DAIR: Data Augmented Invariant Regularization

Tianjian Huang [1 2] **Chinnadhurai Sankar** [2] **Pooyan Amini** [2] **Satwik Kottur** [2] **Alborz Geramifard** [2]
Meisam Razaviyayn [1] **Ahmad Beirami** [2]

Abstract

A fundamental problem in machine learning is to learn representations that are invariant to certain transformations. For example, image representations are desired to be invariant to translation, rotation, changes in color, or background; natural language representations ought to be invariant to named entities. Naturally, data augmentations are a simple yet powerful way to address such invariance. However, such data augmentations requiring either additional data collection or careful engineering to capture all invariances. In this paper, we argue that a simple yet effective additional loss, called Data Augmented Invariant Regularization (DAIR), could improve the performance even further. DAIR promotes additional invariance on top of data augmentations at little marginal cost, and is consistent with any learning model. We empirically evaluate the performance of DAIR on two vision tasks, Colored MNIST and Rotated MNIST, and demonstrate that it provides non-trivial gains beyond data augmentation, outperforming invariant risk minimization.

1. Introduction

Deep neural networks have been widely used in various applications, not only in the field of image recognition ans language processing. Variants of these powerful models solve learning problems from image classification to machine translation outperforming human accuracy However, these neural networks are vulnerable to learning spurious correlations and therefore fail to generalize out-of-distribution (Arjovsky et al., 2019). Domain generalization datasets like Rotated MNIST (Arjovsky et al., 2019), Colored MNIST (Arjovsky et al., 2019), PACS (Li et al., 2017), VLCS (Fang et al., 2013), Office-Home (Venkateswara et al., 2017), Terra Incognita (Beery et al., 2018) and DomainNet (Peng et al., 2019) have shown difficulties to Empirical Risk Minimization (ERM) based algorithms. For example, in learning end-to-end dialogue models, Qian et al. (2021) show 29% performance drop on MultiWOZ (Budzianowski et al., 2018) due to the memorization of named entities.

There are two main existing directions for promoting such invariance to transformations: 1) Data augmentation, which promotes such invariances by curating synthetic examples that exhibit such invariances. 2) Geometric deep learning, which bakes such invariances into the neural network architectures used for training. For example, convolutional layers are fundamentally preserving translations, or some other specifically designed networks (Zaheer et al., 2017; Bloem-Reddy & Teh, 2020; Finzi et al., 2021).

Besides two main directions mentioned above, researchers have proposed several other algorithmic solutions to address this issue: IRM (Arjovsky et al., 2019), DRO (Sagawa et al., 2019), Mixup (Yan et al., 2020), MLDG (Li et al., 2018a), CORAL (Sun & Saenko, 2016), MMD (Li et al., 2018b), DANN (Ganin et al., 2016), C-DANN (Li et al., 2018c) and fine-tuned ERM (Gulrajani & Lopez-Paz, 2020). Most of the approaches listed above are complicated, and cannot be applied to arbitrary tasks, such as generation.

In this paper, we propose a simple yet effective regularization technique, called augmented invariant regularization (DAIR) that promotes invariance with respect to the augmented data. DAIR is simple to implement, offers marginal additional cost to training with data augmentation, is compatible with any task including generation, and is competitive with state-of-the-art algorithmic approaches

2. DAIR: Data Augmented Invariant Regularization

Let $z = (x, y)$ denote a data sample, and let $\ell(z; \theta)$ be a parametric loss function, where θ is the set of model parameters (e.g., network weights). We assume that we have access to a (potentially randomized) data augmenter function $A(\cdot)$. Examples for A include (random) rotation and coloring. Such augmenters in general capture all the desired transformations against which we wish to be invariant. Given a sample z, let $\widetilde{z} = (\widetilde{x}, \widetilde{y}) = A(z)$ denote an augmented

[1]University of Southern California [2]Facebook AI. Correspondence to: Tianjian Huang <tianjian@usc.edu>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning.,

sample. In this paper, we optimize the following augmented objective function:

$$f_{\gamma,\lambda}(z,\widetilde{z};\theta) = \gamma\ell(z;\theta) + (1-\gamma)\ell(\widetilde{z};\theta) + \lambda r(z,\widetilde{z};\theta) \quad (1)$$

where DAIR regularizer is defined as

$$r(z,\widetilde{z};\theta) := \left(\sqrt{\ell(z;\theta)} - \sqrt{\ell(\widetilde{z};\theta)}\right)^2. \quad (2)$$

The main purpose we choose square root function in (2) is to make the scale of $r(\cdot)$ and $\ell(\cdot)$ the same, hence making the tuning of λ easier across different tasks. We observe that empirically there are several working substitutions such as regular MSE and Jensen-Shannon divergence. By setting γ and λ to desired values, one can have the following training scenarios.

- Empirical Risk Minimization (ERM):

$$f_{\text{ERM}}(z,\widetilde{z};\theta) = \ell(z;\theta)$$

- Data Augmented Empirical Risk Minimization (DA-ERM):

$$f_{\text{DA-ERM}}(z,\widetilde{z};\theta) = \frac{1}{2}\ell(z;\theta) + \frac{1}{2}\ell(\widetilde{z};\theta)$$

- Data Augmented Invariant Regularization (DAIR):

$$f_{\text{DAIR},\lambda}(z,\widetilde{z};\theta) = \frac{1}{2}\ell(z;\theta) + \frac{1}{2}\ell(\widetilde{z};\theta) + \lambda r(z,\widetilde{z};\theta)$$

The idea behind the regularizer in (2) is simple. For a fully transformation invariant model, $\ell(z;\theta) \approx \ell(A(z);\theta)$ will hold for the data augmenter, $A(\cdot)$, corresponding to such transformation function.

Lemma 1 (Gradient of augmented loss)**.** *We have*

$$\nabla_\theta f_{\gamma,\lambda}(z,\widetilde{z};\theta) = \left(\gamma + \lambda - \lambda\sqrt{\frac{\ell(\widetilde{z};\theta)}{\ell(z;\theta)}}\right)\nabla_\theta\ell(z;\theta) + \left(1 - \gamma + \lambda - \lambda\sqrt{\frac{\ell(z;\theta)}{\ell(\widetilde{z};\theta)}}\right)\nabla_\theta\ell(\widetilde{z};\theta)$$

The second term in Lemma 1 behaves as the correction term, which promotes invariance. Note the corrections can be huge when $\ell(\cdot)$ is close to zero. One can also observe that ℓ and r are at the same scale, making the tuning of λ and the optimization framework friendly. Unlike other work (Arjovsky et al., 2019) which imposes invariance in the latent space, we impose invariance on the output of the very last layer. This crucial difference brings the possibility of extending our regularizer to generative models.

Scheme	z	Color | $y=0$
C1	with $p=0.8$, $z=y$	Red
	with $p=0.2$, $z=1-y$	Green
C2	with $p=0.9$, $z=y$	Red
	with $p=0.1$, $z=1-y$	Green
C3	with $p=0.1$, $z=y$	Red
	with $p=0.9$, $z=1-y$	Green
C4	$z=2$	Random

Table 1. Color schemes in Colored MNIST. Random color means that the value of each channel of the image is uniformly random chosen from 0 to 255.

3. Experiments

In this paper, we apply the proposed loss function (1) on the following two datasets: Extended Colored MNIST and Extended Rotated MNIST. We compare the performance of DAIR with plain data augmentation, and invariant risk minimization (IRM) as a strong baseline. One crucial difference between our work and IRM is is the motivation. IRM is designed to take two examples from two different environments and learn representations that are invariant to the environment, e.g., in cases where we are aggregating multiple datasets. On the other hand, we are interested in promoting invariance when we have a single dataset. As such, we artificially generate the second environment in IRM using data augmentation. For a given example z, we design an augmenter $A(\cdot)$ and use it to generate additional samples that adhere to the invariance we have in mind. Hence, IRM will be applied in the same way that examples from different environments are augmenting pairs.

Extended Colored MNIST is an extension of the original Colored MNIST (Arjovsky et al., 2019). The label is a noisy function of both digit and color. The digit has a correlation of 0.75 with the label and a certain correlation with the label depending on the color scheme. Besides the two colors in the original dateset, we introduce fully random colored scheme to the dateset, which is the best augmenter one can think of. The three color schemes are detailed in Table 1.

Extended Rotated MNIST is a variant of the original Rotated MNIST (Ghifary et al., 2015). The original dataset contains images of digits rotated d degrees, where $d \in \mathcal{D} \triangleq \{0, 15, 30, 45, 60, 75\}$. Similarly, we introduce the random degree scheme here to serve as the best possible augmenter. To further exploit the potential of the proposed algorithm, we make this dataset more difficult by introducing more challenging degree scheme; The rotation schemes are summarized in Table 2.

Note all the augmented images are generated on the fly. Examples of images from some transformation schemes are shown in Figures 1 to 6.

Scheme	Rotation
R1	$0°$
R2	$90°$
R3	$0°, 180°$
R4	$90°, 270°$
R5	$[0°, 360°]$
R6	$[22.5°, 67.5°], [202.5°, 247.5°]$

Table 2. Rotation schemes in Rotated MNIST. $[a, b]$ means that degrees are unformly random chosen between a and b.

Setup	Train	Aug	Test	λ
1	C1	C2	C3	1000
2	C1	C4	C3	100

Table 3. Training procedure of Colored MNIST. The value of λ has been chosen from $\{1, 10, 100, 1000, 10000\}$ based on validation performance. See Appendix A for details.

Setup	Train	Aug	Test	λ
3	R1	R5	R2	1
4	R4	R6	R3	10

Table 4. Training procedure of Rotated MNIST. The value of λ has been chosen from $\{1, 10, 100, 1000\}$ based on validation performance. See Appendix A for details.

Setup: We train a model consisted of three convolutional layers and two fully connected layers with 20,000 examples. For each dataset we are defining several different schemes on how the dataset could be modified: Table 1 (Colored MNIST) and Table 2 (Rotated MNIST). Then, we define several *setups*. Each setup is consisted of one original dataset, one augmentation dataset, and one test dataset, each of which is selected among the defined schemes. These setups are provided in Table 3 (Colored MNIST) and Table 4 (Rotated MNIST). For each setup, we train the model with the following four algorithms and compare their performances: ERM, DA-ERM, DAIR and Invariant Risk Minimization (IRM). Each experiment is repeated for 10 times; the mean and the standard derivation are reported. The value of λ are chosen base on the validation results. Detailed architectures and training parameters can be found in Appendix B and detailed results can be found in Appendix A.

3.1. Colored MNIST

We conduct two sets of experiments for this dataset: Setup 1 (Table 3) follows the exact same color schemes from the original Colored MNIST (Arjovsky et al., 2019). For Setup 2, we train the model with the strongest possible augmenter: uniformly random color. The entire procedure is summarized in Table 3.

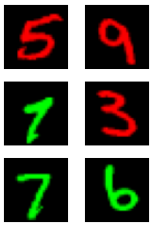

Figure 1. C2

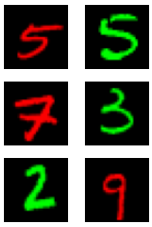

Figure 2. C3

Figure 3. C4

Figure 4. R4

Figure 5. R5

Figure 6. R6

Results: We compare our results with the IRM (Arjovsky et al., 2019) approach, as a strong baseline. For Setup 1, Table 5 suggests that DAIR outperforms IRM by as much as 3% (in the original problem setup that IRM addresses). More importantly, DAIR is much simpler and computationally efficient compared with IRM. One may also observe that ERM does not work at all even with augmentation. However, with the same augmentation set, DAIR boosts the performance by 30%. Note that the regularizer is surprisingly effective given the fact that color schemes C1 and C2 are very similar, meaning that $z = A(z)$ most of the time, whereas the test set C3 is very different. In Setup 2, with a stronger augmenter, the performance of DAIR is even better, approaching the theoretical upper bound in this problem, i.e., 75%. Note that the stronger augmenter does not necessarily give better results for other algorithms. See Table 6. Finally, we note that the performance of IRM decreases in this setup, which is not unexpected as IRM is not designed for this setup.

3.2. Rotated MNIST

We start with the strongest augmenter in Setup 3. One may notice that there is a chance that the augmented images bear the same rotation degrees as the testing set. To make the task more difficult, we will use R6 as the augmented test to test how the trained model generalize to entirely unseen domain. The training procedure is summarized in Table 4.

Results: In the strongest augmenter case in Setup 3. The augmented images are uniformly random rotated and therefore there exists possibilities that the testing samples are

Algorithm	Accuracy
ERM	32.70 ± 0.45
DA-ERM	40.91 ± 0.45
DAIR	72.58 ± 0.11
IRM	68.06

Table 5. CM, Setup #1

Algorithm	Accuracy
ERM	32.70 ± 0.45
DA-ERM	29.61 ± 0.80
DAIR	73.10 ± 0.12
IRM	63.13

Table 6. CM, Setup #2

Algorithm	Accuracy
ERM	18.82 ± 0.62
DA-ERM	95.82 ± 0.06
DAIR	95.95 ± 0.08
IRM	—

Table 7. RM, Setup #3

Algorithm	Accuracy
ERM	17.71 ± 0.52
DA-ERM	71.05 ± 0.38
DAIR	84.08 ± 0.32
IRM	42.33

Table 8. RM, Setup #4

rotated the same way as the augmented samples, i.e., the model may explicitly see the testing distribution during training. Since this augmenter is very strong, as expected, all the algorithms work well in this scenario except vanilla ERM. We do not include IRM here since one can always trivially set the penalty coefficient equals to 0, ending up with DA-ERM. See Table 7 for results. However, it is not always possible to construct such a strong augmentation set for complicated tasks, such as NLP tasks. Therefore, in Setup 4, we introduce a weak augmenter. It clear that the training set, augmenting set and testing set have no overlapping, i.e., the rotation degrees of images from each set are never the same. Table 8 suggests that DAIR outperforms DA-ERM. Models trained with DAIR shows very strong generality towards unseen distributions. We also do not include the IRM result here as we can not get a result better then DA-ERM with non-trivial penalty coefficient. It is clear that DAIR boosts the invariance by a large margin. On top of the augmentation set, we gain additional invariance at almost no cost.

4. Conclusion

In this paper, we proposed a simple yet effective regularizer that can be used wherever data augmentation is used to promote invariance. We empirically verified the performance of the proposed regularization technique on two vision tasks. It remains as future work to expand the empirical verification to more benchmarks, especially to tasks beyond computer vision. Better understanding of the properties of the regularizer and tuning of the corresponding hyperparameter also remains as open areas for research.

References

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

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

Bloem-Reddy, B. and Teh, Y. W. Probabilistic symmetries and invariant neural networks. *J. Mach. Learn. Res.*, 21: 90–1, 2020.

Budzianowski, P., Wen, T.-H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., and Gašić, M. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018.

Fang, C., Xu, Y., and Rockmore, D. N. 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.

Finzi, M., Welling, M., and Wilson, A. G. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. *arXiv preprint arXiv:2104.09459*, 2021.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Ghifary, M., Kleijn, W. B., Zhang, M., and Balduzzi, D. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559, 2015.

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

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

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018a.

Li, H., Pan, S. J., Wang, S., and Kot, A. C. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018b.

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., and Tao, D. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 624–639, 2018c.

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

Qian, K., Beirami, A., Lin, Z., De, A., Geramifard, A., Yu, Z., and Sankar, C. Annotation inconsistency and entity bias in MultiWOZ. *The 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, July 2021.

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

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017.

Yan, S., Song, H., Li, N., Zou, L., and Ren, L. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, 2017.

A. Detailed Results in Section 3

We show results of validation accuracies of the models trained with different λ.

λ	Accuracy
1	47.87 ± 0.62
10	57.02 ± 0.50
100	70.93 ± 0.18
1000	72.73 ± 0.06
10000	70.38 ± 2.10

Table 9. Setup $\#1$

λ	Accuracy
1	63.88 ± 0.28
10	71.31 ± 0.08
100	73.54 ± 0.08
1000	71.23 ± 0.16
10000	57.01 ± 2.24

Table 10. Setup $\#2$

λ	Accuracy
1	96.15 ± 0.03
10	93.95 ± 0.05
100	72.82 ± 0.31

Table 11. Setup $\#3$

λ	Accuracy
1	74.36 ± 0.33
10	84.68 ± 0.19
100	71.00 ± 0.73

Table 12. Setup $\#4$

B. Model Architecture and Training Parameters

Layer Type	Shape
Convolution + ReLU	$4 \times 4 \times 6$
Max Pooling	2×2
Convolution + ReLU	$4 \times 4 \times 16$
Max Pooling	2×2
Convolution + ReLU	$4 \times 4 \times 96$
Fully Connected + ReLU	64
Fully Connected	C

Table 13. Model Architecture, $C = 1$ for Colored MNIST and $C = 10$ for Rotated MNIST

Parameter		
Learning Rate	0.005	0.0005
Epochs	20	20
Batch-size	64	

Table 14. Training parameter