Feed-Forward Source-Free Latent Domain Adaptation via Cross-Attention

Anonymous Authors[1]

Abstract

We study the highly practical but comparatively under-studied problem of latent-domain adaptation, where a source model should be adapted to a target dataset that contains a mixture of unlabelled domain-relevant and domain-irrelevant examples. Motivated by the requirements for data privacy and the need for embedded and resource-constrained devices of all kinds to adapt to local data distributions, we further focus on the setting of feed-forward source-free domain adaptation, where adaptation should not require access to the source dataset, and also be back propagation-free. Our solution is to meta-learn a network capable of embedding the mixed-relevance target dataset and dynamically adapting inference for target examples using cross-attention. The resulting framework leads to consistent strong improvements.

1. Introduction

Domain shift presents a real-world challenge for applying pre-trained models because performance degrades when deployment data are not from the training data distribution. For example, a model that has been only trained on day-time images will perform poorly when presented with night-time images. This issue is ubiquitious as it is often impossible or prohibitively costly to pre-collect and annotate training data that is sufficiently representative of test data statistics. The field of domain adaptation (Kouw & Loog, 2021; Csurka et al., 2022) has therefore attracted a lot of attention with the promise of adapting models during deployment to perform well using only unlabeled deployment data.

We make two main contributions: A conceptual contribution, framing domain adaptation in a new highly practical way; and an algorithm for effective domain adaptation in these conditions.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the First Workshop of Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022. Do not distribute.

Latent domain adaptation While domain adaptation is now very well studied (Kouw & Loog, 2021; Csurka et al., 2022), the vast majority of work assumes that suitable meta-data is available in order to correctly group instances into one or more subsets (domains) that differ statistically across groups, while being similar within groups. We join a growing minority (Mancini et al., 2021; Deecke et al., 2022; Hoffman et al., 2014; Wang et al., 2022) in arguing that this is an overly restrictive assumption that does not hold in most real applications of interest. On one hand some datasets or collection processes may not provide meta-data suitable for defining domain groupings. Alternatively, for other data sources that occur with rich meta-data there may be no obviously correct grouping and existing domain definitions may be sub-optimal (Deecke et al., 2022). Consider the popular iWildCam (Beery et al., 2020) benchmark for animal detection within the WILDS (Koh et al., 2021) suite. The default setup within WILDS defines domains by camera ID. But given that images span different weather conditions and day/night cycles, such domains may neither be internally homogenous, nor similarly distinct. For example there may be more transferability between images from nearby cameras at similar times of day than between images from the same camera taken on a sunny day vs a snowy night. As remarked by some (Hoffman et al., 2014; Wang et al., 2022), domains may more naturally define a continuum, rather than discrete groups. And that continuum may even be multi-dimensional – such as timestamp of image and spatial proximity of cameras. In this paper, we propose a flexible formulation of the domain adaptation problem that can span all these situations where domains are hard to define, while aligning with the requirements of real use cases.

Feed-forward and source-free conditions Unsupervised domain adaptation aims to adapt models from source datasets (e.g. ImageNet) to the peculiarities of specific data distributions in the wild. The mainstream line of work here uses labelled source domain data alongside unlabelled target domain data and updates the model so it performs well on the target domain using back-propagation (Kouw & Loog, 2021; Csurka et al., 2022). However, the key use cases motivating domain adaptation are edge devices such as autonomous vehicles, smartphones and hospital scanners. Storing and processing large source datasets on such devices is usually infeasible. This has led a growing number

of studies to investigate the *source-free* condition (Liang et al., 2020), where a pre-trained model is distributed and adapted using solely unlabeled target data.

In this paper, we go further in considering the practical requirements of an edge device, namely that most edge devices are not designed in either hardware or software stack to support back-propagation. This leads us to focus on the *feed-forward* condition where adaptation algorithms should proceed using only feed-forward operations. For example, simply updating batch normalisation statistics, which can be done without back-propagation, provides a strong baseline for adaptation (Schneider et al., 2020; Zhang et al., 2021).

Feed-forward source-free latent domain adaptation Bringing these ideas together, we envisage a setup where edge devices maintain an unlabeled target dataset that need not be a cleanly meta-data induced domain in the conventional sense, but which may contain examples relevant to the inference of test instances. Instances in the target set may be of varied relevance to a given test instance. For example, if true instance relevance is a function of timestamp similarity. These target examples should then drive model adaptation on the fly, leveraging neither source data, nor back-propagation.

To solve the challenge posed above, we propose a feed-forward adaptation framework based on cross-attention between test instances and the target set. The cross-attention module is meta-learned based on a set of training domains, inspired by (Zhang et al., 2021). During deployment it flexibly enables each inference operation to draw upon any part of the target set, exploiting each target instance to a continuous degree. For example, this could potentially exclude transfer from target instances that would be conventionally in-domain (e.g., same camera/opposite time of day example earlier), include transfer from target instances that would conventionally be out-of-domain (e.g., similar images/different camera example earlier), and continuously weight similarity to each target image (e.g., temporal distance of images taken in sequence). Our experiments show that our cross-attention approach provides useful adaptation in this highly practical setting across a variety of synthetic and real benchmarks.

2. Methods

2.1. Set-up

Preliminaries During deployment the high-level assumption made by many source-free domain adaptation frameworks is that we are provided with a predictive model f_ψ and an unlabeled target dataset $\boldsymbol{x}_s$ whose label-space is the same as that of the pre-trained model (Liang et al., 2020). Given these, source-free DA approaches define some algorithm $\mathcal{A}$ that ultimately leads to classifying a test instance x_q as $y_q \approx \hat{y}_q = \mathcal{A}(x_q, \boldsymbol{x}_s, \psi)$. There are numerous existing algorithms for this. For example, pseudo-label strategies (Liang et al., 2020; Li et al., 2020; Yang et al., 2021) proceed by estimating labels $\hat{\boldsymbol{y}}_s$ for the target set $\boldsymbol{x}_s$, treating these as ground-truth, back-propagating to update the model ψ' such that it predicts $\hat{\boldsymbol{y}}_s$, and then classifying the test point as $f_{\psi'}(x_q)$. We address the *feed-forward* setting where algorithm $\mathcal{A}$ should not use back-propagation. For example, BN-based approaches (Schneider et al., 2020; Zhang et al., 2021) use the target set $\boldsymbol{x}_s$ to update the BN statistics in ψ as ψ' and then classify the test point as $f_{\psi'}(x_q)$.

While the conventional domain adaptation setting assumes that x_q and $\boldsymbol{x}_s$ are all drawn from a common distribution, the *latent domain* assumption has no such requirement. For example, $\boldsymbol{x}_s$ may be drawn from a mixture distribution and x_q may be drawn from only one component of that mixture. In this case only a subset of elements in $\boldsymbol{x}_s$ may be relevant to adapting the inference for x_q.

Deployment phase Rather than explicitly updating model parameters, we aim to define a flexible inference routine f_ψ that processes both x_q and $\boldsymbol{x}_s$ to produce $\hat{y}_q$ in a feed-forward manner, i.e., $\hat{y}_q = \mathcal{A}(x_q, \boldsymbol{x}_s, \psi) = f_\psi(x_q, \boldsymbol{x}_s)$. In this regard our inference procedure follows a similar flow to variants of ARM (Zhang et al., 2021), with the following key differences: (1) ARM is transductive: it processes a batch of instances at once without distinguishing test instances and target adaptation set, so elements x_q are members of $\boldsymbol{x}_s$. (2) ARM makes the conventional domain-observed assumption that domains have been defined by an external process that ensures all x_q and $\boldsymbol{x}_s$ are drawn from the same distribution. We do not make this assumption and require robustness to irrelevant elements in $\boldsymbol{x}_s$.

Training phase To train a model than can be used as described above, we follow an episodic meta-learning paradigm (Zhang et al., 2021; Hospedales et al., 2021). This refers to training f_ψ using a set of simulated domain adaptation tasks. At each iteration, we generate a task with a unique pair of query and support instances $(\boldsymbol{x}_s, (y_q, x_q))$ keeping label space the same across all tasks. We simulate training episodes where $\boldsymbol{x}_s$ contains instances with varying relevance to x_q. The goal is for f_ψ to learn how to select and exploit instances from $\boldsymbol{x}_s$ in order to adapt inference for x_q to better predict y_q.

In particular, our task sampler defines each task as having support examples uniformly sampled across a random set of N_D domains, with the query example being from one of these domains. More formally, each task can be defined as $\mathcal{T} = \{\{x_{s,1}, x_{s,2}, \ldots x_{s,N_s}\}, x_q, y_q\}$ for N_s unlabelled support examples $x_{s,.}$ and query example x_q with label y_q. We give an example of a task in Figure 1 for $K = 3$ domains with $N_s = 3$ support examples and $N_q = 1$ query example.

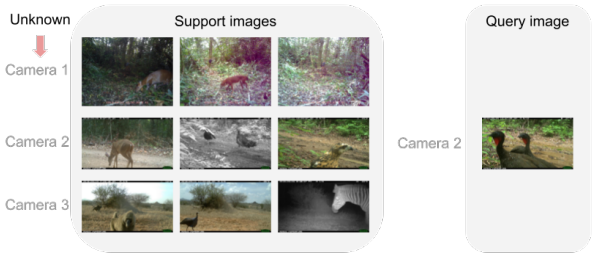

Figure 1. Illustration of how the latent domain adaptation tasks are structured. Support images come from a variety of domains and do not have any class or domain labels.

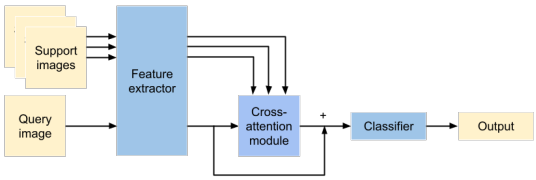

Figure 2. Illustration of the overall architecture.

2.2. Architecture

The key to solving the proposed problem is defining an architecture f_ψ that can identify and exploit relevant support instances within $\boldsymbol{x}_s$. Our solution relies on cross-attention between query and support images, as illustrated in Figure 2. We first embed the support and query examples using the same feature extractor, after which we pass the embeddings through the cross-attention module. The cross-attention module gives us transformed query examples that are then added to the embeddings of the query examples as a residual connection, after which the classifier makes predictions.

Cross-attention module Given a set of support examples $\boldsymbol{x}_s$ and query examples $\boldsymbol{x}_q$, we use the feature extractor $f_\boldsymbol{\theta}$ to extract features $f_\boldsymbol{\theta}(\boldsymbol{x}_s)$, $f_\boldsymbol{\theta}(\boldsymbol{x}_q)$. Cross-attention module $\texttt{CA}_\boldsymbol{\omega}(f_\boldsymbol{\theta}(\boldsymbol{x}_s); f_\boldsymbol{\theta}(\boldsymbol{x}_q))$ parameterized by $\boldsymbol{\omega}$ then transforms query embeddings $f_\boldsymbol{\theta}(\boldsymbol{x}_q)$, using support embeddings $f_\boldsymbol{\theta}(\boldsymbol{x}_s)$ as keys. The output of the cross-attention module is added to the query example features as a residual connection, which is then used by the classifier $f_\boldsymbol{\phi}$ to predict labels of the query examples $\hat{\boldsymbol{y}}_q = f_\boldsymbol{\phi}(f_\boldsymbol{\theta}(\boldsymbol{x}_q) + \texttt{CA}_\boldsymbol{\omega}(f_\boldsymbol{\theta}(\boldsymbol{x}_s); f_\boldsymbol{\theta}(\boldsymbol{x}_q)))$. In our notation $\boldsymbol{\psi} = \{\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}\}$.

The cross-attention module performs image-to-image cross-attention, rather than patch-to-patch. After extracting the features we flatten all spatial dimensions and channels into one vector, which represents the whole image. Image-to-image attention is more suitable for domain adaptation than patch-based option because the overall representation should better capture the nature of the domain rather than a patch. Another benefit of image-to-image attention is efficiency – we attend to the whole image rather than patches, making the overall computations manageable even with more images.

Algorithm 1 Episodic meta-learning for source-free latent domain adaptation with CXDA

`// Meta-training`

Require: # training steps T, # latent domains in a task N_D, # support examples N_s, # query examples N_q, learning rate η

Initialize: $\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}$
for $t = 1, \ldots, T$ **do**
  Sample N_D support domains $\{\mathbb{D}_s\}_1^{N_D}$
  Sample query domain $\mathbb{D}_q$ from support domains $\{\mathbb{D}_s\}_1^{N_D}$
  Sample N_s unlabelled support images $\boldsymbol{x}_s$ uniformly from the selected support domains $\{\mathbb{D}_s\}_1^{N_D}$
  Sample N_q labelled query images $\boldsymbol{x}_q, \boldsymbol{y}_q$ from domain $\mathbb{D}_q$
  Predict $\hat{\boldsymbol{y}}_q = f_\boldsymbol{\phi}(f_\boldsymbol{\theta}(\boldsymbol{x}_q) + \texttt{CA}_\boldsymbol{\omega}(f_\boldsymbol{\theta}(\boldsymbol{x}_s); f_\boldsymbol{\theta}(\boldsymbol{x}_q)))$
  $(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) - \eta \nabla_{(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega})} \sum_{k=1}^{N_q} \ell(\hat{y}_{q,k}, y_{q,k})$
end for

`// Inference on a new task`

Require: $\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}$, support $\boldsymbol{x}_s$ and query $\boldsymbol{x}_q$ examples

$\hat{\boldsymbol{y}}_q = f_\boldsymbol{\phi}(f_\boldsymbol{\theta}(\boldsymbol{x}_q) + \texttt{CA}_\boldsymbol{\omega}(f_\boldsymbol{\theta}(\boldsymbol{x}_s); f_\boldsymbol{\theta}(\boldsymbol{x}_q)))$

Our cross-attention module is parameterized by a set of learnable projection matrices $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v$ (all of size $\mathbb{R}^{C \times (C/R)}$) with additional projection matrix $\boldsymbol{W} \in \mathbb{R}^{(C/R) \times C}$ to transform the queried outputs (we refer to all of these parameters collectively as $\boldsymbol{\omega}$). The output of the feature extractor $f_\boldsymbol{\theta}$ is flattened into one vector, giving C channels, so $f_\boldsymbol{\theta}(\boldsymbol{x}_q) \in \mathbb{R}^{N_q \times C}$, $f_\boldsymbol{\theta}(\boldsymbol{x}_s) \in \mathbb{R}^{N_s \times C}$. We also specify ratio R that leads to rectangular projection matrices with fewer parameters, improving efficiency and providing regularization. Formally we express $\texttt{CA}_\boldsymbol{\omega}$ as:

$$\boldsymbol{q} = f_\boldsymbol{\theta}(\boldsymbol{x}_q)\boldsymbol{W}_q, \quad \boldsymbol{k} = f_\boldsymbol{\theta}(\boldsymbol{x}_s)\boldsymbol{W}_k, \quad \boldsymbol{v} = f_\boldsymbol{\theta}(\boldsymbol{x}_s)\boldsymbol{W}_v,$$

$$\boldsymbol{A} = \texttt{softmax}\left(\boldsymbol{q}\boldsymbol{k}^T / \sqrt{C/h}\right), \quad \texttt{CA}_\boldsymbol{\omega}(f_\boldsymbol{\theta}(\boldsymbol{x}_s)) = \boldsymbol{A}\boldsymbol{v}.$$

Similarly as CrossViT (Chen et al., 2021) and self-attention more broadly, we use multiple heads h and layer norm.

2.3. Meta-Learning

We train the model by meta-learning across many tasks. Meta-learning is efficient here as the inner loop does not include back-propagation based optimization – the adaptation to the support examples is done purely feed-forward. Both meta-training and inference are summarized in Algorithm 1.

3. Experiments

3.1. Benchmarks

We evaluate our approach on a variety of synthetic and real-world benchmarks, namely FEMNIST (Caldas et al.,

Table 1. Main results on synthetic and real-world benchmarks: average and worst-case (worst 10% tasks) test performance, with standard error of the mean across 3 random seeds. Accuracy is reported for all except iWildCam, where F1 score is used (%).

	FEMNIST		CIFAR-10-C		TINYIMAGENET-C		IWILDCAM	
APPROACH	W10%	AVG	W10%	AVG	W10%	AVG	W10%	AVG
ERM	52.7 ± 1.4	77.2 ± 0.9	44.3 ± 0.5	68.6 ± 0.3	4.8 ± 0.2	26.4 ± 0.4	0.0 ± 0.0	38.7 ± 0.8
BN	52.2 ± 1.5	78.0 ± 0.7	45.4 ± 0.7	69.3 ± 0.4	5.9 ± 0.2	27.7 ± 0.3	1.9 ± 1.1	42.5 ± 0.8
CML	50.4 ± 1.3	76.0 ± 0.9	44.8 ± 0.5	69.5 ± 0.5	4.8 ± 0.5	25.7 ± 0.6	0.0 ± 0.0	38.7 ± 1.1
OUR CXDA	53.3 ± 0.6	78.3 ± 0.0	49.4 ± 0.6	72.0 ± 0.3	6.5 ± 0.2	28.6 ± 0.3	3.6 ± 1.5	43.5 ± 1.5

2018), CIFAR-10-C (Hendrycks & Dietterich, 2019), TinyImageNet-C (Hendrycks & Dietterich, 2019) and iWildCam (Beery et al., 2020). We follow the splits into meta-training, meta-validation and meta-testing sets as selected in (Zhang et al., 2021) and (Koh et al., 2021).

3.2. Baselines

We consider three baselines: 1) Empirical risk minimization (ERM) that trains on all training domains and performs no domain adaptation. 2) Batch normalization (BN) statistics update that uses support examples to update the statistics. 3) Contextual meta-learning (CML) (Zhang et al., 2021) that extracts information from the support examples using a context network and uses it as additional channels for adaptation. BN and CML assume examples from one domain, rather than mixture of both relevant and irrelevant domains, so it is unclear how successful they will be in LDA.

3.3. Implementation Details

Our solution - CXDA In line with existing literature (Vaswani et al., 2017; Chen et al., 2021) we use 8 heads and layer normalization on the flattened features of support and query images. The use of layer normalisation means our approach does not rely on a minibatch of query examples i.e. it natively supports streaming mode and does not need mutiple query examples to obtain strong results, unlike existing test-time domain adaptation approaches (Zhang et al., 2021; Wang et al., 2021).

Support images are projected into keys and values, while query images act as queries for cross-attention after transformation by a projection matrix. After calculating the attention map and applying it to the values, we multiply the output by a further projection matrix. We use only one cross-attention layer and our projection matrices have rectangular shape of $C \times C/2$ where C is the dimensionality of the flattened features. No dropout is used.

Data augmentation We use weak data augmentation during meta-training – cropping, horizontal flipping, small rotations (up to 30 degrees). These are different from the corruptions tested in some of the benchmarks and are applied independently with probability 0.5.

Task sampling Our tasks have 5 support domains, with 20 examples in each, overall 100 support examples. Query examples come from one randomly selected support set domain and there are 20 of them. The method fully supports streaming mode, so no statistics are calculated across the minibatch and it works independently for each. There are 420, 11000, 11000 and 2125 test tasks for FEMNIST, CIFAR-10-C, TinyImageNet-C and iWildCam respectively.

Training We follow the hyperparameters used in (Zhang et al., 2021) for FEMNIST, CIFAR-10-C and TinyImageNet-C, and we also train the cross-attention parameters with the same optimizer. For FEMNIST and CIFAR-10-C a small CNN model is used, while for TinyImageNet-C a pre-trained ResNet-50 (He et al., 2015) is fine-tuned. For iWildCam we follow the hyperparameters selected in (Koh et al., 2021), but with images resized to 112×112, training for 50 epochs and with mini-batch size resulting from our task design (100 support and 20 query examples). All our experiments are repeated across three random seeds.

3.4. Results

We report our results in Table 1 and include both average performance as well as reliability via the worst case performance on the most challenging 10% tasks. From the results we can see our cross-attention approach results in consistent improvements over the strong ERM baseline across all benchmarks, as well as the CML and BN baselines. Overall we see CML and BN strategies that naively combine information from all support examples have limited success when the support set has both domain relevant and domain irrelevant examples. The results confirm our proposed mechanism based on cross-attention can successfully select useful information from the set of examples with both relevant and irrelevant examples and achieve superior performance.

4. Conclusion

We have introduced a new highly practical setting where we adapt a model using examples that come from a mixture of domains and are without domain or class labels. To answer this new highly challenging adaptation problem, we have developed a novel solution based on cross-attention that is able to automatically select relevant examples and use them for adaptation on the fly.

References

Beery, S., Cole, E., and Gjoka, A. The iWildCam 2020 competition dataset. In *arXiv*, 2020.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., and Talwalkar, A. LEAF: A benchmark for federated settings. In *Workshop on Federated Learning for Data Privacy and Confidentiality*, 2018.

Chen, C.-F., Fan, Q., and Panda, R. CrossViT: Cross-attention multi-scale vision transformer for image classification. In *ICCV*, 2021.

Csurka, G., Hospedales, T. M., Salzmann, M., and Tommasi, T. Visual domain adaptation in the deep learning era. *Synthesis Lectures on Computer Vision*, 11(1):1–190, 2022.

Deecke, L., Hospedales, T., and Bilen, H. Visual representation learning over latent domains. In *ICLR*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2015.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

Hoffman, J., Darrell, T., and Saenko, K. Continuous manifold based adaptation for evolving visual domains. In *CVPR*, 2014.

Hospedales, T. M., Antoniou, A., Micaelli, P., and Storkey, A. J. Meta-learning in neural networks: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 2021.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

Kouw, W. M. and Loog, M. A review of domain adaptation without target labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):766–785, 2021.

Li, R., Jiao, Q., Cao, W., Wong, H.-S., and Wu, S. Model adaptation: unsupervised domain adaptation without source data. In *CVPR*, 2020.

Liang, J., Hu, D., and Feng, J. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.

Mancini, M., Porzi, L., Bulo, S. R., Caputo, B., and Ricci, E. Inferring latent domains for unsupervised deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):485–498, 2021.

Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., and Bethge, M. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

Wang, Q., Fink, O., Van Gool, L., and Dai, D. Continual test-time domain adaptation. In *CVPR*, 2022.

Yang, S., Wang, Y., van de Weijer, J., Herranz, L., and Jui, S. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In *NeurIPS*, 2021.

Zhang, M., Marklund, H., Dhawan, N., Gupta, A., Levine, S., and Finn, C. Adaptive risk minimization: learning to adapt to domain shift. In *NeurIPS*, 2021.