A Simple Test-time Adaptation Method for Source-free Domain Generalization

Vaasudev Narayanan, Sai Srinivas Kancheti, Sriranjani Ramakrishnan, Vineeth N Balasubramanian
Department of Computer Science & Engineering,
Indian Institute of Technology, Hyderabad, India
{cs20mtech11001,cs21resch01004}@iith.ac.in
sriranjani.ramakrish@gmail.com, vineethnb@iith.ac.in

Abstract

In this paper, we tackle source-free domain generalization (SFDG), where the objective is to perform well on an unseen target domain using only models trained on source domains, without assuming any access to labeled source images. We propose an effective, yet simple method for solving SFDG by using *unlabeled* target data available only during inference to give a dynamic, adaptive prediction at the batch-level. Specifically, during test-time, we (1) pass the test batch through each source model, (2) select as pseudo-label the class with the highest average probability score, (3) minimize cross-entropy loss for each model using the pseudo-label and finally (4) forward pass through the adapted models and predict the class with the highest average probability. We compare our test-time pseudo-labeling method *TEPLA*, with a wide variety of baselines and outperform them on average accuracy across four benchmark DG datasets, namely PACS, OfficeHome, VLCS and TerraIncognita.

1 Introduction

Machine learning models deployed in the real world often encounter out-of-training domain samples including distribution shifts due to weather (Volk et al., 2019), illumination (Dai & Gool, 2018) or location (Varma et al., 2019) conditions. Deep neural networks (DNNs) are known to show performance deterioration in the presence of domain shift (Gulrajani & Lopez-Paz, 2020). A plethora of problem settings and methods (Ben-David et al., 2010; Muandet et al., 2013; Narayanan et al., 2022; Ahmed et al., 2021) have been proposed in recent years to build models which are robust to domain shift. Settings differ in the varying degrees of assumptions that they make on the availability of labeled data in the source and target domains, during training time as well as during inference (see Sec. A). For instance, Unsupervised Domain Adaptation (UDA) methods learn a model by assuming concurrent access to labeled source data and unlabeled target data, whereas DG learns from only a labeled set of source domains while making no assumptions about the target.

Although existing DG methods have been important stepping stones towards building practical models robust to domain shift, these methods require access to data from the source domains, while learning a model for an unseen target domain. Governments are increasingly making stronger personal data sharing laws, inducing privacy concerns in sharing source domain data. In certain other cases, sharing source data might simply not be possible due to privacy, bandwidth, management and storage limitations. We thus propose to tackle the problem setting of source-free domain generalization (SFDG) (Frikha et al., 2021), where the objective is to learn a model that can perform well on an unseen target domain using only domain-specific models trained on source domains, without any explicit access to the labeled source data. By working with only domain-specific models and not the data, SFDG presents a challenging and practical problem setting, which addresses both data privacy concerns as well as bandwidth and storage issues.

In a parallel line of work, there is growing interest in using *unlabeled* data available at test-time to adapt the model on the fly during inference (Wang et al., 2020; Boudiaf et al., 2022; Sun et al., 2020). Since unlabeled test samples are available only during inference, test-time adaptation (TTA) methods use these test samples to optimize on some unsupervised objective, and adapt a DNN model to the test samples before providing its predictions on them. When test data comes from outside the training distribution, TTA can be beneficial in adapting to an incoming batch of test samples, rather than simply using a model frozen after training.

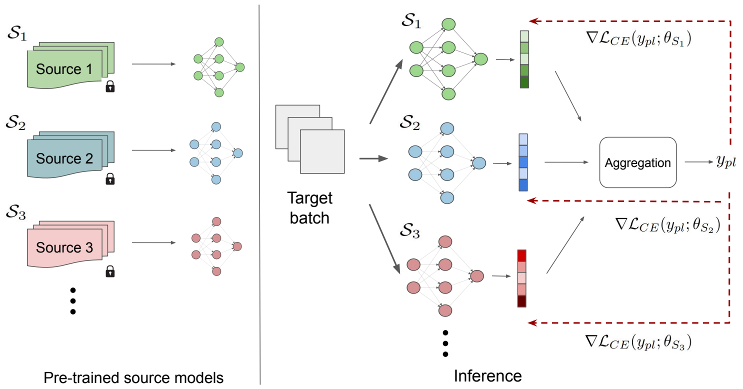

$\mathcal{S}_1$
Source 1
$\mathcal{S}_2$
Source 2
$\mathcal{S}_3$
Source 3
Pre-trained source models
Target batch
$\mathcal{S}_1$
$\mathcal{S}_2$
$\mathcal{S}_3$
Aggregation
y_{pl}
$\nabla\mathcal{L}_{CE}(y_{pl};\theta_{S_1})$
$\nabla\mathcal{L}_{CE}(y_{pl};\theta_{S_2})$
$\nabla\mathcal{L}_{CE}(y_{pl};\theta_{S_3})$
Inference

Figure 1: Overview of our proposed approach TEPLA. **Left**: We are given domain-specific models for each source S_i which were trained using empirical risk minimization. **Right**: For each target image, we get the distribution over classes from each model, take a soft aggregate of the probabilities for each class and choose the pseudo-label y_{pl} as the most probable class. We then update each model by minimizing cross-entropy loss using the pseudo-label. Finally, we make a forward pass through the adapted models and output the soft average label as the prediction.

We propose to solve SFDG by leveraging unlabeled data from the target domain *available only during inference* via a simple test-time pseudo-labeling method. Specifically, as shown in Figure 1 during inference, we: (1) forward-pass the test batch through each source model; (2) select as pseudo-label the class with the highest soft aggregate probability score; (3) minimize cross-entropy loss for each model using the selected pseudo-label; and finally (4) forward pass through the adapted models and output the soft average label as the prediction. To the best of our knowledge, giving an adaptive prediction on an unseen target domain, assuming access only to source models trained on different domains, is yet unexplored. We evaluate our simple method, TEst-time Pseudo-Labeling & Averaging (TEPLA) on four benchmark DG datasets – PACS (Li et al., 2017), OfficeHome (Venkateswara et al., 2017), VLCS (Fang et al., 2013) and TerraIncognita (Beery et al., 2018). We compare our method with: (1) multiple *common sense* baseline methods (methods that one may expect to work on this setting as simple adaptations of well-known strategies); (2) a competing SFDG method; (3) single-source source-free unsupervised domain adaptation (SF-UDA) methods; and (4) multi-source source-free UDA (MSF-UDA) methods adapted to TTA. Our results show that the proposed method outperforms any of these baselines across the considered benchmark datasets.

Our contributions are summarized as follows: (1) We propose TEPLA, a simple method for solving source-free domain generalization, by leveraging unlabeled data during inference (test-time adaptation); (2) We establish and compare with a variety of sensible baselines, and outperform them on average accuracy on four DG datasets – PACS, OfficeHome, VLCS and TerraIncognita; (3) We show connections between our proposed method with self-training and entropy minimization approaches.

2 TEPLA: Test-time Pseudo Labeling & Averaging for Source-free Domain Generalization

2.1 Problem Formulation

We assume access to domain-specific models trained on M source domains $\{D_i^s\}_{i=1}^M$ denoted by $\{f_{\theta_i}^s\}_{i=1}^M$. Here each model $f_{\theta_i}^s : \mathcal{X} \to \Delta_{|\mathcal{Y}|}$ is parameterized by θ_i, where $\mathcal{X}$ is the input space, $\mathcal{Y}$ is the label space, and $\Delta_{|\mathcal{Y}|}$ is the probability simplex over the labels. We drop the superscript s and denote $\theta = [\theta_1^\intercal\ \theta_2^\intercal \dots \theta_M^\intercal]^\intercal$ as the vector of parameters of all source models. Note that we only have access to the trained models $\{f_{\theta_i}\}_{i=1}^M$ and not the training samples from each domain. Given the set of domain-specific classifiers the objective is to classify a batch of unlabeled instances $\mathcal{B}$ from an unseen target domain D^t. The unlabeled batch $\mathcal{B}$ can be used to optimize an unsupervised objective to adapt each f_{θ_i} and output the prediction for that batch. Additionally, we can only use the current batch $\mathcal{B}$ which has arrived for inference – in the real-world where data arrives sequentially, we cannot wait for a new batch to arrive for improving the model to give a better prediction on the current batch.

2.2 Methodology

We now describe TEPLA, our simple method for source-free domain generalization. We are given a batch $\mathcal{B} = \{x_j\}_{j=1}^N$ of N samples and a set of domain-specific classifiers $\{f_{\theta_i}\}_{i=1}^M$. For each test-point x_j we obtain class-probabilities (or soft-labels), from each source model to get $\{f_{\theta_i}(x_j)\}_{i=1}^M$. We then perform an aggregation of all soft-labels to get the hard pseudo-label as $y_j^{pl} = \arg\max_{y\in\mathcal{Y}} \frac{1}{M}\sum_{i=1}^M f_{\theta_i}(x_j)$. Let $\mathcal{L}(\mathcal{B}, \theta_i) = \frac{1}{N}\sum_{j=1}^N \mathcal{L}(y_j^{pl}, f_{\theta_i}(x_j))$ be the average cross-entropy loss for the batch w.r.t source model i. We perform one gradient descent step on the sum of all average cross-entropy losses $\sum_{i=1}^M \mathcal{L}(\mathcal{B}, \theta_i)$ and update all source model parameters as:

$$\theta' = \theta - \eta \sum_{i=1}^{M} \nabla \mathcal{L}(\mathcal{B}; \theta_i) \tag{1}$$

where $\mathcal{L}$ is the average cross-entropy loss, and θ is the vector of all parameters as defined in sec. 2.1. Note that in the case of the hard pseudo-label, eq 1 disentangles into M separate updates, one for each source model – but there is still an interaction between the source-models due to the pseudo-label. The final prediction for $x_j \in \mathcal{B}$ is obtained by taking an average of predictions made by all the updated source models:

$$y_j = \arg\max_{y\in\mathcal{Y}} \frac{1}{M} \sum_{i=1}^{M} f_{\theta_i'}(x_j) \tag{2}$$

It is worth noting that only batch-level access to the test data is assumed, and that too only during inference. We assume that the models are deployed, and that the adaptation and prediction for the current samples has to be performed at that instant; we cannot wait for more data to come in to improve the model.

Connections with entropy minimization and self-training Entropy minimization and its variants have proven successful in out-of-distribution (OOD) generalization (Wang et al., 2020). The connection between self-training (Zou et al., 2019) and entropy minimization has been explored in the past (Chen et al., 2020; Goyal et al., 2022), which we exploit in our method – minimizing the classifier's prediction entropy is equivalent to minimizing the cross-entropy when the target pseudo-label is the soft label obtained from a forward pass of the data on the classifier itself. However, classical self-training does not incorporate information from other source-models. TEPLA combines information from multiple domains by selecting a pseudo-label that is an aggregate prediction of all source models. We also experiment with a soft pseudo-labeling variant using the soft pseudo-label $\bar{y}_j^{pl} = \frac{1}{M}\sum_{i=1}^M f_{\theta_i}(x_j)$.

3 Experiments

3.1 Baselines

We compare our method with the following baselines: (1) *Majority Voting:* Output the prediction as a majority vote of all source models; (2) *Lowest Entropy:* Selecting the model which has the highest confidence as measured by the lowest entropy; (3) *Entropy Weighting:* Weighted average of the source model predictions, where the weights are the inverse entropy of predictions; (4) *Random Source:* Select a random source model and use it for prediction; (5) *Average Prediction:* Take a soft label average of all source predictions and output the class with the maximum score; (6) *DEKAN* Frikha et al. (2021): A student-teacher based SFDG method which inverts source models to generate images from each domain for training the student; (7) *Single-source-free unsupervised domain adaptation (SF-UDA) Ensembles:* We extend multiple SOTA SF-UDA methods (SHOT (Liang et al., 2021), NRC (Yang et al., 2021), SHOT++ (Liang et al., 2021), BAIT (Yang et al., 2020)) for TTA by adapting the target batch individually to each source model and taking a soft label average; (8) *Multi-source-free domain adaptation (MSF-UDA):* We adapt SOTA MSFDA methods (DECISION Dong et al. (2021), CAiDA Ahmed et al. (2021)) to the TTA setting, i.e. we train the model using the respective losses on the target batch, and output the prediction; and (9) *Best Source:* An oracle baseline where we look at the best individual source for each target batch and use its predictions. Additionally, we evaluate on these methods with and without their key modules for completeness. Further implementation details are provided in Appendix B.2

Table 1: Average leave-one-domain-out accuracies on PACS (Li et al., 2017), OfficeHome (Venkateswara et al., 2017), VLCS (Fang et al., 2013) and Terra Incognita (Beery et al., 2018) (Sec. B.1), using ResNet50. Standard deviation across three seeds is reported here. For each dataset, the best results are in bold, the second best results are underlined. Baselines explained in Section 3.1

	PACS	OfficeHome	VLCS	TerraIncognita	Average
Majority Voting	70.4 ± 0.0	64.1 ± 0.0	74.0 ± 0.0	36.0 ± 0.0	61.1
Entropy Weighting	78.1 ± 0.0	68.3 ± 0.0	77.4 ± 0.0	39.5 ± 0.0	65.8
Lowest Entropy	77.5 ± 0.0	66.5 ± 0.0	76.7 ± 0.0	38.4 ± 0.0	64.8
Random Source	75.3 ± 0.0	60.2 ± 0.0	69.6 ± 0.0	36.7 ± 0.0	60.5
Average Prediction	77.7 ± 0.0	68.5 ± 0.0	78.0 ± 0.0	**40.5 ± 0.0**	66.2
DEKAN	83.7 ± 1.8	65.1 ± 1.0	71.1 ± 3.2	35.5 ± 2.2	63.9
SHOT Ensemble	82.1 ± 3.5	68.0 ± 6.3	76.0 ± 0.9	39.2 ± 2.2	66.3
BAIT Ensemble	82.3 ± 4.3	68.2 ± 6.5	73.3 ± 0.9	37.9 ± 2.5	65.4
SHOT++ Ensemble	86.3 ± 1.1	69.3 ± 0.6	77.5 ± 2.1	40.0 ± 3.8	68.3
NRC Ensemble	82.5 ± 3.4	68.0 ± 6.3	76.0 ± 1.1	38.9 ± 1.5	66.4
CAiDA w/o Ent	80.7 ± 3.9	67.4 ± 8.4	68.3 ± 3.8	37.5 ± 5.3	63.5
CAiDA w/o Div	81.1 ± 4.2	62.2 ± 5.5	68.2 ± 3.9	38.6 ± 5.4	62.5
CAiDA w/o Cls	80.8 ± 3.9	62.1 ± 5.5	68.2 ± 4.1	38.6 ± 5.6	62.4
CAiDA w/o Crc	81.1 ± 4.3	62.0 ± 5.3	68.4 ± 4.0	38.2 ± 6.0	62.4
CAiDA	81.1 ± 4.2	67.7 ± 8.5	68.4 ± 4.0	38.2 ± 6.0	63.9
DECISION	81.5 ± 1.0	67.5 ± 6.8	72.4 ± 1.6	38.0 ± 2.6	64.9
TEPLA-Soft (Ours)	85.8 ± 0.1	69.7 ± 0.2	**78.4 ± 0.1**	39.8 ± 0.1	68.4
TEPLA (Ours)	**88.1 ± 0.3**	**70.2 ± 0.2**	78.0 ± 0.2	40.3 ± 0.2	**69.2**
Best Source (Oracle)	74.7 ± 0.0	65.1 ± 0.0	77.8 ± 0.0	47.1 ± 0.0	66.2

3.2 Results

Table 1 summarizes the results. As stated above, we compare TEPLA with a wide variety of simple baselines along with more complicated SFDG, SF-UDA and MSF-UDA methods adapted to work in the TTA setting. TEPLA has the highest average accuracy across all four datasets, and is second-best only on TerraIncognita – despite its simplicity. We outperform simple baselines by 3.0% $\sim$ 8.7%, DEKAN by 5.3%, SF-UDA methods by 0.9% $\sim$ 3.8% and MSF-UDA methods by 4.3% $\sim$ 6.8%. Notably, our method has far less variance than UDA baselines on all datasets. The high variance of the UDA methods may be explained by the fact that they often have multiple loss terms in their objective, each with their own hyperparameters, and are constrained to use only images available in the batch that has arrived for inference. We also observe that simple baselines are competitive for most datasets we consider, the notable exception being PACS. Particularly, average prediction seems to be performing on par with even the best source oracle method. We also experiment with soft pseudo-labeling (TEPLA-Soft), but empirically find hard pseudo-labeling to work better.

4 Conclusion

In this paper, we propose TEPLA, a simple, yet effective test-time adaptation method based on pseudo-labeling for solving the task of source-free domain generalization. We establish, compare and outperform a variety of baselines on four benchmark domain shift datasets, as measured by average accuracy. Our results indicate that for SFDG, although using test-time unlabeled data is beneficial, it may make more sense to employ simple methods instead of complicated multi-objective, hyper-parameter sensitive pipelines, considering that a deployed model might be constrained by latency and storage limitations. It is worth noting that in the SFDG setting, due to the absence of labeled data in both source and target domains, providing performance guarantees may be challenging, and it is possible for the model to silently degrade. Although, one can use unsupervised empirical measures such as the prediction entropy as proxy measures for accuracy, reliable unsupervised monitoring of model performance is still a difficult, open problem. Until reliable correlations between unsupervised measures and actual model performance are established, it might be prudent to have periodic manual checks in deployed models, even more so in safety-critical applications.

REFERENCES

Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10103–10112, 2021.

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

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8344–8353, 2022.

Yining Chen, Colin Wei, Ananya Kumar, and Tengyu Ma. Self-training avoids using spurious features under domain shift. *Advances in Neural Information Processing Systems*, 33:21061–21071, 2020.

Myung Jin Choi, Joseph J Lim, Antonio Torralba, and Alan S Willsky. Exploiting hierarchical context on a large database of object categories. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 129–136. IEEE, 2010.

Dengxin Dai and Luc Van Gool. Dark Model Adaptation: Semantic Image Segmentation from Daytime to Nighttime. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2018-November:3819–3824, 2018. doi: 10.1109/ITSC.2018.8569387.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jiahua Dong, Zhen Fang, Anjin Liu, Gan Sun, and Tongliang Liu. Confident anchor-induced multi-source free domain adaptation. *Advances in Neural Information Processing Systems*, 34:2848–2860, 2021.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–308, 2009.

Chen Fang, Ye Xu, and Daniel 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*, pp. 1657–1664, 2013.

Yuqi Fang, Pew-Thian Yap, Weili Lin, Hongtu Zhu, and Mingxia Liu. Source-free unsupervised domain adaptation: A survey. *arXiv preprint arXiv:2301.00265*, 2022.

Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

Ahmed Frikha, Haokun Chen, Denis Krompaß, Thomas Runkler, and Volker Tresp. Towards data-free domain generalization. *arXiv preprint arXiv:2110.04545*, 2021.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 597–613. Springer, 2016.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *ArXiv*, abs/1706.02677, 2017.

Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and Zico Kolter. Test-time adaptation via conjugate pseudo-labels. *arXiv preprint arXiv:2207.09640*, 2022.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

Da Li, Yongxin Yang, Yi Zhe Song, and Timothy M. Hospedales. Deeper, Broader and Artier Domain Generalization. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:5543–5551, 2017. ISSN 15505499. doi: 10.1109/ICCV.2017.591.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 6028–6039. PMLR, 2020.

Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8602–8617, 2021.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18. PMLR, 2013.

Vaasudev Narayanan, Aniket Anand Deshmukh, Urun Dogan, and Vineeth N Balasubramanian. On challenges in unsupervised domain generalization. In *NeurIPS 2021 Workshop on Pre-registration in Machine Learning*, pp. 42–58. PMLR, 2022.

Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. *arXiv preprint arXiv:2106.15326*, 2021.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. corr abs/1512.00567 (2015), 2015.

Antonio Torralba, Bryan C Russell, and Jenny Yuen. Labelme: Online image annotation and applications. *Proceedings of the IEEE*, 98(8):1467–1484, 2010.

Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and C. V. Jawahar. IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*, pp. 1743–1751, 2019. doi: 10.1109/WACV.2019.00190.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027, 2017.

G. Volk, S. Müller, A. v. Bernuth, D. Hospach, and O. Bringmann. Towards robust cnn-based object detection through augmentation with synthetic rain variations. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 285–292, 2019. doi: 10.1109/ITSC.2019.8917269.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Qian Wang and Toby Breckon. Unsupervised domain adaptation via structured prediction based selective pseudo-labeling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6243–6250, 2020.

Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. Unsupervised domain adaptation without source data by casting a bait. *arXiv preprint arXiv:2010.12427*, 1(2): 5, 2020.

Shiqi Yang, Joost van de Weijer, Luis Herranz, Shangling Jui, et al. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Advances in neural information processing systems*, 34:29393–29405, 2021.

Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3801–3809, 2018.

Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5982–5991, 2019.

Setting	Source Domains		Target Domain Access	
	Source-Free	Multiple Sources	Offline	Online
Domain Generalization (DG) (Gulrajani & Lopez-Paz (2020))	×	✓	×	×
Source-Free Domain Generalization (SFDG) (Frikha et al. (2021))	✓	✓	×	×
Unsupervised Domain Adaptation (UDA) (Ben-David et al. (2010))	×	×	✓	×
Source-Free Domain Adaptation (SF-UDA) (Liang et al. (2020))	✓	×	✓	×
Multi-Source Source-Free DA (MSF-UDA) (Ahmed et al. (2021))	✓	✓	✓	×
Test-time Adaptation (TTA) (Wang et al. (2020))	✓	×	×	✓
Ours	✓	✓	×	✓

Table 2: Comparison with other related settings in literature. We propose to address a setting where we assume access only to multiple source models, and not the data used to train those models, to give an adaptive prediction during inference.

A Related Work

Domain Generalization & Unsupervised Domain Adaptation. Domain Generalization (DG) (Gulrajani & Lopez-Paz, 2020; Wang et al., 2022) methods try to learn a model using labeled data from multiple source domains in order to maximize performance on an unseen target domain. These methods generally transform the source data to a representation space where domain-invariant features, relevant for classification in the target domain, are captured. Please see (Wang et al., 2022) for a detailed survey of DG methods. On the other hand, Unsupervised Domain Adaptation (UDA) (Wilson & Cook, 2020) methods work under a relatively more relaxed setting where simultaneous access to source and target domains is assumed. Multiple classes of methods exist for UDA such as adversarial-learning (Zhang et al., 2018), pseudo-labeling (Wang & Breckon, 2020) and reconstruction-based (Ghifary et al., 2016) methods. Nonetheless, UDA methods cannot be used in scenarios where data privacy needs to be ensured. Additionally, it can be cumbersome to manage multiple large source domain datasets when compared to sharing source models alone.

Source-Free Domain Adaptation. To alleviate the UDA assumption of simultaneous access to source and target images, source-free domain adaptation methods (Fang et al., 2022) were proposed, which learn to adapt to the target domain using only the source model without access to source data, along with an offline unlabeled target dataset. These methods typically refine the pseudo-labels generated by the source model for the target domain using self-supervision (Liang et al., 2020) or regularization (Qiu et al., 2021) approaches. However, unlike our approach, beyond having offline access to target domain samples, these methods adapt only from a single source model.

Multi-Source Source-free UDA (MSFDA) (Dong et al., 2021; Ahmed et al., 2021) methods, similar to our setup, learn from multiple domain-specific models without assuming access to the labeled source data, and the model is adapted only using unlabeled images from the target domain. But, similar to SFDA, the adaptation process is done in an offline manner with knowledge of the target data. Unlike MSFDA, our method works completely during inference time on an unseen target domain, of which there is no prior knowledge.

Test-time Adaptation. As a development in recent years, test-time adaptation (TTA) methods make use of the incoming test batch to optimize the model on some unsupervised objective to give a more aligned prediction for that batch. Unlike standard supervised learning methods, which keep the model frozen once training is done, TTA methods attempt to alter the decision surface during inference based on auxiliary, unsupervised objectives. For instance, Sun et al. (2020) optimize the network to do well on rotation prediction, but require altering the training procedure in order to train the rotation head. Wang et al. (2020) modulate only the batch-normalization parameters by minimizing average prediction entropy of the target batch. Boudiaf et al. (2022) propose an objective which enforces samples in the test batch which are close in the representation space, to be assigned to the same class. Iwasawa & Matsuo (2021) propose an approach similar to the prototypical networks by interpreting the classification layer weights as templates or prototypes for each class. These prototypes are updated during test-time based on pseudo-labels generated by the model, and the nearest prototype is given as the final prediction. The majority of TTA methods in literature conform to the fully-test time adaptation setting (Wang et al., 2020) and are source-free. While closely related to our work, these methods are proposed for learning from only a single model, and not for learning from multiple domains. To the best of our knowledge, we are the first to address giving an adaptive prediction in the presence of multiple domain-specific models during inference.

B Experimental Details

B.1 Datasets

We perform our experiments on four benchmark datasets which are commonly used in the domain shift literature – PACS (Li et al., 2017), OfficeHome (Venkateswara et al., 2017), VLCS (Fang et al., 2013) and TerraIncognita (Beery et al., 2018). PACS comprises 9,991 images divided in 7 classes across 4 different domains: Photo, Art, Cartoon and Sketch. The Office-Home dataset contains 15,500 images across 4 domains: Product, Art, Clipart and Real World. Each domain is divided into 65 different classes. VLCS is a combination of four photographic datasets: Caltech101 (Fei-Fei et al., 2006), LabelMe (Torralba et al., 2010), SUN09 (Choi et al., 2010), VOC2007 (Everingham et al., 2009), containing 10729 examples of 5 classes. Following (Gulrajani & Lopez-Paz, 2020), we consider a subset of the TerraIncognita dataset which contains 24,788 images divided into 10 classes across 4 domains: L100, L38, L43 and L46.

B.2 Implementation Details

For each source model, following Liang et al. (2020), we use an ImageNet Deng et al. (2009) pre-trained ResNet-50 He et al. (2016) and train it using standard empirical risk minimization Vapnik (1991) (i.e. training using a suitable task loss function). The penultimate fully-connected layer is replaced with a bottleneck layer and a classifier with weight normalization. Batch normalization Ioffe & Szegedy (2015) is employed to normalize the outputs of the bottleneck layer. Similar to Liang et al. (2020), we employ label smoothing for source training. During inference, similar to Wang et al. (2020), we choose to update only the Batch Normalization (BN) parameters and keep the rest of the network frozen. We select between two scales (the default values, and scaled down values) of BN momentum and learning rate (LR) using a simple heuristic. If we have knowledge of the fact that source domains are visually similar (from prior knowledge or practitioner wisdom), we use a lower momentum and LR, else we use the standard values. This heuristic operates under the common assumption that domains are sampled from some unknown but fixed prior, and if all the source domains are close together, it is likely that the target domain would also be similar. We note that we do not violate the SFDG setting herein since this heuristic does not depend on target data, which we are not privy to. Default values are 0.1 for momentum and $5e^{-4}$ for LR set by linear-scaling Goyal et al. (2017), which we use for PACS and OfficeHome. For VLCS and TerraIncognita, we use a scaled-down momentum of 0.001 and a LR of $1e^{-4}$. In the hard variant of TEPLA, we perform label smoothing Szegedy et al. (2015) with default parameter 0.1 on the hard pseudo-label before updating the model. We set the test-time batch size as 32 and perform all experiments on one V100 GPU.

Algorithm 1 TEPLA for SFDG

Input: Domain-specific models $\{f^s_{\theta_i}\}_{i=1}^M$ trained on $\{D^s_i\}_{i=1}^M$, Test batch $\mathcal{B} = \{x_j\}_{j=1}^N$ from target domain D^T

```
$\theta = [\theta_1^\intercal \, \theta_2^\intercal \ldots \theta_M^\intercal]^\intercal$
for j ∈ $\{1, 2, \ldots N\}$ do
    $y_j^{pl} = \arg\max_{y \in \mathcal{Y}} \frac{1}{M} \sum_{i=1}^M f_{\theta_i}(x_j)$
end for
for i ∈ $\{1, 2, \ldots M\}$ do
    $\mathcal{L}(\mathcal{B}, \theta_i) = \frac{1}{N} \sum_{j=1}^N \mathcal{L}(y_j^{pl}, f_{\theta_i}(x_j))$
end for
$\theta' = \theta - \eta \sum_{i=1}^M \nabla \mathcal{L}(\mathcal{B}; \theta_i)$
for j ∈ $\{1, 2, \ldots N\}$ do
    $y_j = \arg\max_{y \in \mathcal{Y}} \frac{1}{M} \sum_{i=1}^M f_{\theta'_i}(x_j)$
end for
return $y$
```

	P	A	C	S	Avg.
Majority Voting	95.2	70.6	56.8	58.8	70.4
Entropy Weighting	97.9	80.7	66.4	67.2	78.1
Lowest Entropy	97.9	80.2	67.0	65.0	77.5
Random Source	97.0	77.2	63.4	63.7	75.3
Average Prediction	98.0	80.3	65.7	66.9	77.7
DEKAN	96.1	83.4	76.1	79.2	83.7
SHOT-Ensemble	97.3	88.2	73.3	69.5	82.1
BAIT-Ensemble	96.5	86.9	74.7	71.1	82.3
SHOT++ Ensemble	**98.9**	88.9	79.5	**77.8**	86.3
NRC Ensemble	97.4	88.4	74.0	70.4	82.5
DECISION	97.9	88.5	66.7	72.9	81.5
CAiDA	97.1	86.4	73.3	67.8	81.1
Tent Ensemble	97.7	87.3	82.1	73.5	85.2
LAME Ensemble	91.3	62.3	45.1	50.4	62.3
TEPLA-Soft	98.6	87.5	82.8	74.1	85.8
TEPLA	98.7	**90.2**	**86.4**	77.0	**88.1**
Best Source	98.0	73.4	63.7	63.7	74.7

Table 3: Domain-wise results for Table 1 on PACS

	P	A	C	R	Avg.
Majority Voting	73.5	63.0	44.6	75.2	64.1
Entropy Weighting	77.4	68.0	47.9	79.9	68.3
Lowest Entropy	75.8	66.3	45.9	78.1	66.5
Random Source	68.8	51.0	45.8	75.0	60.2
Average Prediction	77.8	68.1	48.0	79.9	68.5
DEKAN	73.7	64.2	46.9	75.4	65.1
SHOT-Ensemble	78.4	66.3	48.4	78.7	68.0
BAIT-Ensemble	78.3	66.5	48.0	79.8	68.2
SHOT++ Ensemble	76.6	65.3	**53.2**	**82.2**	69.3
NRC Ensemble	77.6	66.5	47.4	80.4	68.0
DECISION	**80.2**	66.7	44.8	78.1	67.5
CAiDA	78.0	66.3	48.1	78.5	67.7
Tent Ensemble	75.4	69.2	49.1	79	68.2
LAME Ensemble	75.2	65.7	42.3	77.5	65.2
TEPLA-Soft	76.9	70.7	50.7	80.5	69.7
TEPLA	77.3	**71.4**	51.6	80.6	**70.2**
Best Source	76.9	66.5	43.5	73.6	65.1

Table 4: Domain-wise results for Table 1 on OfficeHome

	C	L	S	V	Avg.
Majority Voting	94.4	66.0	64.4	71.3	74.0
Entropy Weighting	96.8	65.6	72.3	75.0	77.4
Lowest Entropy	96.0	65.0	71.8	73.8	76.7
Random Source	69.8	68.8	81.3	58.3	69.6
Average Prediction	97.4	66.0	71.9	76.8	78.0
DEKAN	87.4	61.2	63.4	72.4	71.1
SHOT-Ensemble	93.6	63.9	66.9	79.6	76.0
BAIT-Ensemble	87.8	61.9	64.3	79.2	73.3
SHOT++ Ensemble	96.8	65.0	**73.1**	75.1	77.5
NRC Ensemble	94.8	63.4	66.0	**79.9**	76.0
DECISION	95.8	60.4	61.5	71.9	72.4
CAiDA	78.4	61.9	61.2	72.0	68.4
Tent Ensemble	82.4	63.3	63.1	70.3	69.8
LAME Ensemble	91.3	51.7	63.3	61.4	66.9
TEPLA-Soft	**97.4**	**68.6**	71.5	76.3	**78.4**
TEPLA	97.3	68.3	71.1	75.2	78.0
Best Source	98.6	65.9	76.6	70.1	77.8

Table 5: Domain-wise results for Table 1 on VLCS

	L100	L38	L43	L46	Avg.
Majority Voting	48.0	31.0	33.6	31.2	36.0
Entropy Weighting	55.4	32.0	38.4	32.3	39.5
Lowest Entropy	53.2	29.9	37.9	32.5	38.4
Random Source	56.3	27.1	22.9	40.6	36.7
Average Prediction	56.0	33.8	39.1	32.9	**40.5**
DEKAN	41.9	34.3	32.1	33.7	35.5
SHOT-Ensemble	42.5	35.6	38.8	**39.9**	39.2
BAIT-Ensemble	41.9	33.9	36.1	39.8	37.9
SHOT++ Ensemble	46.6	35.5	**43.7**	34.3	40.0
NRC Ensemble	42.3	**36.8**	37.0	39.4	38.9
DECISION	41.7	33.3	39.6	37.5	38.0
CAiDA	40.8	35.0	41.2	35.6	38.2
Tent Ensemble	48.4	28.0	35.5	37.8	37.4
LAME Ensemble	59.6	36.9	30.5	31.9	39.7
TEPLA-Soft	59.6	29.3	38.8	31.6	39.8
TEPLA	**60.7**	27.1	40.2	33.0	40.3
Best Source	60.3	33.5	51.3	43.2	47.1

Table 6: Domain-wise results for Table 1 on TerraIncognita

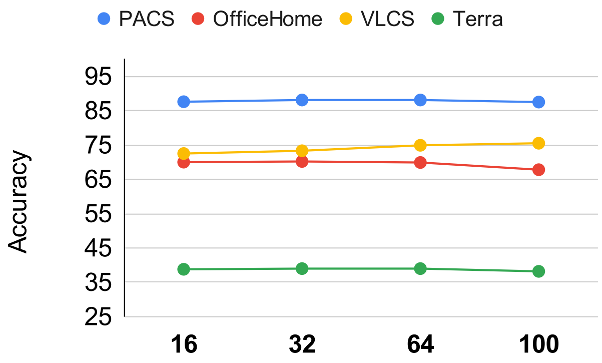


Figure 2: Robustness of TEPLA to batch size. Since the number of samples that arrive for inference cannot be known beforehand, it is beneficial if the TTA algorithm is stable across a range of batch sizes. For all datasets considered, we see that TEPLA maintains performance across different batch sizes.

C Complete Domain-Wise Results for Table 2

Tables 3, 4, 5 and 6 present the complete domain-wise results for Table 1. Please refer to the main paper for details on the experiment. We observe that for most domains across all datasets, TEPLA outperforms all competing baselines. Although the hard variant of TEPLA achieved the highest accuracy on average, the soft version (TEPLA-Soft) is performing competitively on most domains. Additionally, we observe that SHOT++ Ensemble seems to be doing well on certain domains, possibly due to its confidence-based semi-supervision module which provides better source-target pair calibrations before its aggregation step. Specifically, for PACS (Table 3), it is marginally better than TEPLA for Photo (0.2%) and Sketch (0.8%), but is much worse on the Cartoon domain (6.9%). We make similar observations for VLCS (Table 5) as well, where it is the best on SUN09 but is worse than TEPLA-Soft on Caltech101 and LabelMe.

D Additional Experiments

D.1 Effect of Batch Size

Figure 2 shows the performance of TEPLA across different batch sizes. Since in the TTA setting the number of samples that arrive for inference cannot be known beforehand, it is important for the method to be stable across a range of batch sizes. For all datasets considered, we can see that TEPLA maintains performance across different batch sizes.

D.2 Batch Index vs Accuracy

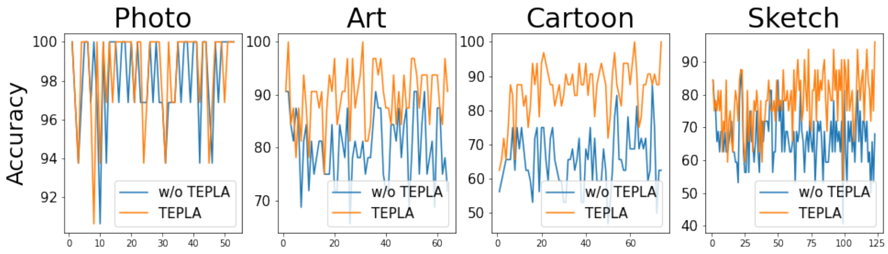
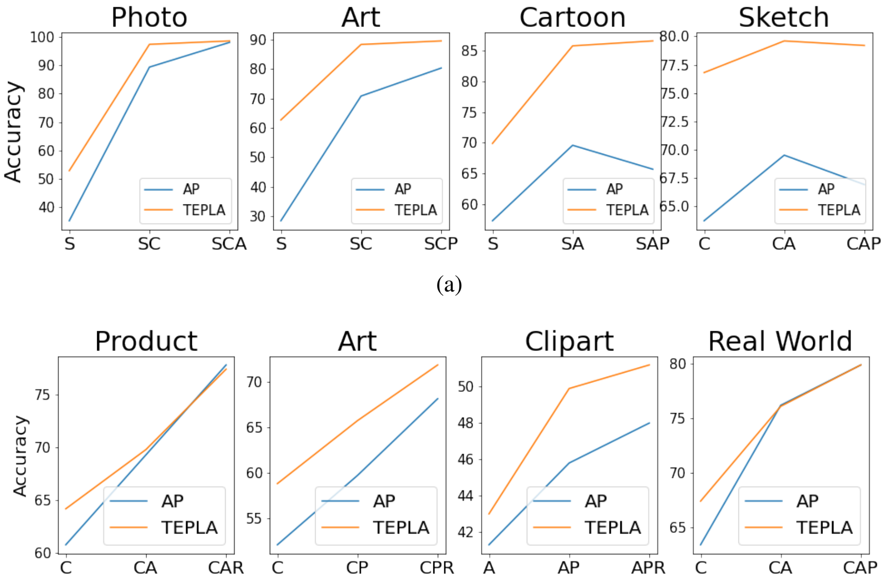


Figure 3: Accuracy across test batches: note that TEPLA outperforms the baseline consistently across the batches and domains



(a)

(b)

Figure 4: TEPLA evaluated in the domain incremental setting, compared with Average Prediction (AP). Each subplot title refers to that particular domain being treated as the target domain. The labels on the horizontal axis in each subplot refers to the source models available at that time step. For instance, in the first subplot, ”Photo” is the target domain, while Sketch (S), Cartoon (C) and Art (A) are source models which are available incrementally. Exact numbers are available in Table 7

PACS	Photo			Art			Cartoon			Sketch		
	S	SC	SCA	S	SC	SCP	S	SA	SAP	C	CA	CAP
Average Prediction	35.3	89.3	98	28.5	70.8	80.3	57.3	69.6	65.7	63.7	69.5	66.9
TEPLA (Ours)	52.9	97.3	98.5	62.7	88.3	89.5	69.9	85.8	86.6	76.8	79.6	79.2
OfficeHome	Product			Art			Clipart			Real World		
	C	CA	CAR	C	CP	CPR	A	AP	APR	C	CA	CAP
Average Prediction	60.8	69.3	77.8	52.1	59.7	68.1	41.3	45.8	48	63.4	76.2	79.9
TEPLA (Ours)	64.2	69.8	77.4	58.8	65.7	71.8	43.0	49.9	51.2	67.4	76.1	79.9

Table 7: As new source domain models are added incrementally, TEPLA consistently improves performance on the target domain.

We plot the accuracy for each target batch in Figure 3 for PACS. We can clearly see the benefits of performing TEPLA adaptation to the target batch, specifically for the harder Art, Cartoon and Sketch domains. In addition, we note that the quantitative results are not influenced by improvements in sporadic batch indices, but the improvement is consistent across the batches at test time.

D.3 Domain-Incremental Setting

In the real world, due to various reasons, it is possible that source models might arrive in an incremental manner, and not be available all at once. In Figure 4, we thus study how TEPLA fares in a domain incremental setup for PACS and OfficeHome. We use the leave-one-domain-out evaluation strategy, where we loop through each domain in the dataset as the target. At each time step, we add one model to the source domain set and adapt using TEPLA. For PACS, we fix the ordering as S → C → A → P and C → A → P → R for OfficeHome. We find that even when limited source models are available, TEPLA performs better than the average prediction (AP) baseline. Additionally, we

PACS	P	A	C	S	Avg.
Episodic	98.5	83.4	71.1	71.1	81.1
Non-Episodic	**98.7**	**90.2**	**86.4**	**77.0**	**88.1**
OfficeHome	P	A	C	R	Avg.
Episodic	77.2	69.2	48.9	80.3	68.9
Non-Episodic	**77.3**	**71.4**	**51.6**	**80.6**	**70.2**
VLCS	C	L	S	V	Avg.
Episodic	97.2	67.0	**71.3**	**76.4**	**78.0**
Non-Episodic	**97.3**	**68.3**	71.1	75.2	**78.0**
TerraIncognita	L100	L38	L43	L46	Avg.
Episodic	56.1	**33.3**	38.7	**33.7**	**40.4**
Non-Episodic	**60.7**	27.1	**40.2**	33.0	40.3

Table 8: Comparing Episodic vs Non-Episodic evaluation

observe that performance on the target increases as more source models are introduced, showing the usefulness of TEPLA in a domain incremental setting.

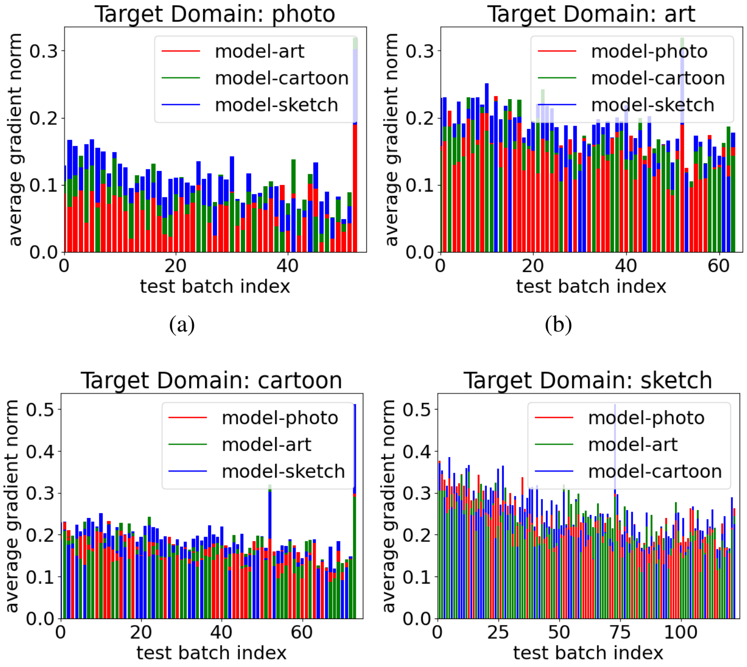
D.4 Evaluation Schemes: Episodic vs Non-episodic

In Table 8, we analyze the effect of episodic vs non-episodic evaluation. Episodic evaluation implies that we adapt to every batch using the original source model weights, i.e. ones received after pre-training. Non-episodic evaluation implies that we never reset to the original source weights and continuously adapt to the incoming target batches. We find that across all datasets, non-episodic evaluation, i.e. continuously adapting to the target batch is always beneficial. This can be explained by the fact that in the SFDG setting, since incoming batches are assumed to be coming from the same target distribution, adapting the source models continuously seems to be a helpful strategy.

We also experiment with stochastic weight restore strategy (see Table 9) where at each batch we make a decision to whether to reset to the original source models. We observe a marginal drop in performance across datasets using the stochastic strategy, apart from TerraIncognita, where we see an incremental gain. At the same time, adapting even with a small restoration probability seems to be better than evaluating episodically, indicating the success of the continuous adaptation strategy for SFDG. We note that the all results in the main paper use non-episodic evaluation.

D.5 Gradient Analysis

We perform a gradient analysis for the PACS dataset to visualize the cumulative gradient for each source domain. The average gradient of a source model is computed by taking the mean of 2-norm of the flattened gradient of every parameter tensor being updated. Figure 5 shows the average gradient across target batches. We observe that across batches, the gradient for the Art source model is the lowest for target images from Photo. We hypothesize that since images from the Photo and Art domain are visually similar (compared to say, Sketch), the Art source is already a good predictor for Photo, hence requiring a smaller gradient update. Similarly, since the Sketch domain is visually disparate from other domains, the Sketch model transfers poorly to all target domains, which is reflected in the figure, where the Sketch source model often has the largest average gradient across different domains. Additionally, we also observe that across all domains, apart from a few anomalous batches, as more batches come in, the average gradient is *decreasing*. This is possibly an indication that source models are getting better calibrated towards making prediction on the target domain.



(a)

(b)

(c)

(d)

Figure 5: Average gradient norm of different source models for a given target domain.

PACS	P	A	C	S	Avg.
Prob. = 1 (Episodic)	98.5	83.4	71.1	71.1	81.1
Prob. = 0.75	98.6	83.7	72.3	71.3	81.5
Prob. = 0.5	98.6	83.8	73.5	71.5	81.9
Prob. = 0.25	98.6	84.9	76.1	71.5	82.8
Prob. = 0 (Non-Episodic)	98.7	90.2	86.4	77.0	88.1
OfficeHome	P	A	C	R	Avg.
Prob. = 1 (Episodic)	77.2	69.2	48.9	80.3	68.9
Prob. = 0.75	77.2	69.4	49.1	80.3	69.0
Prob. = 0.5	77.4	69.4	49.2	80.4	69.1
Prob. = 0.25	77.4	69.7	49.5	80.4	69.2
Prob. = 0 (Non-Episodic)	77.3	71.4	51.6	80.6	70.2
VLCS	C	L	S	V	Avg.
Prob. = 1 (Episodic)	97.2	67.0	71.3	76.4	78.0
Prob. = 0.75	97.3	66.2	71.7	76.8	78.0
Prob. = 0.5	97.4	66.2	71.7	76.8	78.0
Prob. = 0.25	97.5	66.3	71.8	76.8	78.1
Prob. = 0 (Non-Episodic)	97.3	68.3	71.1	75.2	78.0
TerraIncognita	L100	L38	L43	L46	Avg.
Prob. = 1 (Episodic)	56.1	33.3	38.7	33.7	40.4
Prob. = 0.75	56.4	33.8	39.4	33.0	40.6
Prob. = 0.5	56.6	33.8	39.5	33.0	40.7
Prob. = 0.25	57.4	33.9	39.7	33.0	41.0
Prob. = 0 (Non-Episodic)	60.7	27.1	40.2	33.0	40.3

Table 9: Results using stochastic source model restoration. Restoration probability of 0 implies source models are never restored, i.e. the non-episodic setting, whereas probability of 1 implies source models are restored for every batch before adaptation.

	PACS	OfficeHome	VLCS	TerraIncognita	Average
ERM	85.5 ± 0.6	66.5 ± 0.4	77.5 ± 0.8	46.1 ± 2.9	68.9
IRM	83.5 ± 1.0	64.3 ± 2.3	78.6 ± 0.6	47.6 ± 1.5	68.5
GroupDRO	84.4 ± 1.0	66.0 ± 0.8	76.7 ± 0.7	43.2 ± 1.5	67.6
Mixup	84.6 ± 0.8	68.1 ± 0.5	77.4 ± 0.7	47.9 ± 1.4	69.5
MLDG	84.9 ± 1.1	66.8 ± 0.8	77.2 ± 0.8	47.8 ± 1.7	69.2
CORAL	86.2 ± 0.6	68.7 ± 0.4	78.8 ± 0.7	47.7 ± 1.8	70.4
MMD	84.7 ± 0.8	66.4 ± 0.3	77.5 ± 1.2	42.2 ± 1.9	67.7
DANN	83.7 ± 1.1	65.9 ± 0.7	78.6 ± 0.7	46.7 ± 1.6	68.7
CDANN	82.6 ± 0.9	65.7 ± 1.4	77.5 ± 1.0	45.8 ± 2.7	67.9
MTL	84.6 ± 1.0	66.4 ± 0.5	77.2 ± 0.8	45.6 ± 2.4	68.5
SagNet	86.3 ± 0.5	68.1 ± 0.3	77.8 ± 0.7	48.6 ± 1.8	70.2
ARM	85.1 ± 0.7	64.8 ± 0.4	77.6 ± 0.7	45.5 ± 1.3	68.3
VREx	84.9 ± 1.1	66.4 ± 0.6	78.3 ± 0.9	46.4 ± 2.4	69.0
RSC	85.2 ± 1.0	65.5 ± 1.0	77.1 ± 0.7	46.6 ± 1.4	68.6
TEPLA-Soft	83.8 ± 0.8	63.7 ± 0.1	71.9 ± 0.2	37.3 ± 0.2	64.2
TEPLA	85.1 ± 0.3	63.9 ± 0.2	71.5 ± 0.2	37.7 ± 0.2	64.6

Table 10: Comparison with Domain Generalization methods with ResNet50 without label smoothing. Results from ERM to RSC as in Gulrajani & Lopez-Paz (2020).

E Comparison with DG methods

Table 10 compares the performance of our method with domain generalization (DG) methods, under the same training conditions, i.e. ResNet50 architecture trained without label smoothing. We note that in contrast to SFDG, DG works under a slightly relaxed assumption of simultaneous access to *data* from multiple source domains. DG methods exploit the availability of labeled data to learn domain-invariant models, which is not possible in the SFDG setting. We observe that for PACS, even under the stricter SFDG setting, TEPLA is competitive with DG methods. On the other hand, for OfficeHome, VLCS and TerraIncognita, we see a gap with DG methods, noting that there is significant room for improvement for SFDG methods.