Provable domain adaptation using privileged information

Adam Breitholtz [*1] **Anton Matsson** [*1] **Fredrik D. Johansson** [1]

Abstract

Successful unsupervised domain adaptation is guaranteed only under strong assumptions such as covariate shift and overlap between input domains. The latter is often violated in high-dimensional applications such as image classification which, despite this challenge, continues to serve as inspiration and benchmark for algorithm development. In this work, we show that access to side information about examples from the source and target domains can help relax sufficient assumptions on input variables and increase sample efficiency at the cost of collecting a richer variable set. We call this unsupervised domain adaptation by learning using privileged information (DALUPI). Tailored for this task, we propose algorithms for both multi-class and multi-label classification tasks. In our experiments we demonstrate that incorporating privileged information in learning can reduce errors in domain transfer and increase sample efficiency compared to classical learning.

1. Introduction

Deployment of machine learning (ML) systems relies on generalization from training samples to new instances in a target domain. When these instances differ in distribution from the source of training data, performance tends to degrade and guarantees are often weak. For example, a supervised ML model trained to identify medical conditions in X-ray images from one hospital may work poorly in another if the two hospitals have different equipment or examination protocols (Zech et al., 2018). If no labeled examples are available from the target domain, the so-called *unsupervised domain adaptation* (UDA) problem (Ben-David et al., 2006), strong assumptions are needed for success.

A common assumption in UDA is that the object of the learning task is identical in source and target domains but that input distributions differ (Shimodaira, 2000). This "covariate shift" assumption is plausible in our X-ray example above; doctors are likely to give the same diagnosis based on X-rays of the same patient from similar but different equipment. Additionally, however, guarantees for consistent domain transfer require either distributional overlap between inputs from source and target domains or known parametric forms of the labeling function (Ben-David & Urner, 2012; Wu et al., 2019; Johansson et al., 2019). Without these, adaptation cannot be verified by statistical means.

Incorporating side information in training has been proposed to improve generalization without domain shift. Through learning using privileged information (PI)(Vapnik & Vashist, 2009; Lopez-Paz et al., 2016), algorithms that access auxiliary variables during training that are unavailable in deployment have been proven to learn from fewer examples compared to algorithms trained without these variables (Karlsson et al., 2022). While PI has been used in domain adaptation, see e.g., Sarafianos et al. (2017); Vu et al. (2019), the literature has yet to characterize the benefits of this practice.

Contributions. In this work, we define *unsupervised domain adaptation by learning using privileged information* (DALUPI), where unlabeled examples with privileged information (PI) are available from the target domain. For this problem, we give conditions under which it is possible to identify a model which predicts optimally in the target domain, without assuming statistical overlap between source and target input domains. We instantiate this problem setting in multi-class and multi-label image classification and propose practical algorithms for these tasks. On image classification tasks with boxes indicating regions of interest as PI we compare our algorithms to baselines for supervised learning and unsupervised domain adaptation. We find that they perform favorably to the alternatives, particularly when spurious correlations are strong.

2. UDA and assumptions

In unsupervised domain adaptation (UDA), our goal is to learn a hypothesis h to predict outcomes (or labels) $Y \in \mathcal{Y}$ for input covariates $X \in \mathcal{X}$, drawn from a target domain with density $\mathcal{T}(X, Y)$. During training, we have access to

[*] Equal contribution [1] Department of Computer Science & Engineering, Chalmers University of Technology, Göteborg, Sweden. Correspondence to: Adam Breitholtz <adambre@chalmers.se>.

Published at the ICML 2023 Workshop on Spurious Correlations, Invariance, and Stability. Honolulu, Hawaii, USA. Copyright 2023 by the author(s).

labeled samples (x, y) only from a source domain $\mathcal{S}(X, Y)$ and unlabeled samples $\tilde{x}$ from $\mathcal{T}(X)$.

We aim to minimize the expected target-domain prediction error (risk) $R_{\mathcal{T}}$ of a hypothesis $h \in \mathcal{H}$ from a hypothesis set $\mathcal{H}$, with respect to a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$,

$$\min_{h \in \mathcal{H}} R_{\mathcal{T}}(h), \quad R_{\mathcal{T}}(h) := \mathbb{E}_{X,Y \sim \mathcal{T}}[L(h(X), Y)] \,. \qquad (1)$$

A solution to the UDA problem returns a minimizer of (1) without ever observing labeled samples from $\mathcal{T}$. However, if $\mathcal{S}$ and $\mathcal{T}$ are allowed to differ arbitrarily, finding such a solution cannot be guaranteed (Ben-David & Urner, 2012). Under the assumption of *covariate shift* (Shimodaira, 2000), the labeling function is the same on both domains, but the covariate distributions differ.

Assumption 2.1 (Covariate shift)**.** For domains $\mathcal{S}, \mathcal{T}$ on $\mathcal{X} \times \mathcal{Y}$, *covariate shift* holds w.r.t. X if

$$\exists x : \mathcal{T}(x) \neq \mathcal{S}(x) \;\; \text{and} \;\; \forall x : \mathcal{T}(Y \mid x) = \mathcal{S}(Y \mid x) \,.$$

To guarantee consistent learning without further assumptions, these distributions cannot be *too* different; the source domain input $\mathcal{S}(x)$ must sufficiently *overlap* the target input domain $\mathcal{T}(x)$.

Assumption 2.2 (Domain overlap)**.** A domain $\mathcal{S}$ overlaps another domain $\mathcal{T}$ w.r.t. Z on $\mathcal{Z}$ if

$$\forall z \in \mathcal{Z} : \mathcal{T}(Z = z) > 0 \implies \mathcal{S}(Z = z) > 0 \,.$$

In high-dimensional problems, overlap is often violated partly due to irrelevant information. One example is background pixels in image object detection (Beery et al., 2018) which may be distinct to domains (e.g., indoor/outdoor) but only have spurious association with the label Y. Still, the image may *contain* information W which is both sufficient for prediction and supported in both domains. Which information in X satisfies these conditions is not self-evident, but can be supplied during training as added supervision. Learning from these data is an example of *learning using privileged information* (LUPI) (Vapnik & Vashist, 2009).

3. Unsupervised domain adaptation using privileged information

We define domain adaptation by learning using privileged information (DALUPI) as follows. During training, learners observe samples of covariates X, labels Y and privileged information $W \in \mathcal{W}$ from $\mathcal{S}$ in a dataset $D_{\mathcal{S}} = \{(x_i, w_i, y_i)\}_{i=1}^m$, as well as samples of covariates and privileged information from $\mathcal{T}$, $D_{\mathcal{T}} = \{(\tilde{x}_i, \tilde{w}_i)\}_{i=1}^n$. *At test time, trained models only observe covariates $\tilde{x} \sim \mathcal{T}(X)$ and our learning goal remains to minimize the target risk* (1). Access to privileged information from $\mathcal{T}$, but not labels, can be justified by noting that it is often easier to annotate observations with privileged information W than with labels Y. For example, a non-expert may be able to reliably recognize the outline (W) of an animal in an image but not its species (Y). Figure 1 illustrates several applications with the DALUPI structure. To identify $R_{\mathcal{T}}$ (1) without overlap in X, we make the assumption that W is sufficient for Y.

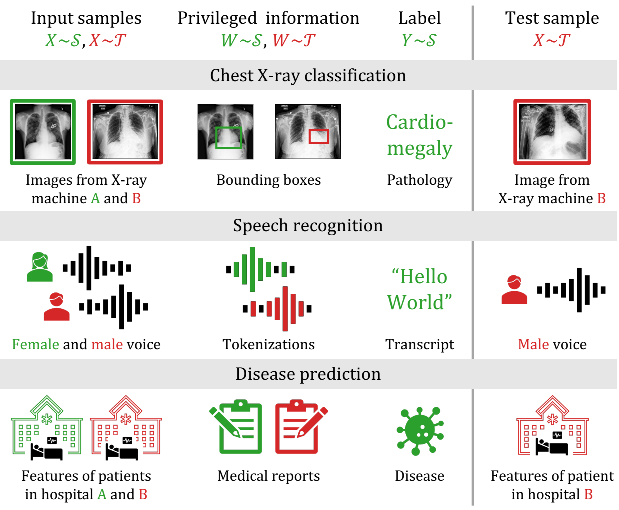


Figure 1: Examples of domain adaptation with privileged information. Top: Chest X-ray images X together with pathology masks W and diagnosis label Y. Middle: Tokenizations W accompany recorded speech X during learning of a speech recognition system. Bottom: Clinical notes W used in learning to predict disease progression from patient features X.

Assumption 3.1 (Sufficiency of privileged information)**.** Privileged information W is sufficient for the outcome Y given covariates X if $Y \perp X \mid W$ in both $\mathcal{S}$ and $\mathcal{T}$.

Applying Assumptions 2.1–3.1 to privileged information W allows us to identify the target risk even for models that do not make use of W as input. See Appendix B for the proof.

Proposition 3.2. *Let Assumptions 2.1 and 2.2 be satisfied w.r.t. W (not necessarily w.r.t. X) and let Assumption 3.1 hold as stated. Then, the target risk $R_{\mathcal{T}}$ is identified for hypotheses $h : \mathcal{X} \to \mathcal{Y}$,*

$$R_{\mathcal{T}}(h) = \sum_x \mathcal{T}(x) \sum_w \mathcal{T}(w \mid x) \sum_y \mathcal{S}(y \mid w) L(h(x), y) \,.$$

and, for L the squared loss, a minimizer of $R_{\mathcal{T}}$ is $h^*_{\mathcal{T}}(x) = \sum_w \mathcal{T}(w \mid x) \mathbb{E}_{\mathcal{S}}[Y \mid w]$.

A two-stage learning algorithm. In light of Proposition 3.2, a natural learning strategy is to model privileged information as a function of the input, $\mathcal{T}(W \mid x)$, and the outcome as a function of privileged information,

$\hat{g}(w) \approx \mathbb{E}_{\mathcal{S}}[Y \mid w]$, and combining these. In the case where W is a deterministic function of X, $\mathcal{T}(W \mid x)$ is a map $f : \mathcal{X} \to \mathcal{W}$, which may be estimated as a regression $\hat{f}$ and combined with the outcome regression to form $\hat{h} = \hat{g}(\hat{f}(X))$. We may find such functions $\hat{f}$, $\hat{g}$ by separately minimizing the empirical risks

$$\hat{R}^W_{\mathcal{T}}(f) = \sum_{i=1}^{n} \frac{L(f(\tilde{x}_i), \tilde{w}_i)}{n}, \hat{R}^Y_{\mathcal{S}}(g) = \sum_{i=1}^{m} \frac{L(g(w_i), y_i)}{m} \tag{2}$$

Under some additional assumptions we prove a generalization bound based on the setup in (2), see Appendix C.

Sufficiency and overlap. Assumption 3.1 is satisfied when X provides no more information about Y in the presence of W. For example, if W is the subset of X-ray pixels corresponding to an area indicating a medical problem, the other pixels in X may be unnecessary to predict Y. When W retains more information, sufficiency becomes more plausible but domain overlap in W is reduced. The sufficiency assumption is used to replace $\mathcal{T}(y \mid x)$ with $\mathcal{T}(y \mid w)$ in Proposition 3.2. If sufficiency is violated but it is plausible that the degree of insufficiency is comparable across domains, we can still obtain a bound on the target risk which may be estimated from observed quantities. We give such a result in Appendix D. It is instructive to compare Assumptions 2.1–3.1 w.r.t. W to the desired properties of the representation in Construction 4.1 of Wu et al. (2019).

4. Experiments

We use image classification as proof of concept for DALUPI, solving both multi-class and multi-label tasks where privileged information W highlights regions of interest in the images X, related to the labels Y, in the form of pixels contained by bounding boxes with coordinates T. We study identifiability and robustness to spurious correlation in a digit classification task(Section 4.1). We also use a multi-label variant of DALUPI based on Faster R-CNN for classifying entities from the MS-COCO dataset(Section 4.2).

All baselines are based on ResNet architectures matching those used for DALUPI, adapted to the multi-class, or multi-label, settings. First, we use standard supervised classifiers, SL-S and SL-T, trained on labeled examples, (x, y) or $(\tilde{x}, \tilde{y})$, from the source and target domain, respectively. The latter serves as an oracle comparison since labels are unavailable from the target in UDA. Second, we include UDA techniques that regularize representations to encourage adaptation using labeled data from the source domain and unlabeled data from the target domain: domain adversarial neural networks (DANN) (Ganin et al., 2016) (Section 4.1–4.2) and the margin disparity discrepancy (MDD) (Zhang et al., 2019) (Section 4.2). Finally, we include a model (LUPI) in the entity classification task, trained with privileged information but without any target data. We report accuracy and area under the ROC curve (AUC) with 95 % confidence intervals computed by bootstrapping the results over several seeds. All details of the experimental setup, the model architectures and hyperparameter choices are described in Appendix A.

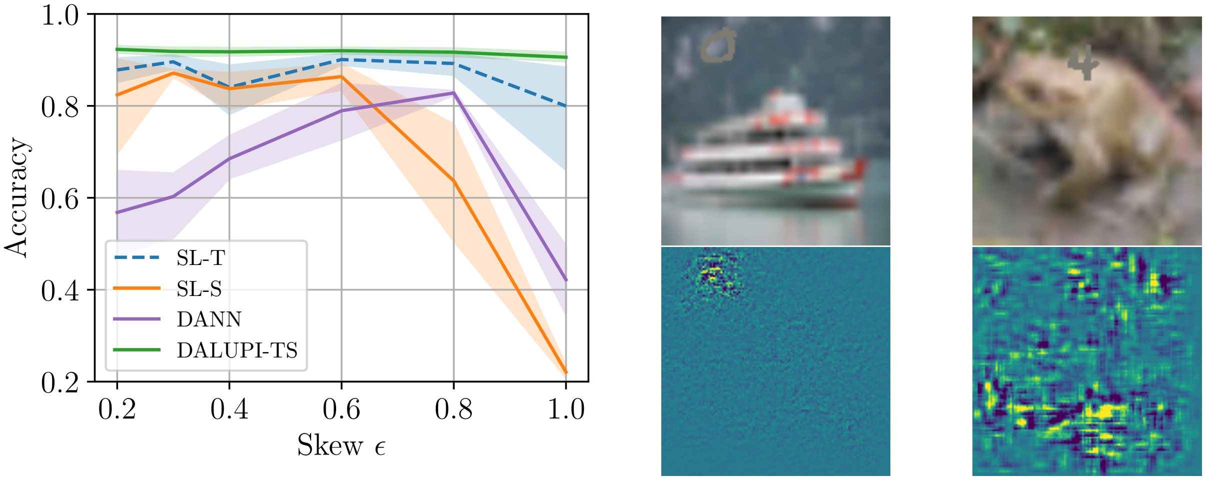
4.1. Multi-class digit classification

In this experiment, we make use of a slightly modified version of the two-stage estimator from Section 3. We use two convolutional neural networks to approximate the setting. See Appendix A.1 for architectures and other details.

We construct a synthetic dataset, based on the assumptions of Proposition 3.2, to verify that there are problems where DALUPI consistently succeeds in domain transfer but UDA does not. We start with images from CIFAR-10 (Krizhevsky, 2009) which have been upscaled to 128×128. We insert a random 28×28 digit image, with a label in the range 0–4, from the MNIST dataset (Lecun, 1998) into a random location of the CIFAR-10 image, forming the input image X (see Figure 2b–2c for examples). The label $Y \in \{0, \ldots, 4\}$ is determined by the MNIST digit. We store the bounding box around the inserted digit image and use the pixels contained within it as privileged information W.

To understand how successful transfer depends on domain overlap and access to sufficient privileged information, we include a *domain skew parameter* $\epsilon \in [\frac{1}{c}, 1]$, where $c = 5$ is the number of digit classes, which determines the (spurious) correlation between digits and backgrounds. For a source image i with digit label $Y_i \in \{0, \ldots, 4\}$, we select a random CIFAR-10 image with class $B_i \in \{0, \ldots, 4\}$ with probability $P(B_i = b \mid Y_i = y) = \{\epsilon, \text{ if } b = y; \; (1 - \epsilon)/(c - 1), \text{otherwise}\}$. For further details on the construction see appendix A.3. For target images, digits and backgrounds are matched uniformly at random. Note that the choice of $\epsilon = \frac{1}{c}$ yields a uniform distribution and $\epsilon = 1$ being equivalent to the background carrying as much signal as the privileged information.

In Figure 2a, we can observe as the skew ϵ increases, the performance of SL-S decreases substantially on the target domain. It is clear that the spurious association with the background harms generalization performance, which has been observed in previous work where background artifacts in X-ray images confused the resulting classifier. See the saliency maps in Figure 2b–2c for an illustration of this behavior. We also observe that DANN does not seem to be robust to the increase in correlation between the label and the background. In contrast, DALUPI is unaffected by the skew as the subset of pixels only carries some of the background information with it, while containing sufficient information to make good predictions. Interestingly, DALUPI also seems to be as good or slightly better than the



(a) Target test accuracy.

(b) SL-S, $\epsilon = 0.2$.

(c) SL-S, $\epsilon = 1.0$.

Figure 2: Digit classification. Left: Performance on source (a) and target (b) domains as a function of association ϵ between background and label in $\mathcal{S}$. As the skew increases, the target-domain performance of the non-privileged models deteriorates. Right: Example images (top) and saliency maps (bottom) from SL-S trained with source skew $\epsilon = 0.2$ (c) and $\epsilon = 1$ (d).

oracle SL-T in this setting. This may be due to improved sample efficiency from using PI.

4.2. Multi-label entity classification with Faster R-CNN

In our second experiment, we propose a more practically useful algorithm for the multi-label setting. For this purpose, we adapt the Faster R-CNN architecture (Ren et al., 2016) which uses a region proposal network to generate regions that are fed to a detection network for classification and refined bounding box regression. Since bounding boxes are considered privileged information in our setup, they serve as supervision for this regression when available. We describe the learning objective and architecture in detail in Appendix A.2. Unlike the two-stage multi-class model, we train the Faster R-CNN model end-to-end, minimizing both losses at once. We also train this model in a LUPI setting, where no information from the target domain is used.

We study the end-to-end model on a multi-label entity classification task based on images from MS-COCO (Lin et al., 2014). We define the source and target domains as indoor and outdoor images, respectively, and consider four entity classes for the label Y: person, cat, dog, and bird. We extract indoor images by filtering out images from the super categories "indoor" and "appliance" that also contain at least one of the entity classes. Outdoor images are extracted in the same way using the super categories "vehicle" and "outdoor". Images that match both domains are removed, as are any gray-scale images. We also include negative examples, i.e., images with no entities present, in both domains.

Table 1: Entity classification. Test AUC in the target domain. UDA models have access to all unlabeled target samples, LUPI to all PI (source), and DALUPI to all PI (source and target). Best feasible model in boldface.

	AUC
SL-T	70.6 (69.7, 71.3)
SL-S	64.5 (63.5, 65.3)
DANN	61.6 (60.5, 62.9)
MDD	52.8 (51.6, 53.9)
LUPI	65.9 (64.4, 67.3)
DALUPI	**71.3 (70.2, 72.4)**

Table 1 shows the models' target domain AUC, averaged over the four entity classes, when the UDA models have access to all unlabeled target samples, LUPI to all PI from the source domain, and DALUPI to all PI from both domains. Clearly, DALUPI yields a significant gain in adaptation. Varying the amount of PI (see Appendix A.4.1) shows that the DALUPI model performs worse than SL-S when no PI is available from source or target, but that it approaches SL-T (oracle) when all samples have PI.

5. Related work

Learning using privileged information was first introduced by Vapnik & Vashist (2009) for support vector machines (SVMs), and was later extended to empirical risk minimization (Pechyony & Vapnik, 2010). Methods using PI, which is sometimes called hidden information or side information, has since been applied in many diverse settings such as healthcare (Shaikh et al., 2020), finance (Silva et al., 2010), clustering (Feyereisl & Aickelin, 2012) and image recognition (Vu et al., 2019; Hoffman et al., 2016). Related concepts include knowledge distillation (Hinton et al., 2015; Lopez-Paz et al., 2016), where a teacher model trained on additional variables adds supervision to a student model, and weak supervision (Robinson et al., 2020) where so-called weak labels are used to learn embeddings, subsequently used for the task of interest. The use of PI for deep image classification has been investigated by Chen et al. (2017) but this work only covers regular supervised learning. Finally, Sharmanska et al. (2014) used regions of interest in images as privileged information to improve the accuracy of image classifiers, but did not consider domain shift.

Domain adaptation using PI has been considered before with SVMs (Li et al., 2022; Sarafianos et al., 2017). Vu et al. (2019) used scene depth as PI in semantic segmentation using deep neural networks. However, they only used PI from the source domain and they did not provide any theoretical analysis. Motiian (2019) investigated PI and domain adaptation using the information bottleneck method for visual recognition. However, their setting differs from ours in that each observation comprises source-domain and target-domain features, a label and PI.

6. Conclusion

We have presented DALUPI: unsupervised Domain Adaptation by Learning Using Privileged Information. The framework builds on an alternative set of assumptions that can more plausibly be satisfied in high-dimensional domain adaptation, at the cost of collecting a larger variable set only during training. Our analysis of this setting inspired practical algorithms for multi-class and multi-label image classification, and our experiments demonstrate tasks where these are successful while regular adaptation methods fail. We also observe empirically that methods using privileged information can be resistant to spurious correlations in data.

To avoid assuming that domain overlap is satisfied with respect to input covariates, we require that the privileged information (PI) is sufficient to determine the label—that once this information is known, knowing the input variables don't improve prediction. This can be limiting in problems where sufficiency is difficult to verify or reason about. However, in our motivating example of image classification, a domain expert could deliberately choose PI so that sufficiency is reasonably satisfied. In future work, our proposed framework might be applied to a more diverse set of tasks, with different modalities to investigate if the findings here can be replicated. Using PI may be viewed as "building in" domain knowledge in the structure of the adaptation problem and we see this as a promising approach for further research.

References

Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018.

Ben-David, S. and Urner, R. On the hardness of domain adaptation and the utility of unlabeled target samples. In *International Conference on Algorithmic Learning Theory*, pp. 139–153. Springer, 2012.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

Chen, Y., Jin, X., Feng, J., and Yan, S. Training group orthogonal neural networks with privileged information. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 1532–1538, 2017.

Cortes, C., Mansour, Y., and Mohri, M. Learning bounds for importance weighting. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

de Mathelin, A., Deheeger, F., Richard, G., Mougeot, M., and Vayatis, N. ADAPT: Awesome Domain Adaptation Python Toolbox. *arXiv preprint arXiv:2107.03049*, 2021.

Feyereisl, J. and Aickelin, U. Privileged information for data clustering. *Information Sciences*, 194:4–23, 2012.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-Adversarial Training of Neural Networks. *arXiv:1505.07818 [cs, stat]*, May 2016. arXiv: 1505.07818.

Girshick, R. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hoffman, J., Gupta, S., and Darrell, T. Learning with Side Information through Modality Hallucination. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 826–834. IEEE, 2016.

Johansson, F. D., Sontag, D., and Ranganath, R. Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 527–536. PMLR, 2019.

Karlsson, R., Willbo, M., Hussain, Z., Krishnan, R. G., Sontag, D. A., and Johansson, F. D. Using time-series privileged information for provably efficient learning of prediction models. In *Proc. of The 25th International Conference on Artificial Intelligence and Statistics*, 2022.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Lecun, Y. Gradient-Based Learning Applied to Document Recognition. *proceedings of the IEEE*, 86(11):47, 1998.

Li, Y., Sun, H., and Yan, W. Domain adaptive twin support vector machine learning using privileged information. *Neurocomputing*, 469:13–27, 2022.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., and Zitnick, L. Microsoft COCO: Common Objects in Context. In *ECCV*. European Conference on Computer Vision, September 2014.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Lopez-Paz, D., Bottou, L., Schölkopf, B., and Vapnik, V. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR 2016)*, 2016.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. MIT Press, second edition, 2018.

Motiian, S. *Domain Adaptation and Privileged Information for Visual Recognition*. PhD thesis, West Virginia University, 2019. Graduate Theses, Dissertations, and Problem Reports. 6271.

Pechyony, D. and Vapnik, V. On the theory of learnining with privileged information. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

Ren, S., He, K., Girshick, R., and Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2016.

Robinson, J., Jegelka, S., and Sra, S. Strength from Weakness: Fast Learning Using Weak Supervision. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 8127–8136. PMLR, November 2020.

Sarafianos, N., Vrigkas, M., and Kakadiaris, I. A. Adaptive SVM+: Learning with privileged information for domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

Shaikh, T. A., Ali, R., and Beg, M. M. S. Transfer learning privileged information fuels CAD diagnosis of breast cancer. *Machine Vision and Applications*, 31(1):9, February 2020.

Sharmanska, V., Quadrianto, N., and Lampert, C. H. Learning to Transfer Privileged Information. *arXiv:1410.0389 [cs, stat]*, October 2014. arXiv: 1410.0389.

Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Silva, C., Vieira, A., Gaspar-Cunha, A., and Carvalho das Neves, J. Financial distress model prediction using SVM+. In *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–7, July 2010.

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density Ratio Estimation in Machine Learning*. Cambridge books online. Cambridge University Press, 2012.

Tietz, M., Fan, T. J., Nouri, D., Bossan, B., and skorch Developers. *skorch: A scikit-learn compatible neural network library that wraps PyTorch*, July 2017.

Vapnik, V. and Vashist, A. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22 (5):544–557, July 2009.

Vu, T.-H., Jain, H., Bucher, M., Cord, M., and Pérez, P. P. Dada: Depth-aware domain adaptation in semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7363–7372, 2019.

Wu, Y., Winston, E., Kaushik, D., and Lipton, Z. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International conference on machine learning*, pp. 6872–6881. PMLR, 2019.

Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., and Oermann, E. K. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683, 2018.

Zhang, Y., Liu, T., Long, M., and Jordan, M. Bridging theory and algorithm for domain adaptation. In *International conference on machine learning*, pp. 7404–7413. PMLR, 2019.

A. Experimental details

In this section, we give further details of the experiments. All code is written in Python and we mainly use PyTorch in combination with skorch (Tietz et al., 2017) for our implementations of the networks. For Faster R-CNN, we adapt the implementation provided by torchvision through the function `fasterrcnn_resnet50_fpn`. For DANN and MDD, we use the ADAPT TensorFlow implementation (de Mathelin et al., 2021) with a ResNet-50-based encoder for the entity classification and our own implementation with a ResNet-18 encoder for the digit classification. We set the trade-off parameter λ, which controls the amount of domain adaption regularization, to a fixed value 0.1. For MDD, we fix the margin parameter γ to 3.

The source and target baselines are based on the ResNet-50 architecture for the entity classification and ResNet-18 for the digit classification. We use the Adam optimizer in all experiments.

For each experiment setting (task, label skew, amount of privileged information, etc.), we train 10 models from each relevant class using hyperparameters randomly selected from given ranges. For DANN and MDD, the trade-off parameter is set to 0.1; for MDD, the margin parameter is set to 3. All models are evaluated on a held-out validation set from the source domain and the best-performing model in each class is then evaluated on held-out test sets from both domains. For SL-T, we use a held-out validation set from the target domain. We repeat this procedure over up to 10 seeds controlling the data splits and the random number generation.

For all models except LUPI and DALUPI, the classifier network following the encoder is a simple MLP with two settings; either it is a single linear layer from inputs to outputs or a three layer network with ReLU activations between the layers. If nothing else is stated, this choice is treated as a hyperparameter. The nonlinear case has the following structure where n is the number of input features:

- fully connected layer with n neurons
- ReLU activation layer
- fully connected layer with $n/2$ neurons
- ReLU activation layer
- fully connected layer with $n/4$ neurons.

A.1. DALUPI with two-stage multi-class classifier

Here, we describe in more detail how we construct our two stage classifier for the multi-class task. The privileged information W highlights regions of interest in the images X, related to the labels Y, in the form of pixels contained by bounding boxes with coordinates T. Each image x_i has a single label $y_i \in \{1, \ldots, K\}$ determined by the presence of a single object or feature. Privileged information is given by a single bounding box with coordinates $t_i \in \mathbb{R}^4$ enclosing a subset of pixels w_i corresponding to the object or features sufficient to determine the label. For example, x_i may be a photograph of a nature scene where w_i contains the pixels depicting an animal and y_i indicates its species.

We first learn $\hat{d}$ which is a function that takes target image data, $\tilde{x}_i$, and bounding box coordinates, t_i, and learns to output the bounding box coordinates, $\hat{t}_i$, which should contain the privileged information w_i. Note that we do not exactly follow the setup in (2) and do not need to actually predict the pixel values within the bounding box. If we find a good enough estimator of t_i we should minimize the loss of f in (2). To obtain the privileged information we apply a deterministic function ϕ which crops and scales an image using the associated bounding box, t_i. We can now write the composition of these two functions as $\hat{f}(x_i) = \phi(x_i, \hat{d}(x_i))$ which outputs the privileged information. The function ϕ is hard-coded and therefore not learned.

In the second step, we train $\hat{g}$ by learning to predict the label from the privileged information w_i, which is a cropped version of x_i where the cropping is defined by the bounding box t_i around the digit. This cropping and resizing is done by ϕ. When we evaluate the performance of this classifier we combine the two models into one, $\hat{h}(x) = \hat{g}(\phi(x, \hat{d}(x)))$. We use the mean squared error loss for learning $\hat{d}$ and categorical cross-entropy (CCE) loss for $\hat{g}$.

A.2. DALUPI with Faster R-CNN

For multi-label classification, we adapt Faster R-CNN (Ren et al., 2016) outlined in Figure 3 and described below. Faster R-CNN uses a region proposal network (RPN) to generate region proposals which are fed to a detection network for

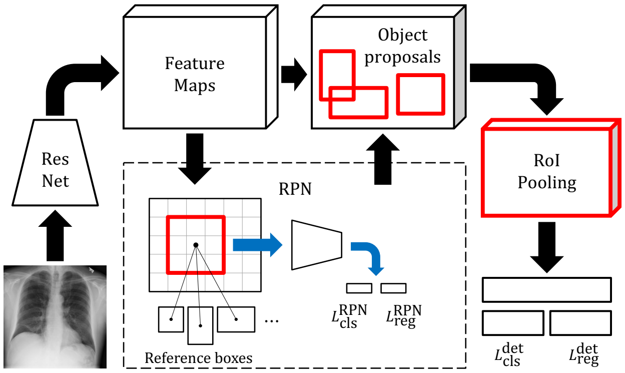

Feature
Maps
Object
proposals
Res
Net
RPN
RoI
Pooling
Reference boxes
...
$L^{\text{RPN}}_{\text{cls}}$
$L^{\text{RPN}}_{\text{reg}}$
$L^{\text{det}}_{\text{cls}}$
$L^{\text{det}}_{\text{reg}}$

Figure 3: Faster R-CNN (Ren et al., 2016). The RoI pooling layer and the classification and regression layers are part of the Fast R-CNN detection network (Girshick, 2015).

classification and bounding box regression. This way of solving the task in subsequent steps has similarities with our two-stage algorithm although Faster R-CNN can be trained end-to-end. We make small modifications to the training procedure of the original model in the end of this section.

The RPN generates region proposals relative to a fixed number of reference boxes—anchors—centered at the locations of a sliding window moving over convolutional feature maps. Each anchor is assigned a binary label $p \in \{0, 1\}$ based on its overlap with ground-truth bounding boxes; positive anchors are also associated with a ground-truth box with location t. The RPN loss for a single anchor is

$$L^{\text{RPN}}(\hat{p}, p, \hat{t}, t) := L^{\text{RPN}}_{\text{cls}}(\hat{p}, p) + pL^{\text{RPN}}_{\text{reg}}(\hat{t}, t), \tag{3}$$

where $\hat{t}$ represents the refined location of the anchor and $\hat{p}$ is the estimated probability that the anchor contains an object. The binary cross-entropy loss and a smooth L_1 loss are used for the classification loss $L^{\text{RPN}}_{\text{cls}}$ and the regression loss $L^{\text{RPN}}_{\text{reg}}$, respectively. For a mini-batch of images, the total RPN loss is computed based on a subset of all anchors, sampled to have a ratio of up to 1:1 between positive and negative ditto.

A filtered set of region proposals are projected onto the convolutional feature maps. For each proposal, the detection network—Fast R-CNN (Girshick, 2015)—outputs a probability $\hat{p}(k)$ and a predicted bounding box location $\hat{t}(k)$ for each class k. Let $\hat{p} = (\hat{p}(0), \ldots, \hat{p}(K))$, where $\sum_k \hat{p}(k) = 1$, K is the number of classes and 0 represents a catch-all background class. For a single proposal with ground-truth coordinates t and multi-class label $u \in \{0, \ldots, K\}$, the detection loss is

$$L^{\text{det}}(\hat{p}, u, \hat{t}_u, t) = L^{\text{det}}_{\text{cls}}(\hat{p}, u) + \mathbf{I}_{u \geq 1} L^{\text{det}}_{\text{reg}}(\hat{t}_u, t), \tag{4}$$

where $L^{\text{det}}_{\text{cls}}(\hat{p}, u) = -\log \hat{p}(u)$ and $L^{\text{det}}_{\text{reg}}$ is a smooth L_1 loss. To obtain a probability vector for the entire image, we maximize, for each class k, over the probabilities of all proposals.

During training, Faster R-CNN requires that all input images x come with at least one ground-truth annotation (bounding box) w and its corresponding label u. To increase sample-efficiency, we enable training the model using non-annotated but labeled samples (x, y) from the source domain and annotated but unlabeled samples $(\tilde{x}, \tilde{w})$ from the target domain. In the RPN, no labels are needed, and we simply ignore anchors from non-annotated images when sampling anchors for the loss computation. For the computation of (4), we handle the two cases separately. We assign the label $u = -1$ to all ground-truth annotations from the target domain and multiply $L^{\text{det}}_{\text{cls}}$ by the indicator $\mathbf{I}_{u \geq 0}$. For non-annotated samples (x, y) from the source domain, there are no box-specific coordinates t or labels u but only the labels y for the entire image. In this case, (4) is undefined and we instead compute the binary cross-entropy loss between binarized labels and the probability vector for the entire image.

We train the RPN and the detection network jointly as described in (Ren et al., 2016). To extract feature maps, we use a Feature Pyramid Network (Lin et al., 2017) on top of a ResNet-50 architecture (He et al., 2016). We use the modfied model—DALUPI—in the experiments in Section 4.2. Note that we may also train the model in a LUPI setting, where no information from the target domain is used.

By default, Faster R-CNN requires that all training input images x come with at least one ground-truth annotation (bounding box) w and its corresponding label u. To increase sample-efficiency, we enable training the model using non-annotated but labeled samples $(x, y) \sim \mathcal{S}$ from the source domain and annotated but unlabeled samples $(\tilde{x}, \tilde{w}) \sim \mathcal{T}$ from the target domain. We mask the classification loss so that unlabeled target samples do not influence it. For non-annotated samples (x_i, y_i) from the source domain, there are no box-specific coordinates t_{ij} or labels u_{ij} but only the per-image label y_i. In this case, instead of a per-annotation cross-entropy loss we use the binary cross-entropy loss between the per-image label and the probability vector for the entire image.

A.3. Digit classification

The domains are constructed using CIFAR-10's first five and last five classes as source and target backgrounds, respectively. Both source and target datasets contain 15 298 images each. We use $20\,\%$ of the available source and target data for testing and $20\,\%$ of the training data for validation. To increase the difficulty of the task, we make the digit be the mean color of the dataset and make the digit background transparent so that the border of the image is less distinct. This may slightly violate Assumption 2.2 w.r.t. W since the backgrounds differ between domains.

We use $20\,\%$ of the available source and target data in the test set. We likewise use $20\,\%$ of the training data for validation purposes. For the baselines SL-S and SL-T we use a ResNet-18 network without pretrained weights. We change the final fully connected layer from stock to the following sequence:

- fully connected layer with 256 neurons
- batch normalization layer
- dropout layer with $p = 0.2$
- fully connected layer with 128 neurons
- fully connected layer with 5 neurons.

For DALUPI we use a non-pretrained ResNet-18 for the function $\hat{f}$ where we replace the default fully connected layer with a fully connected layer with 4 neurons to predict the coordinates of the bounding box. The predicted bounding box is resized to a 28×28 square no matter the initial size. We use a simpler convolutional neural network for the function $\hat{g}$ with the following structure:

- convolutional layer with 16 output channels, kernel size of 5, stride of 1, and padding of 2
- max pooling layer with kernel size 2, followed by a ReLU activation
- convolutional layer with 32 output channels, kernel size of 5, stride of 1, and padding of 2
- max pooling layer with kernel size 2, followed by a ReLU activation
- dropout layer with $p = 0.4$
- fully connected layer with 50 neurons
- dropout layer with $p = 0.2$
- fully connected layer with 5 neurons.

DANN is implemented with different optimizers for the discriminator and the generator. Here, the generator is the main network consisting of a ResNet-18-based encoder and a classifier network. We use the nonlinear version of the classifier network.

The discriminator is a simple MLP network with depth and width specified as parameters. We add a gradient penalty to the discriminator loss which we control with the parameter "gradient penalty". We also include a parameter controlling the number of discriminator steps per generator steps (default 1).

The model training is stopped when the best validation error does not improve over 5 epochs or when 100 epochs have been trained, whichever occurs first. The training of the DANN model is stopped when the validation loss has not improved over 10 epochs. DANN is pretrained on ImageNet and the number of layers that are further trainable is treated as a hyperparameter.

A.3.1. Hyperparameters

We randomly choose hyperparameters from the following predefined sets of values:

- SL-S and SL-T:
  - batch size: $(16, 32, 64)$
  - learning rate: $(1.0 \times 10^{-4}$, 5.0×10^{-4} , 1.0×10^{-3} $)$
  - weight decay: $(1.0 \times 10^{-6}$, 1.0×10^{-5} , 1.0×10^{-4} , 1.0×10^{-3} $)$.
- DALUPI:
  - batch size: $(16, 32, 64)$
  - learning rate: $(1.0 \times 10^{-4}$, 5.0×10^{-4} , 1.0×10^{-3} $)$.
- DANN:
  - batch size: (64)
  - learning rate: $(1.0 \times 10^{-3}$ $)$
  - gradient penalty: $(0, 0.01, 0.1)$
  - weight decay: $(1.0 \times 10^{-4}$, 1.0×10^{-3} $)$
  - number of trainable layers (encoder): $(1, 2, 3)$
  - dropout (encoder): $(0, 0.1, 0.2, 0.5)$
  - width of discriminator network: $(64, 128, 256)$
  - depth of discriminator network: $(2, 3)$.

A.4. Entity classification

In the entity classification experiment, we train all models for at most 50 epochs. If the validation AUC does not improve for 10 subsequent epochs, we stop the training earlier. No pretrained weights are used in this experiment since we find that the task is too easy to solve with pretrained weights. For DALUPI and LUPI we use the default anchor sizes for each of the feature maps (32, 64, 128, 256, 512), and for each anchor size we use the default aspect ratios (0.5, 1.0, 2.0). We use the binary cross entropy loss for SL-S, SL-T, DANN, and MDD.

We use the 2017 version of the MS-COCO dataset (Lin et al., 2014). In Table 2 we describe the label distribution in the defined source and target domains, respectively. How the domains are defined is described in Section 4.2. All images are resized to 320×320. In total, there are 5231 images in the source domain and 5719 images in the target domain; the distribution of labels is provided in Appendix A.4. From these pools, we randomly sample 3000 , 1000 , and 1000 images for training, validation, and testing, respectively. As privileged information W, we use bounding boxes localizing the different entities, provided as annotations by MS-COCO. Covariate shift and sufficiency are likely to hold in this task; the pixels contained in a box are likely sufficient for classifying the object, whether it is indoor or outdoor.

Table 2: Marginal label distribution in source and target domains for the entity classification task based on the MS-COCO dataset. The background class contains images where none of the four entities are present.

Domain	Person	Dog	Cat	Bird	Background
Source	2963	569	1008	213	1000
Target	3631	1121	423	712	1000

A.4.1. Experiment on adding privileged information

In this experiment we study the value of adding privileged information for our multi-label task. We give LUPI access to all (x, y) samples from the source domain and increase the fraction of inputs for which PI is available, $n_{\text{PI}}(\mathcal{S})$, from 0 to 1. For DALUPI, we increase the fraction of $(\tilde{x}, \tilde{w})$ samples from the target domain, $n_{\text{PI}}(\mathcal{T})$, from 0 to 1, while keeping $n_{\text{PI}}(\mathcal{S}) = 1$. We train SL-S and SL-T using all available data and increase the fraction of unlabeled target samples used by

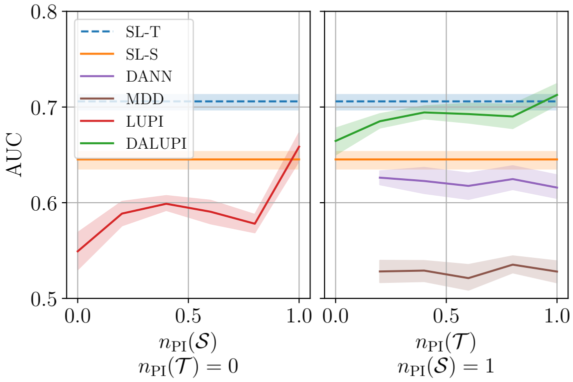


Figure 4: Entity classification, target domain AUC. The performance of SL-S and SL-T is extended across the x-axes for visual purposes. DANN and MDD use an increasing fraction of target samples $\tilde{x}$ but no PI.

DANN and MDD from 0.2 to 1 while giving them access to all data from the source domain. As we see in Figure 4, the performance of LUPI increases as $n_{\mathrm{PI}}(\mathcal{S})$ increases, but even with access to all PI in the source domain, LUPI barely beats SL-S. However, when additional $(\tilde{x}, \tilde{w})$ samples from the target domain are added, DALUPI quickly outperforms SL-S and eventually reaches the performance of SL-T. We note that DANN and MDD do not benefit as much from added unlabeled target samples as DALUPI does. Their weak performance could be explained by difficulties in adversarial training. The large gap between LUPI and SL-S for $n_{\mathrm{PI}}(\mathcal{S}) = 0$ is not too surprising; we do not expect an object detector to work well without bounding box supervision.

A.4.2. Hyperparameters

We randomly choose hyperparameters from the following predefined sets of values. For information about the specific parameters in LUPI and DALUPI, we refer to the paper by Ren et al. (2016). RoI and NMS refer to region of interest and non-maximum suppression, respectively.

- SL-S and SL-T:
  - batch size: $(16, 32, 64)$
  - learning rate: $(1.0 \times 10^{-4}, 5.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - dropout (encoder): (0, 0.1, 0.2, 0.3)
  - nonlinear classifier: (`True`, `False`).
- DANN:
  - batch size: $(16, 32, 64)$
  - learning rate: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - weight decay: $(1.0 \times 10^{-4}, 1.0 \times 10^{-3})$
  - number of trainable layers (encoder): (1, 2, 3, 4, 5)
  - dropout (encoder): (0, 0.1, 0.2, 0.3)
  - width of discriminator network: (64, 128, 256)
  - depth of discriminator network: (2, 3)
  - nonlinear classifier: (`True`, `False`).
- MDD:
  - batch size: $(16, 32, 64)$

  - learning rate: $(1.0 \times 10^{-4}\,, 1.0 \times 10^{-3}\,)$
  - weight decay: $(1.0 \times 10^{-4}\,, 1.0 \times 10^{-3}\,)$
  - number of trainable layers (encoder): (1, 2, 3, 4, 5)
  - dropout (encoder): (0, 0.1, 0.2, 0.3)
  - nonlinear classifier: (`True`, `False`).

- LUPI and DALUPI:

  - batch size: $(16, 32, 64)$
  - learning rate: $(1.0 \times 10^{-4}\,, 1.0 \times 10^{-3}\,)$
  - weight decay: $(1.0 \times 10^{-4}\,, 1.0 \times 10^{-3}\,)$
  - IoU foreground threshold (RPN): (0.6, 0.7, 0.8, 0.9)
  - IoU background threshold (RPN): (0.2, 0.3, 0.4)
  - batchsize per image (RPN): (32, 64, 128, 256)
  - fraction of positive samples (RPN): (0.4, 0.5, 0.6, 0.7)
  - NMS threshold (RPN): (0.6, 0.7, 0.8)
  - RoI pooling output size (Fast R-CNN): (5, 7, 9)
  - IoU foreground threshold (Fast R-CNN): (0.5, 0.6)
  - IoU background threshold (Fast R-CNN): (0.4, 0.5)
  - batchsize per image (Fast R-CNN): (16, 32, 64, 128)
  - fraction of positive samples (Fast R-CNN): (0.2, 0.25, 0.3)
  - NMS threshold (Fast R-CNN): (0.4, 0.5, 0.6)
  - detections per image (Fast R-CNN): (25, 50, 75, 100).

B. Proof of Proposition 3.2

Proposition B.1. *Let Assumptions 2.1 and 2.2 be satisfied w.r.t. W (not necessarily w.r.t. X) and let Assumption 3.1 hold as stated. Then, the target risk $R_\mathcal{T}$ is identified for hypotheses $h : \mathcal{X} \to \mathcal{Y}$,*

$$R_\mathcal{T}(h) = \sum_x \mathcal{T}(x) \sum_w \mathcal{T}(w \mid x) \sum_y \mathcal{S}(y \mid w) L(h(x), y)\ .$$

and, for L the squared loss, a minimizer of $R_\mathcal{T}$ is $h^*_\mathcal{T}(x) = \sum_w \mathcal{T}(w \mid x) \mathbb{E}_\mathcal{S}[Y \mid w]$.

Proof. By definition, $R_\mathcal{T}(h) = \sum_{x,y} \mathcal{T}(x,y) L(h(x), y)$. We marginalize over W to get

$$\begin{aligned}
\mathcal{T}(x,y) &= \mathcal{T}(x) \mathbb{E}_{\mathcal{T}(W|x)} \left[\mathcal{T}(Y \mid W, x) \mid x]\right] \\
&= \mathcal{T}(x) \mathbb{E}_{\mathcal{T}(W|x)} [\mathcal{T}(y \mid W) \mid x] \\
&= \mathcal{T}(x) \sum_{w : \mathcal{T}(w) > 0} \mathcal{T}(w \mid x) \mathcal{S}(y \mid w) \\
&= \mathcal{T}(x) \sum_{w : \mathcal{S}(w) > 0} \mathcal{T}(w \mid x) \mathcal{S}(y \mid w)\ .
\end{aligned}$$

where the second equality follows by sufficiency and the third by covariate shift and overlap in W. $\mathcal{T}(x), \mathcal{T}(w \mid x)$ and $\mathcal{S}(y \mid w)$ are observable through training samples. That $h^*_\mathcal{T}$ is a minimizer follows from the first-order condition of setting the derivative of the risk with respect to h to 0. This strategy yields the well-known result that

$$h^*_\mathcal{T} = \arg\min_h \mathbb{E}_\mathcal{T}[(h(X) - Y^2)] = \mathbb{E}_\mathcal{T}[Y \mid X]\ .$$

By definition and the previous result, we have that

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}[Y \mid X = x] &= \sum_y y \frac{\mathcal{T}(x,y)}{\mathcal{T}(x)} \\ &= \sum_y \sum_{w:\mathcal{S}(w)>0} \mathcal{T}(w \mid x)\mathcal{S}(y \mid w)y \\ &= \sum_w \mathcal{T}(w \mid x)\mathbb{E}_{\mathcal{S}}[Y \mid x]\end{aligned}$$

and we have the result. □

C. Generalization bound

We can bound the generalization error of estimators $\hat{h} = \hat{g} \circ \hat{f}$ when $W \in \mathbb{R}^{d_{\mathcal{W}}}$ and L is the squared loss by placing an assumption of Lipschitz smoothness on the space of prediction functions: $\forall g \in \mathcal{G}, w, w' \in \mathcal{W} : \|g(w) - g(w')\|_2 \leq M\|w - w'\|_2$. To arrive at a bound, we first define the ρ-weighted empirical risk of the outcome model g in the source domain, $\hat{R}_{\mathcal{S}}^{Y,\rho}(g) = \frac{1}{m}\sum_{i=1}^m \rho(w_i)L(g(w_i), y_i)$ where ρ is the density ratio of $\mathcal{T}$ and $\mathcal{S}$, $\rho(w) = \frac{\mathcal{T}(w)}{\mathcal{S}(w)}$. When the density ratio ρ is unknown, we may use density estimation (Sugiyama et al., 2012) or probabilistic classifiers to estimate it. We arrive at the following result, proven for univariate Y but generalizable to multivariate outcomes.

Proposition C.1. *Suppose that privileged information W and outcomes Y are deterministic in X and W, respectively, and that Assumptions 2.1–3.1 hold with respect to W. Next, let $\mathcal{G}$ comprise M-Lipschitz mappings from the privileged information space $\mathcal{W} \subseteq \mathbb{R}^{d_W}$ to $\mathcal{Y}$ and let L be the squared Euclidean distance and assume that L is uniformly bounded over $\mathcal{W}$ by a constant B. Let ρ be the domain density ratio of W and d and d' be the pseudo-dimensions of $\mathcal{G}$ and $\mathcal{F}$, respectively. Assume that there are m observations from the source (labeled) domain and n from the target (unlabeled) domain. Then, for any $h = g \circ f \in \mathcal{G} \times \mathcal{F}$, with probability at least $1 - \delta$,*

$$\begin{aligned}\frac{R_{\mathcal{T}}(h)}{2} &\leq \hat{R}_{\mathcal{S}}^{Y,\rho}(g) + M^2\hat{R}_{\mathcal{T}}^{W}(f) \\ &+ 2^{5/4}\sqrt{d_2(\mathcal{T}\|\mathcal{S})}\sqrt[\frac{3}{8}]{\frac{d\log\frac{2me}{d} + \log\frac{4}{\delta}}{m}} \\ &+ d_{\mathcal{W}}BM^2\left(\sqrt{\frac{2d'\log\frac{en}{d'}}{n}} + \sqrt{\frac{\log\frac{d_{\mathcal{W}}}{\delta}}{2n}}\right).\end{aligned}$$

Proposition 2. *Assume that $\mathcal{G}$ comprises M-Lipschitz mappings from the privileged information space $\mathcal{W} \subseteq \mathbb{R}^{d_W}$ to $\mathcal{Y}$. Further, assume that both the ground truth privileged information W and label Y are deterministic in X and W respectively. Let ρ be the domain density ratio of W and let Assumptions 2.1–3.1 (Covariate shift, Overlap and Sufficiency) hold w.r.t. W. Further, let the loss L be uniformly bounded by some constant B and let d and d' be the pseudo-dimensions of $\mathcal{G}$ and $\mathcal{F}$ respectively. Assume that there are n observations from the source (labeled) domain and m from the target (unlabeled) domain. Then, with L the squared Euclidean distance, for any $h = h \circ f \in \mathcal{G} \times \mathcal{F}$, w.p. at least $1 - \delta$,*

$$\begin{aligned}\frac{R_{\mathcal{T}}(h)}{2} &\leq \hat{R}_{\mathcal{S}}^{Y,\rho}(g) + M^2\hat{R}_{\mathcal{T}}^{W}(f) \\ &+ 2^{5/4}\sqrt{d_2(\mathcal{T}\|\mathcal{S})}\sqrt[\frac{3}{8}]{\frac{d\log\frac{2me}{d} + \log\frac{4}{\delta}}{m}} \\ &+ d_{\mathcal{W}}BM^2\left(\sqrt{\frac{2d'\log\frac{en}{d'}}{n}} + \sqrt{\frac{\log\frac{d_{\mathcal{W}}}{\delta}}{2n}}\right).\end{aligned}$$

Proof. Decomposing the risk of $h \circ \phi$, we get

$$
\begin{aligned}
&R_{\mathcal{T}}(h) = \mathbb{E}_{\mathcal{T}}[(g(f(X)) - Y)^2] \\
&\leq 2\mathbb{E}_{\mathcal{T}}[(g(W) - Y)^2 + (g(f(X)) - g(W))^2] \\
&\leq 2\mathbb{E}_{\mathcal{T}}[(g(W) - Y)^2 + M^2\|f(X)) - g(W)\|^2] \\
&\leq 2\mathbb{E}_{\mathcal{T}}[(g(W) - Y)^2] + 2M^2\mathbb{E}_{\mathcal{T}}[\|(f(X) - W)\|^2] \\
&= 2R^Y_{\mathcal{T}}(g) + 2M^2 R^W_{\mathcal{T}}(f) = 2\underbrace{R^{Y,\rho}_S(g)}_{(I)} + 2M^2\underbrace{R^W_{\mathcal{T}}(f)}_{(II)}.
\end{aligned}
$$

The first inequality follows from the relaxed triangle inequality, the second inequality from the Lipschitz property and the third equality from Overlap and Covariate shift. We will bound these quantities separately starting with (I).

We assume that the pseudo-dimension of $\mathcal{G}$, d is bounded. Further, we assume that the second moment of the density ratios, equal to the Rényi divergence $d_2(\mathcal{T}\|\mathcal{S}) = \Sigma_{w\in cG}\mathcal{T}(w)\frac{\mathcal{T}(w)}{\mathcal{S}(w)}$ are bounded and that the density ratios are non-zero for all $w \in \mathcal{G}$. Let $D_1 = \{w_i, y_i\}_{i=0}^m$ be a dataset drawn i.i.d from the source domain. Then by application of Theorem 3 from Cortes et al. (2010) we obtain with probability $1 - \delta$ over the choice of D_1,

$$
(I) = R^{Y,\rho}_S(g) \leq \hat{R}^{Y,\rho}_S(g) + 2^{5/4}\sqrt{d_2(\mathcal{T}\|\mathcal{S})}\sqrt[3/8]{\frac{d\log\frac{2me}{d} + \log\frac{4}{\delta}}{m}}
$$

Now for (II) we treat each component of $w \in \mathcal{W}$ as a regression problem independent from all the others. So we can therefore write the risk as the sum of the individual component risks

$$
R^W_{\mathcal{T}}(f) = \Sigma_{i=1}^{d_{\mathcal{W}}} R^W_{\mathcal{T},i}(f)
$$

Let the pseudo-dimension of $\mathcal{F}$ be denoted d, $D_2 = \{x_i, w_i\}_{i=0}^n$ be a dataset drawn i.i.d from the target domain. Then, using theorem 11.8 from Mohri et al. (2018) we have that for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of D_2, the following inequality holds for all hypotheses $f \in \mathcal{F}$ for each component risk

$$
R^W_{\mathcal{T},i}(f) \leq \hat{R}^W_{\mathcal{T},i}(f) + B\left(\sqrt{\frac{2d'\log\frac{en}{d'}}{n}} + \sqrt{\frac{\log\frac{1}{\delta}}{2n}}\right)
$$

We then simply make all the bounds hold simultaneously by applying the union bound and having it so that each bound must hold with probability $1 - \frac{\delta}{d_{\mathcal{W}}}$ which results in

$$
\begin{aligned}
R^W_{\mathcal{T}}(f) = \Sigma_{i=1}^{d_{\mathcal{W}}} R^W_{\mathcal{T},i}(f) &\leq \Sigma_{i=1}^{d_{\mathcal{W}}} \hat{R}^W_{\mathcal{T},i}(f) + \Sigma_{i=1}^{d_{\mathcal{W}}} B\left(\sqrt{\frac{2d'\log\frac{en}{d'}}{n}} + \sqrt{\frac{\log\frac{d_{\mathcal{W}}}{\delta}}{2n}}\right) \\
&= \hat{R}^W_{\mathcal{T}}(f) + d_{\mathcal{W}}B\left(\sqrt{\frac{2d'\log\frac{en}{d'}}{n}} + \sqrt{\frac{\log\frac{d_{\mathcal{W}}}{\delta}}{2n}}\right)
\end{aligned}
$$

Combination of these two results then yield the proposition statement.

Consistency follows as Y is a deterministic function of W and W is a deterministic fundtion of X and both $\mathcal{H}$ and $\mathcal{F}$ are well-specified. Thus both empirical risks and sample complexity terms will converge to 0 in the limit of infinite samples. □

When $\mathcal{F}$ and $\mathcal{G}$ contain the ground-truth mappings between X and W and between W and Y, in the infinite-sample limit, minimizers of (2) minimize $R_{\mathcal{T}}$ as well.

D. A bound on the target risk without suffiency

The sufficiency assumption is used to replace $\mathcal{T}(y \mid x)$ with $\mathcal{T}(y \mid w)$ in the proof of Proposition 3.2. If sufficiency is violated but it is plausible that the degree of insufficiency is comparable across domains, we can still obtain a bound on the target risk which may be estimated from observed quantities. One way to formalize such an assumption is that there is some $\gamma \geq 1$, for which

$$\sup_{x \in \mathcal{T}(x \mid w)} \mathcal{T}(y \mid w, x)/\mathcal{T}(y \mid w) \leq \gamma \sup_{x \in \mathcal{S}(x \mid w)} \mathcal{S}(y \mid w, x)/\mathcal{S}(y \mid w) \tag{5}$$

This may be viewed as a relaxation of suffiency. If Assumption 3.1 holds, both left-hand and right-hand sides of the inequality are 1. Under (5), with $\Delta_\gamma(w, y)$ equal to the right-hand side the inequality,

$$R_\mathcal{T}(h) \leq \sum_x \mathcal{T}(x) \sum_w \mathcal{T}(w \mid x) \sum_y \Delta_\gamma(w, y) \mathcal{S}(y \mid w) L(h(x), y) \ .$$

However, the added assumption is not verifiable statistically.