches* and *invariant predictors*. We specifically consider DANN [4] and G2DM [3] corresponding to the former, while IRM [5] and Rex [6] are considered for the latter. Additionally, two further baselines are evaluated: an *unconditional model* in the form of a standard classifier that disregards domain labels as well as a model replacing M_{domain} by a standard embedding layer[2].

4.3 Ablations

In order to investigate different sources of potential improvement, we will drop the domain classification term of the loss ($\mathcal{L}_{domain}$), obtaining a predictor with the same capacity as the proposed model while having no explicit mechanism to conditional modeling. A drop in performance should serve as evidence that the conditioning approach yields improvement. Moreover, similarly to ablations performed in the original FiLM paper [10], we plan on evaluating cases where scaling and offset parameters (i.e. γ and β as indicated in Eq. 3) are all set to 1 or 0, indicating which parameter set is more important for the conditioning strategy to be effective.

4.4 Further evaluation details

As discussed in [3], the chosen validation data source used to implement model selection and stopping criteria significantly affects the performance of domain generalization approaches. We remark that, in this work, the access model to left out domains is such that no access to target data is allowed for model selection or hyperparameter tuning. We thus only use in-domain validation data. However, we further consider the so-called "privileged" variants of both our models and baselines in the sense that they are given access target data. In doing so, we can get a sense of the gap in performance observed across the settings.

5 Proof-of-concept evaluation

We used MNIST to perform validation experiments and considered different domains simulated through transformations applied on training data. Considered such transformations are as follows: (**i**)-horizontal flip of digits, (**ii**)-switching colors between digits and background, (**iii**)-blurring, and (**iv**)-rotation. Examples from each transformation are shown in Figures 1-4. Test data corresponds to the standard test examples without any transformation. In-domain performance is further assessed by applying the same transformations on the test data. Two baselines are considered in this set of experiments consisting of an unconditional model as well as a domain adversarial approach similar

[2]For this case, evaluation is performed in-domain, i.e. fresh data from the same domains observed at training time are used for testing.

Table 1: Classification performance in terms of top-1 accuracy (%). In-domain performance is measured using distorted MNIST test images while out-of-domain results correspond to evaluation on the standard test set of MNIST. The ablation with a leaned embedding layer can only be used for in-domain predictions. For the in-domain evaluation, we loop over the test data 10 times to reduce the evaluation variance since each test example will be transformed differently each time.

Model	**In-domain test accuracy (%)**	**Out-of-domain test accuracy (%)**
Unconditional baseline	94.97	92.51
Adversarial baseline	89.19	88.49
Ablation: switching M_{domain} for an embedding layer	92.56	–
Ablation: dropping $\mathcal{L}_{domain}$	96.20	92.94
Conditional predictor (*Ours*)	96.00	93.66

Table 2: Domain classification top-1 accuracy (%) measured for predictions of the underlying domain when the same transformations applied to train data are applied to test examples. In the ablation case, a linear classifier is trained on top of z with the rest of the model frozen.

Model	**Test accuracy (%)**
Adversarial baseline	84.22
Ablation: dropping $\mathcal{L}_{domain}$	96.01
Conditional predictor (*Ours*)	99.90

to DANN. For the case of the adversarial baselines, training is carried out so that alternate updates are performed to jointly train a task classifier and a domain classifier. The task classifier trains to minimize its classification loss and further maximizes the entropy of the domain classifier aiming to enforce domain invariance in the representations extracted after its convolutional layer. The domain classifier trains to correctly classify domains. Two ablations are also considered. The first one consists of a conditional model with learned domain-level context variables used for conditioning, in which case the conditioning model is replaced by an embedding layer. Such model can only be evaluated in in-domain test data. Additionally, we consider the ablation described above so that the domain classification term of the loss is dropped.

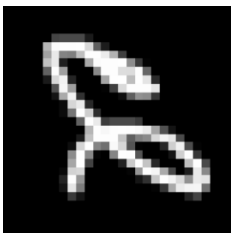

Figure 1: Horiz. flip.

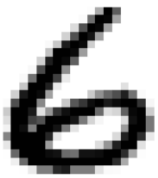

Figure 2: Switch colors.

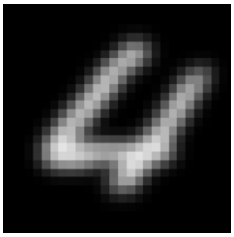

Figure 3: Blurring.

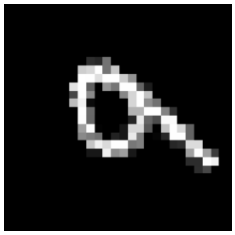

Figure 4: Rotation.

Results, as reported in Table 1, indicate that the conditioning approach boosts performance with respect to standard classifiers that disregard domain information as well as domain invariant approaches. In fact, the conditional predictors presented the highest out-of-domain accuracy amongst all evaluated methods. Surprisingly, in the ablation case where we drop $\mathcal{L}_{domain}$, domains can still be inferred from z with high accuracy (c.f. Table 2) which indicates the domain conditioning strategy enabled by the proposed architecture is exploited even if not enforced by an explicit training objective when multiple domains are present in the training sample.

6 Conclusion

A research proposal was introduced in this document with the goal of defining domain conditional predictors, i.e. models that adapt to the underlying environment corresponding to the data being accounted for. We argue such a setting can alleviate well-known issues limiting the adoption of state-of-the-art classifiers in real-world scenarios where data sources are prone to variations over time, thus violating i.i.d. assumptions. A detailed training approach is provided along with an evaluation plan which we intend to employ in order to assess the effectiveness of the proposed framework.

References

[1] J. Langford, "Tutorial on practical prediction theory for classification," *Journal of machine learning research*, vol. 6, no. Mar, pp. 273–306, 2005.

[2] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Advances in neural information processing systems*, 2007, pp. 137–144.

[3] I. Albuquerque, J. Monteiro, M. Darvishi, T. H. Falk, and I. Mitliagkas, "Generalizing to unseen domains via distribution matching," *arXiv preprint arXiv:1911.00804*, 2019.

[4] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[5] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.

[6] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, R. L. Priol, and A. Courville, "Out-of-distribution generalization via risk extrapolation (rex)," *arXiv preprint arXiv:2003.00688*, 2020.

[7] H. Zhao, R. T. Des Combes, K. Zhang, and G. Gordon, "On learning invariant representations for domain adaptation," in *International Conference on Machine Learning*, 2019, pp. 7523–7532.

[8] G. Bhattacharya, J. Monteiro, J. Alam, and P. Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6226–6230.

[9] P. Bashivan, B. Richards, and I. Rish, "Adversarial feature desensitization," *arXiv preprint arXiv:2006.04621*, 2020.

[10] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," *arXiv preprint arXiv:1709.07871*, 2017.

[11] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[12] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[13] H. Prol, V. Dumoulin, and L. Herranz, "Cross-modulation networks for few-shot learning," *arXiv preprint arXiv:1812.00273*, 2018.

[14] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "Ctrl: A conditional transformer language model for controllable generation," *arXiv preprint arXiv:1909.05858*, 2019.

[15] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[16] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[17] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5542–5550.

[18] C. Fang, Y. Xu, and D. N. Rockmore, "Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1657–1664.

[19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[20] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[21] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[22] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 129–136.

[23] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[24] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1406–1415.