Randomly projecting out distribution shifts for improved robustness

Anonymous Author(s)
Affiliation
Address
email

Abstract

Real-world applications of machine learning require a model to be capable of dealing with domain shifts that might occur at test time due to natural perturbations to the data distribution induced by, for example, changes in the data collection conditions, or synthetic distortions such as adversarial attacks. While a learning system might be simultaneously vulnerable to natural and hand-engineered perturbations, previous work has mainly focused on developing techniques to alleviate the effects of specific types of distribution shifts. In this work, we propose a unified and versatile approach to mitigate both natural and artificial domain shifts via the use of random projections. We show that such projections, implemented as convolutional layers with random weights placed at the input of a model, are capable of increasing the overlap between the different distributions that may appear at training/testing time. We evaluate the proposed approach on settings where different types of distribution shifts occur, and show it provides gains in terms of improved out-of-distribution generalization in the domain generalization setting, as well as increased robustness to two types of adversarial perturbations on the CIFAR-10 dataset without requiring adversarial training.

1 Introduction

Different forms of distribution shifts often affect model's prediction performance in machine learning applications. In recent years, new techniques have emerged to allow learning under naturally-occurring data variations, in settings such as domain adaption and domain generalization [1, 2, 3, 4, 5]. Simultaneously, the vulnerability of neural networks to hand-crafted perturbations has also drawn attention due to the threat it poses to safety-critical applications. Thus, a myriad of techniques tailored to improve the robustness against artificially generated out-of-distribution examples have been proposed [6]. Although previous work has proposed to leverage advances in domain adaptation approaches to improve adversarial robustness [7, 8] and to mitigate the effect of distribution shifts by performing some type of adversarial training [9, 10], only few contributions [11] attempted to devise strategies capable of dealing with both types of distribution shifts.

In this work, we propose an efficient and unified framework to deal with both natural and artificial domain changes: *Randomly Projecting Out Distribution Shifts* (RPODS). Motivated by the earlier success of random projections for applications such as generative modeling [12, 13], data augmentation [14, 15], among others [16, 17, 18], we employ random data transformations as a means for distribution matching, i.e., we map input samples to a space where the overlap between domains is likely to be higher, thus decreasing the amount of available domain-specific information. As an extra practical contribution, the proposed approach further increases the robustness of a model to white-box adversarial attacks. That is, random projection layers are re-sampled prior to every prediction. As such, a subset of the model's parameters is always unknown to the attacker. Doing so limits the action

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

of attacks that rely on previous knowledge about the model, while not harming the original accuracy, unlike methods that include adversarial examples at training time. Our contributions are summarized in the following:

1. We propose RPODS, a principled and versatile approach to improve neural networks robustness to natural and artificial distribution shifts via random projections;
2. We empirically confirm that random projections filter away domain-specific information by estimating the divergence between pairs of domains before and after projections;
3. We challenge the versatility of RPODS by performing experiments in two settings where distributions shifts are present, namely domain generalization and adversarial robustness, and show that RPODS outperform approaches tailored to tackle either one of the settings.

2 Using random projections to mitigate distribution shifts

In this section, we motivate the use of random projections to handle distribution shifts, and introduce the proposed approach that leverages such result on neural networks.

Let the input space be represented by $\mathcal{X} \subset \mathbb{R}^d$, while $\mathcal{Y}$ denotes the label space. In this case, examples correspond to pairs $(x, y) : x \in \mathcal{X}, y \in \mathcal{Y}$. Let $\mathcal{D}$ denote a distribution over $\mathcal{X} \times \mathcal{Y}$ and be referred to a as *domain*. We consider settings where a predictor $h : \mathcal{X} \to \mathcal{Y}$ must present a good generalization performance (in terms of expected risk) on different domains, including those not available at training time. In particular, we tackle the domain generalization setting [19, 20]. We are concerned with cases where a set with N domains is available, and their marginal distributions $\mathcal{D}^i(x), i \in \{1, ..., N\}$, differ, while data-conditional label distributions $\mathcal{D}(y|x)$ remain unchanged (i.e. the standard covariate shift scenario [2, 21, 22])[1]. Consider the input space $\mathcal{X}$ is the d-dimensional ball B^d of radius 1 centered at 0 and define the support of a domain $\mathcal{D}$, $supp(\mathcal{D}) \subset B^d$, as the set where the corresponding density is greater than some small threshold. The following result shows that the support of a projected domain occupies a higher fraction of the projected input space volume.

Theorem 1. *(Neyshabur et al. [12]) Assume $\mathcal{D}(x) = \sum_j \tau_j \mathcal{N}(x|\mu_j, \Sigma_j)$ is a mixture of Gaussians, such that there is no overlap between the supports or the projections of the components. If $supp(\mathcal{D}) \subset B^d$ and $Vol(supp(\mathcal{D})) > 0$, then, with high probability:*

$$\text{Vol}\left(\text{supp}\left(\mathcal{D}_W\right)\right) / \text{Vol}\left(\mathcal{X}_W\right) > \text{Vol}\left(\text{supp}\left(\mathcal{D}\right)\right) / \text{Vol}\left(\mathcal{X}\right), \tag{1}$$

where $\mathcal{D}_W$ represents the marginal of $\mathcal{D}$ along a random projection W and $\mathcal{X}_W$ denotes the projection of the input space. Theorem 1 shows that random projections increase the overlap between the supports of distributions over the input space and thus can reduce covariate shifts. More specifically, in case two domains over $\mathcal{X}$, $\mathcal{D}^1$ and $\mathcal{D}^2$, are considered, the projection W acts in such a way that it likely increases the overlap between both domains. In the next section, we empirically confirm this observation by showing that the $\mathcal{A}$-distance [23], a proxy for the $\mathcal{H}$-divergence [24], that accounts for mismatches between distributions over the input space, is indeed decreased when estimated over projected inputs, i.e., $d_{\mathcal{A}}(\mathcal{D}^1, \mathcal{D}^2) > d_{\mathcal{A}}(\mathcal{D}^1_W, \mathcal{D}^2_W)$.

Random projections as convolutional layers. In practice, we consider applications of neural networks and implement our approach to Randomly Project Out Distribution Shifts (RPODS) using convolutional layers. More specifically, we consider a bank of K projections such that $\mathcal{X}_W \subset \mathbb{R}^m$ and each random projection matrix $W_k \in R^{d \times m}$, $k = \{1, \ldots, K\}$, has entries drawn from a Gaussian distribution. In all of our experiments, we considered $\mathcal{N}(0, \sigma^2)$, where σ is set as per the scheme introduced in [25]. In order to prevent the projections to be drastically distorted, we project the parameters of the convolutional layers to the L2 unitary ball. A model is then trained considering examples in the projected input space $\mathcal{X}_W$. Figure 1 illustrates the use of RPODS and shows examples from the PACS [26] dataset in the projected space.

Re-initializing the projections for improved robustness to white-box attacks. We further highlight that RPODS induce a further benefit in terms of improving a models' robustness to attacks that rely on knowledge about the model parameters (i.e. the white-box access model). By re-sampling the projection matrices at every iteration, in addition to having an input where distribution shifts are reduced, a part of the model parameters is constantly changing and, therefore, the attacker will never have access to the complete model when generating adversaries.

[1] We assume that adversarial perturbations induce a shift on the marginal distribution of the original data.

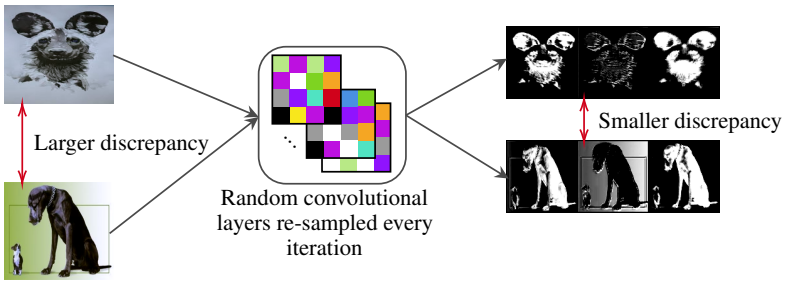

Figure 1: Illustration of the proposed approach. Input images correspond to examples from the "Photo" and "Art Painting" domains from the PACS dataset and were projected via random convolutions to a space where discrepancies between domains are reduced.

3 Experiments

In this section, we empirically show that, as stated by Theorem 1, the use of RPODS in fact helps to reduce distribution shifts, and evaluate the capability of RPODS to improve robustness to artificial and natural shifts in practical scenarios. In the case of natural domain shifts, we consider the domain generalization setting under a *leave-one-domain-out* scheme. We thus train a model with RPODS via empirical risk minimization with examples drawn from training distributions while evaluating it on an unseen domain. Finally, we show that RPODS are also able to improve robustness to adversarial attacks. For that, we use the CIFAR-10 dataset and evaluate on common adversarial perturbations. In all cases, we compare RPODS with methods tailored to deal with either natural or artificial shifts.

Random projections decrease domain divergences. We consider the PACS dataset and a ResNet-18 [27] as backbone architecture. To evaluate whether randomly projecting distributions can in fact help to reduce distribution shifts, we estimate the $\mathcal{A}$-distance for each pair of domains within the PACS dataset, and compare the values obtained with raw inputs versus projected inputs using RPODS. To do so, we train a ResNet-18 to predict domain labels, and use the error rate on the test set to estimate the distances (refer to the Appendix A.2 for experimental details). Figures 2 and 3 show the $\mathcal{A}$-distance values for all pairs of domains for a model with and without RPODS, respectively. Each entry in the matrix depicted in the figure represents a value of distance computed considering a pair of domains, which are indicated by their initials (i.e. the "Sketch" domain is denoted by "S"). Further visualizations for this experiment can be found in Appendix A.1.1.

We remark that, for all pairs of domains within the PACS dataset, the use of RPODS consistently decreases divergences between domains in comparison to raw data. As anticipated in Theorem 1, this result provides further evidence that random transformations yield a decrease in domain discrepancies. Therefore, RPODS are suitable for domain generalization/adaptation settings, since previous work [24, 21, 2] showed that encodings resulting in smaller $\mathcal{A}$-distance between domains favour out-of-distribution generalization. Based on the examples of projected inputs provided in Figure 1, we argue that RPODS act by removing domain-specific information such as texture, and thus enforces a model to focus on higher-level features such as shape, which are more uniform across different domains.

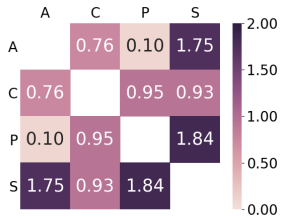

Figure 2: $\mathcal{A}$-distance for pairs of domains on PACS estimated by a ResNet-18 **with RPODS**.

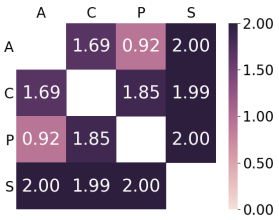

Figure 3: $\mathcal{A}$-distance for pairs of domains on PACS estimated by a ResNet-18.

Domain generalization. To evaluate RPODS on the domain generalization setting, we once more consider the PACS dataset and a ResNet-18 architecture. Results in Figures 2 and 3 show that RPODS reduce mismatches between data marginal distributions. Now, we are interested in verifying whether

the projected input space preserves enough task-related information so that a model trained with RPODS is still capable of predicting class labels. For that, we compare the out-of-domain accuracy achieved by a model with RPODS to several approaches tailored to the domain generalization setting, as well as a model trained via standard empirical risk minimization. Following recent work [28], we consider a ResNet-18 *trained from scratch* in order to favour a fair comparison with previous approaches, i.e. the impact of pre-training on final performances is ruled out.

In Table 1, we report the out-of-domain performances of models trained with RPODS on the source domains (e.g., results under column "Photo" correspond to models trained on "Art"+"Cartoon"+"Sketch"). We report the average performance across three independent training runs of the model when it presented its best in-domain accuracy (c.f. model selection protocol called *training domain validation set* in [29]). Baselines correspond to standard classifiers (denoted ERM) as well as recent methods specifically designed to tackle the domain generalization setting: self-challenge (SC) [30], Group DRO [31], GNN-Tag [32], and MLDG [33]. Further experimental details and results, including confidence intervals and other model selection criteria are presented in Appendix A.1.2. Results show that RPODS exceed the performance of the majority of the considered baselines in all domains and present the highest average accuracy on PACS, showing that performing ERM on top of random projected input spaces improves out-of-distribution generalization.

Table 1: Domain generalization results on PACS considering a leave-one-domain-out training scheme using the accuracy on the validation set of the training domains as model selection criterion. The * indicates results reported in [28].

	Photo	Art	Cartoon	Sketch
SC*	55.02	42.38	53.28	37.15
GroupDRO*	51.20	32.20	37.30	35.70
MLDG*	47.30	29.30	40.30	28.80
GNN-Tag*	53.23	33.26	49.16	54.15
ERM*	14.07	11.31	15.72	20.69
RPODS	63.90	42.63	51.74	56.67

Table 2: Adversarial robustness evaluation in terms of accuracy (%) considering PGD and FGSM attackers under a L_∞ budget of $\frac{8}{255}$ for the CIFAR-10 dataset. The number of steps employed for each attack is represented in subscript.

	Clean	PGD_7	PGD_{20}	$FGSM_1$
AT	87.14	55.63	49.79	45.72
ALP	89.79	60.29	51.89	48.50
TLA	86.21	58.88	53.87	51.59
TRADES	84.92	-	56.61	56.43
ERM	95.01	0.00	0.00	13.35
RPODS	89.70	75.62	46.35	47.49

Adversarial robustness. Lastly, we evaluate the performance of RPODS against white-box adversarial perturbations. For that, we train a wide-ResNet [34] with RPODS on the CIFAR-10 dataset for 600 epochs, and report the robust accuracy of the model with best validation performance. We consider FGSM [35] and PGD [34] attacks under L_∞ budgets. Importantly, we compare RPODS performance with approaches that *have access to adversarial examples at training time*, namely: adversarial training (AT) [34], adversarial logit pairing (ALP) [36], triplet loss adversarial training (TLA) [37], and TRADES [38]. Results in Table 2 show that RPODS achieve better accuracy on clean samples than most of the baselines and competitive robust accuracy for both attacks. Moreover, when compared with the undefended model (ERM), we observe that RPODS greatly improve the robust accuracy despite the fact that no adversarial training is performed in this case.

4 Conclusions

We introduced RPODS – a simple and efficient approach to mitigate the effects of distribution shifts on neural networks performance. In practice, RPODS project the input space via a bank of random projections, implemented as a convolutional layer added to the input of a model, with weights re-sampled at every iteration. We show that RPODS improve out-of-distribution generalization in scenarios where distribution shifts stem from different sources. More specifically, experiments on the PACS dataset showed that RPODS improve upon a number of approaches tailored to the domain generalization setting, improving the average accuracy on unseen domains by almost 6.8% with respect to the best performing baseline. We also evaluated RPODS in a setting where domain shifts were given by adversarial perturbations and showed that, despite its simplicity, RPODS greatly improved robustness to white-box attacks on the CIFAR-10 dataset in comparison to the undefended model. Notably, models employing RPODS are competitive when compared to adversarial training approaches, specifically designed to attenuate the effects of adversarial perturbations. Future work includes exploring the use of RPODS in situations where robustness to natural and adversarial distribution shifts is simultaneously required, as well as other out-of-distribution generalization settings such as single-source domain generalization and domain adaptation.

References

[1] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *arXiv preprint arXiv:2103.02503*, 2021.

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

[3] J. Wang, C. Lan, C. Liu, Y. Ouyang, W. Zeng, and T. Qin, "Generalizing to unseen domains: A survey on domain generalization," *arXiv preprint arXiv:2103.03097*, 2021.

[4] I. Albuquerque, N. Naik, J. Li, N. Keskar, and R. Socher, "Improving out-of-distribution generalization via multi-task self-supervised pretraining," *arXiv preprint arXiv:2003.13525*, 2020.

[5] J. Monteiro, X. Gibert, J. Feng, V. Dumoulin, and D.-S. Lee, "Domain conditional predictors for domain adaptation," in *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, ser. Proceedings of Machine Learning Research, L. Bertinetto, J. F. Henriques, S. Albanie, M. Paganini, and G. Varol, Eds., vol. 148. PMLR, 11 Dec 2021, pp. 193–220. [Online]. Available: https://proceedings.mlr.press/v148/monteiro21a.html

[6] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "A survey on adversarial attacks and defences," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 1, pp. 25–45, 2021.

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

[8] K. Han, B. Xia, and Y. Li, "Adversarial domain adaptation to defense with adversarial perturbation removal," *Pattern Recognition*, p. 108303, 2021.

[9] L. Zhao, T. Liu, X. Peng, and D. Metaxas, "Maximum-entropy adversarial data augmentation for improved generalization and robustness," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[10] R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," *arXiv preprint arXiv:1805.12018*, 2018.

[11] M. Awais, F. Zhou, H. Xu, L. Hong, P. Luo, S.-H. Bae, and Z. Li, "Adversarial robustness for unsupervised domain adaptation," *arXiv preprint arXiv:2109.00946*, 2021.

[12] B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti, "Stabilizing gan training with multiple random projections," *arXiv preprint arXiv:1705.07831*, 2017.

[13] I. Albuquerque, J. Monteiro, T. Doan, B. Considine, T. Falk, and I. Mitliagkas, "Multi-objective training of generative adversarial networks with multiple discriminators," in *International Conference on Machine Learning*. PMLR, 2019, pp. 202–211.

[14] N. X. Vinh, S. Erfani, S. Paisitkriangkrai, J. Bailey, C. Leckie, and K. Ramamohanarao, "Training robust models using random projection," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 531–536.

[15] Z. Xu, D. Liu, J. Yang, C. Raffel, and M. Niethammer, "Robust and generalizable visual representation learning via random convolutions," in *International Conference on Learning Representations*, 2020.

[16] C. Hegde, M. Wakin, and R. Baraniuk, "Random projections for manifold learning," *Advances in neural information processing systems*, vol. 20, pp. 641–648, 2007.

[17] A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng, "On random weights and unsupervised feature learning," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011, pp. 1089–1096.

[18] G.-A. Thanei, C. Heinze, and N. Meinshausen, "Random projections for large-scale regression," in *Big and complex data analysis*. Springer, 2017, pp. 51–68.

[19] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," *Advances in neural information processing systems*, vol. 24, pp. 2178–2186, 2011.

[20] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *International Conference on Machine Learning*, 2013, pp. 10–18.

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

[22] S. B. David, T. Lu, T. Luu, and D. Pál, "Impossibility theorems for domain adaptation," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 129–136.

[23] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 180–191.

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

[25] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

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

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

[28] M. Narayanan, V. Rajendran, and B. Kimia, "Shape-biased domain generalization via shock graph embeddings," in *The IEEE International Conference on Computer Vision (ICCV)*, 2021.

[29] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," *arXiv preprint arXiv:2007.01434*, 2020.

[30] Z. Huang, H. Wang, E. P. Xing, and D. Huang, "Self-challenging improves cross-domain generalization," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 124–140.

[31] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization," *arXiv preprint arXiv:1911.08731*, 2019.

[32] M. Narayanan, V. Rajendran, and B. Kimia, "Shape-biased domain generalization via shock graph embeddings," *arXiv preprint arXiv:2109.05671*, 2021.

[33] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[34] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[35] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[36] H. Kannan, A. Kurakin, and I. Goodfellow, "Adversarial logit pairing," *arXiv preprint arXiv:1803.06373*, 2018.

[37] C. Mao, Z. Zhong, J. Yang, C. Vondrick, and B. Ray, "Metric learning for adversarial robustness," in *Advances in Neural Information Processing Systems*, 2019, pp. 480–491.

[38] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning*, 2019, pp. 7472–7482.

[39] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2229–2238.

[40] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1446–1455.

A Appendix

A.1 Extra results

A.1.1 Divergence estimation

In order to further highlight that the use of RPODS help to decrease domain divergences, we plot in Figure 4 the values of the $\mathcal{A}$-distance obtained by a training a model on the original input space versus the $\mathcal{A}$-distance values achieved by a model trained on top of the projected space (i.e. with RPODS) for each pair of domain. Each value is indicated by an "x", and markers lying below the dashed line indicate the cases where the $\mathcal{A}$-distance was higher for the model trained on the original input space. Notice that all points lie below the diagonal, indicating that, for all studied cases, RPODS were able to reduce domain shift.

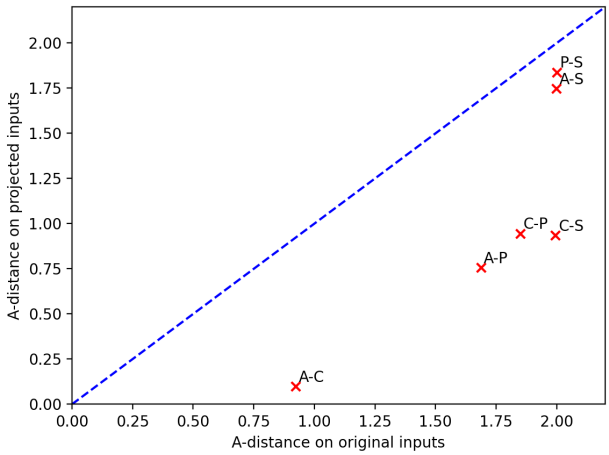

Figure 4: Pair-wise $\mathcal{A}$-distances for all domains within the PACS dataset. Each 'x' indicate the distance values for a pair of domains. The x-axis represents the distance estimated on the original input space, while the y-axis corresponds to distance values computed with RPODS. Points lying below the dashed line indicate a decrease in $\mathcal{A}$-distance when using RPODS.

A.1.2 Domain Generalization

We provide additional results for the experiments under the domain generalization setting in Table 3. In addition to the results presented in Table 1, we include JiGen [39] and Episodic-DG [40] in the comparison and present the standard deviation of accuracy values obtained by RPODS across three run. Additionally, we report the performance of RPODS considering the "Oracle" selection criterion as introduced by [29]. In this case, the best performance selected by computing the accuracy on a partition of the data corresponding to the unseen domain.

Table 3: Domain generalization results on PACS considering a leave-one-domain-out training scheme using the accuracy on the validation set of the training domains as model selection criterion. The * indicates results reported in [28].

Method	Selection	Photo	Art	Cartoon	Sketch
SC*	Training domain val. set	55.02	42.38	53.28	37.15
GroupDRO*	Training domain val. set	51.20	32.20	37.30	35.70
Episodic-DG *	Training domain val. set	41.13	29.83	42.15	37.69
JiGen*	Training domain val. set	42.34	30.37	45.65	29.14
MLDG*	Training domain val. set	47.30	29.30	40.30	28.80
GNN-Tag*	Training domain val. set	53.23	33.26	49.16	54.15
ERM*	Training domain val. set	14.07	11.31	15.72	20.69
RPODS	Training domain val. set	63.90 $\pm$ 0.78	42.63 $\pm$ 0.42	51.74 $\pm$ 1.74	56.67 $\pm$ 1.39
RPODS	Oracle	64.15 $\pm$ 0.11	38.34 $\pm$ 1.64	52.65 $\pm$ 0.52	58.18 $\pm$ 2.26

A.2 Experimental details

$\mathcal{A}$-distance estimation. In order to estimate the $\mathcal{A}$-distance, we consider a hypothesis class corresponding to all models parameterized by a ResNet-18. We train both models with SGD with a learning set to equal to 0.001 and weight decay parameter equal to 0.00001. We report the accuracy on the validation partition of each domain after 10 training epochs.

Domain generalization. We implemented RPODS and run the experiments on the domain generalization setting using [29] with the following hyperparameters:

- Batch size: 32
- Iterations: 5000
- Learning rate: 5e-4
- Number of random projections: 3
- Random projection kernel size: 8
- Random projection stride: 1
- Weight decay: 0.0
- Dropout: 0.0

Adversarial robustness. We trained a ResNet with SGD using the hyperparameters reported below. Attacks were implemented using *FoolBox*[2].

- Batch size: 64
- Epochs: 600
- Initial learning rate: 0.1
- Schedule: Decay the learning by a factor of 10 at epochs $[10, 150, 250, 350]$.
- Number of random projections: 3
- Random projection kernel size: 3
- Random projection stride: 1
- Weight decay: 0.0005
- Dropout probability: 0.3

[2] `https://foolbox.readthedocs.io/en/stable/index.html`