DAIR: DATA AUGMENTED INVARIANT REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

While deep learning through empirical risk minimization (ERM) has succeeded at achieving human-level performance at a variety of complex tasks, ERM generalizes poorly to distribution shift. This is partly explained by overfitting to spurious features such as background in images or named entities in natural language. Synthetic data augmentation followed by empirical risk minimization (DA-ERM) is a simple and widely used solution to remedy this problem. In addition, consistency regularization could be applied to further promote model performance to be consistent on the augmented sample and the original one. In this paper, we propose data augmented invariant regularization (DAIR), a simple form of consistency regularization that is applied directly on the loss function rather than intermediate features. Through extensive empirical experiments, we show that DAIR consistently performs well in a variety of settings. We apply DAIR to multiple real-world learning problems, namely robust regression, visual question answering, robust deep neural network training, and neural task-oriented dialog modeling. Our experiments show that DAIR consistently outperforms ERM and DA-ERM with little marginal cost and sets new state-of-the-art results in several benchmarks.

1 INTRODUCTION

Deep neural networks are widely used in various applications ranging from computer vision to language processing. While deep learning has surpassed human-level performance in numerous tasks, neural networks are extremely vulnerable to overfitting to spurious correlations and therefore fail to generalize even under slight perturbations of the test distribution (Arjovsky et al., 2019). This observation motivated the research community to tackle the problem of *domain generalization* (see (Ribeiro et al., 2020) for a detailed literature review). Recent benchmark datasets, such as 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 for the generalization of deep neural network models under distribution shifts, and have sparked invention of many new algorithmic frameworks to address domain generalization.

A standard approach for improving out-of-distribution performance is to guarantee that learned models are invariant to certain transformations. For example, trained models for computer vision should generally be invariant to rotations, changes in color, or background.

Geometric deep learning bakes such invariances into the neural network architecture. For example, convolutional layers (Lecun et al., 1998) are fundamentally preserving translations. There are other specifically designed networks to maintain invariances: Zaheer et al. (2017) studied the problem of designing models for machine learning tasks defined on sets and characterized the permutation invariant functions. Bloem-Reddy & Teh (2020) obtained generative functional representations of probability distributions that are invariant under the action of a compact group. Finzi et al. (2021) provided an algorithm for solving for the equivariant layers of matrix groups.

Data augmentation promotes invariances in models by curating synthetic examples that exhibit the desired invariances. Tensmeyer & Martinez (2016) showed simple image transformations affect the CNN representations. Mixup (Zhang et al., 2017), CutMix (Yun et al., 2019) and Cutout (DeVries & Taylor, 2017) showed linear combination and random blocking features improves generalization of state-of-the-art neural network architectures. Volpi et al. (2018); Zhou et al. (2020) showed data augmentation with adversarial images could make the label classifier more robust to unknown domain shifts. Cubuk et al. (2018); Lim et al. (2019) introduced a procedure which automatically searches for improved data augmentation policies. Zhou et al. (2020) showed data augmentation with adversarial images could make the label classifier more robust to unknown domain shifts. Nam et al. (2021)

improved domain generalization by reducing the intrinsic style bias of CNNs through training a separate network for randomizing the style of images and generating augmented data during training.

Consistency regularization can be further applied on top of data augmentation to enhance invariance by enforcing similarities on the model. Engstrom et al. (2018); Kannan et al. (2018); Zhang et al. (2019) utilized consistency regularization to train robust neural networks against adversarial attacks. This has been applied to unsupervised learning (Sinha & Dieng, 2021), self-supervised learning (Chen et al., 2020; von Kügelgen et al., 2021), and semi-supervised learning to exploit unlabeled data (Bachman et al., 2014; Laine & Aila, 2016; Sohn et al., 2020; Xie et al., 2020).

Besides the directions mentioned above, researchers have proposed numerous algorithmic solutions to impose invariance and improve domain generalization such as DANN (Ganin et al., 2016), IRM (Ghifary et al., 2015), DRO (Sagawa et al., 2019), MLDG (Li et al., 2018a), CORAL (Sun & Saenko, 2016), MMD (Li et al., 2018b) and CDANN (Li et al., 2018c) and REx (Krueger et al., 2021). The approaches listed above are more complex than simple training mechanisms such as empirical risk minimization (ERM) and hence they cannot be readily applied to involved tasks with non-trivial model architectures. For example, in generative language models imposing a constraint on the intermediate data representations is non-trivial, which is required by CORAL (Sun & Saenko, 2016). Recently, Gulrajani & Lopez-Paz (2020) demonstrated that ERM may even outperform many such complex methods in real-world scenarios, while ERM itself is known to generalizes poorly to distribution shift. For example, in learning neural dialog models, Qian et al. (2021) showed up to 29% performance drop due to the memorization of named entities. Ribeiro et al. (2020) showed that both commercial and state-of-art language models fail on up to 76.4% of the generalization tests.

In this paper, we propose a consistency regularization technique, called data augmented invariant regularization (DAIR). DAIR is applicable when data augmentation results in pairs of data samples expecting consistent performance, it specifically penalizes the inconsistency of loss on augmented samples with respect to the original ones. This is in contrast to many feature consistency regularizers that apply on an intermediate embedding space. As a result, DAIR only requires marginal additional cost on top of data augmentation, and is simple and broadly applicable to a wide host of supervised and unsupervised learning tasks, including generative models. We introduce the DAIR formulation, motivate it, and theoretically prove some of its properties in Section 2. We empirically evaluate DAIR on a variety of problem setups ranging from defense against adversarial attacks to domain generalization in the presence of environment shift in Section 3, where our experimental results show that DAIR is competitive with or even outperforms state-of-the-art algorithms specifically designed for imposing invariance in these problems.

2 DAIR: Data Augmented Invariant Regularization

For a data sample $z = (x, y)$, let $\ell(z; \theta)$ be its parametric loss function, where θ is the set of model parameters (e.g., network weights). The popular Empirical Risk Minimization (ERM) framework trains the model by minimizing the expected value of the following loss over the training data:

$$f_{\text{ERM}}(z; \theta) = \ell(z; \theta). \tag{ERM}$$

We assume that we have access to a (potentially randomized) data augmenter function $A(\cdot)$. Examples for A include (random) rotation, change of background, or change of entity names. Such augmenters aim at capturing the transformations against which we wish to be invariant to. Given a sample z, let $\widetilde{z} = (\widetilde{x}, \widetilde{y}) = A(z)$ denote an augmented sample. Previous work has used both original and augmented examples during training, which leads to the following standard objective function, called Data Augmented Empirical Risk Minimization (DA-ERM):

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

While DA-ERM has been successful in many applications, one natural question is whether we can further improve upon it using the knowledge that the performance on augmented samples should be consistent with the original ones. Consistency regularization further penalizes DA-ERM for any such inconsistency at the feature/loss level: $f_{\text{Consistency},\mathcal{D},\lambda}(z, \widetilde{z}; \theta) = f_{\text{DA-ERM}}(z, \widetilde{z}; \theta) + \lambda\mathcal{D}(z, \widetilde{z}; \theta)$, where $\mathcal{D}(z, \widetilde{z}; \theta)$ is a proper divergence between the original sample representation and the augmented sample representation, and where the goal of the regularizer applied at some intermediate feature space is to maintain the performance of the model on z and $\tilde{z}$ consistent. In this paper, we focus on a specific type of such regularization, called data augmented invariant regularization (DAIR):

$$\begin{aligned} f_{\text{DAIR},\mathcal{R},\lambda}(z,\widetilde{z};\theta) &= f_{\text{DA-ERM}}(z,\widetilde{z};\theta) + \lambda\mathcal{D}(z,\widetilde{z};\theta) \\ &= \frac{1}{2}\ell(z;\theta) + \frac{1}{2}\ell(\widetilde{z};\theta) + \lambda\mathcal{R}(\ell(z;\theta),\ell(\widetilde{z};\theta)), \end{aligned} \tag{DAIR}$$

where the regularization is directly applied to the loss. The idea behind DAIR is to simply promote $\ell(z;\theta) \approx \ell(\widetilde{z};\theta)$, and ignore the features or even the rest of the possible outcomes of y and simply focus on the current sample's loss. Hence, DAIR is a relatively weak form of consistency regularization only enforcing an original sample and an augmented one to be equally likely under the learned model (assuming loss is a log-likelihood function). This weaker form of consistency is suitable for problems where feature consistency may not be conceptually meaningful. For instance, in language modeling when a pair of sentences differ in their corresponding named entities, it is not clear why we should enforce their embeddings to be similar, however, loss consistency is still meaningful promoting the probability of label given input to be the same on the original and the augmented samples.

We remark that DAIR requires pairing information between original and augmented samples, which may not always be available (e.g., DomainBed (Gulrajani & Lopez-Paz, 2020)). However, we show that this simple approach is still broadly applicable to various real-world problems regardless of model architecture, and is indeed competitive with state-of-the-art methods for imposing invariance. As it turns out, we are particularly interested in a particular form of the DAIR regularizer:

$$\mathcal{R}_{\text{sq}}(\ell(z;\theta),\ell(\widetilde{z};\theta)) := \left(\sqrt{\ell(z;\theta)} - \sqrt{\ell(\widetilde{z};\theta)}\right)^2, \tag{SQ Regularizer}$$

and we call this variant DAIR-SQ. Note that $\mathcal{R}_{\text{sq}}$ has the same scale as the loss function ℓ, making it easier to tune λ. Empirically we observe that the optimal λ for all the experiments mentioned later in the paper falls in $[0.2, 100]$, across various tasks (from regression to sequence-to-sequence generative modeling). Further justification on DAIR-SQ will be provided through the rest of this section.

Finally, in most (real-world) applications performance is measured through 0-1 metrics other than the loss function. For example, we are usually concerned with accuracy in image classification while we optimize cross-entropy loss. Let $F(z;\theta) \in \{0,1\}$ denote a 0-1 evaluation performance metric of interest, e.g., accuracy. Given the sample z (or $\widetilde{z}$), the model performance is captured by $F(z;\theta)$ (or $F(\widetilde{z};\theta)$). For any z such that $F(z;\theta) = 1$, we define the corresponding consistency metric as:

$$\text{CM}(z,\widetilde{z};\theta) = \mathbb{I}\{F(\widetilde{z};\theta) = 1 \mid F(z;\theta) = 1\}. \tag{Consistency Metric}$$

Notice that similarly to the original performance metric, which is only used for model evaluation, we use the consistency metric at evaluation time only.

2.1 What does DAIR offer beyond DA-ERM?

To motivate DAIR, we consider a toy example through which we demonstrate that DAIR can fundamentally outperform DA-ERM, even in the limit of infinite training samples (no overfitting due to finite samples). Consider a linear regression problem where at the training time the input is $\mathbf{x}_{\text{train}} = (x, s = y)$ and the label y, i.e., $z_{\text{train}} = (\mathbf{x}_{\text{train}}, y)$. Here, $x \sim \mathcal{N}(0,\sigma_x^2)$, and $y = x + \varepsilon$, where ε is independent of x and $\varepsilon \sim \mathcal{N}(0,\sigma_\varepsilon^2)$. In this example, the target is explicitly provided as a spurious feature to the learner at the training time. At test time, the spurious feature is absent, i.e., $\mathbf{x}_{\text{test}} = (x, s = 0)$.

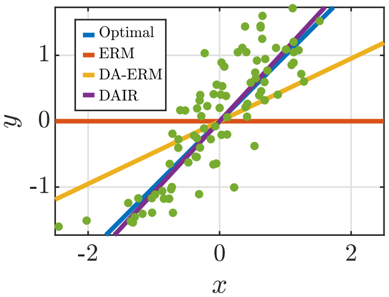

Figure 1: The plot of the optimal, ERM, DA-ERM and DAIR-SQ ($\lambda = 100$).

Clearly, in this toy example, the optimal regressor is $w^\star = (w_1^\star, w_2^\star)^\top = (1,0)^\top$. However, absent the knowledge of the spurious feature vanilla ERM will learn $w_{\text{ERM}} \approx (0,1)^\top$, completely overfitting the spurious feature. We assume that the learner has access to a data augmentation module that generates $\widetilde{z} = A(z; a, \sigma_n^2) = (\mathbf{x}_{\text{aug}}, y)$, such that $\mathbf{x}_{\text{aug}} = (x, s = ay + n)$ where $n \sim \mathcal{N}(0,\sigma_n^2)$. The augmented data will encourage the learned model to become invariant to the spurious feature. In Figure 1, we perform simulations with $a = 0.5,\ \sigma_x^2 = 1,\ \sigma_\varepsilon^2 = 0.25,\ \sigma_n^2 = 0.1$ and plot four lines associated with each regressor with the slope of their respective w_1. We ignore w_2 as the second spurious feature is absent at test time and hence w_2 does not impact test performance. The optimal regressor is shown as the blue line, with a slope of 1. ERM (red line) completely fails due to the overfitting to the spurious feature. DA-ERM (orange line) significantly improves over ERM but still is far from optimal performance. DAIR-SQ (purple line) almost recovers the optimal solution. This is not a coincidence. We prove that DAIR-SQ is optimal for a class of linear regression problems, while DA-ERM does not approach optimal performance even in the limit of infinite samples. In other words, DAIR can lead to better generalizing models beyond simply offering better sample complexity.

Proposition 1. *Consider the class of linear regression problems described above with a spurious feature (highly correlated with the output). Assume that the learner has access to a data augmentation module that perturbs the spurious feature. Then, for any value of a and σ_n, DAIR-SQ achieves optimal test error as number of samples grows and $\lambda \to \infty$. On the other hand, DA-ERM cannot recover optimal performance even in the limit of infinite training data unless $\sigma_n \to \infty$.*

The proof of Proposition 1 is relegated to Appendix A. One can show that simple data independent regularization methods (e.g. weight decay) cannot help close the gap between the performance of DA-ERM and DAIR (see Proposition 2) in Appendix A. While we only analyzed DAIR-SQ, we believe the content of this proposition extends to other variants of DAIR as well. Note that when $\sigma_n \to \infty$, DA-ERM could also recover $w^\star$. One can interpret that as $\sigma_n \to \infty$, the augmenter becomes stronger and forces w_2 to vanish. On the other hand, DAIR recovers $w^\star$ with a much weaker augmenter. This is crucial since in real-world applications, designing strong augmentation schemes requires careful design. We will expand on this in Section 2.2.

2.2 Variants of DAIR vs other consistency regularization techniques

In this section, we empirically compare ERM, DA-ERM and some variants of consistency regularization, including two DAIR variants on two classification tasks using CNNs. Let $\mathbf{q}(z;\theta)$ be the output of the model right after the softmax layer. If we treat the loss function as (un-normalized) negative log-likelihood of the output distribution, and let $\mathbf{q}(z;\theta) \propto e^{-\ell(z;\theta)}$. In addition to DAIR variants, we consider the regularizer to be any proper divergence between the output distributions $\mathbf{q}(z;\theta)$ and $\mathbf{q}(\widetilde{z};\theta)$, such as $\mathcal{L}_2$ distance or KL divergence, which will promote $\mathbf{q}(z;\theta) \approx \mathbf{q}(\widetilde{z};\theta)$.

Rotated MNIST (Ghifary et al., 2015) is a dataset where MNIST digits are rotated. We work with two different sets of degrees of rotation for Rotated MNIST. The first one is *Weak Rotation* where the digits are rotated uniformly at random $[0, \frac{\pi}{6})$ radians. In *Strong Rotation* the digits are rotated uniformly at random $[0, 2\pi)$ radians. To evaluate the robustness of the methods, we further add label noise at training time where the label is replaced with a digit chosen from {0,. . . ,9} uniformly at random with a certain probability. No label noise is added at test time. Detailed setup is in Appendix D.

In the first experiment, we use Weak Rotation for data augmentation while at test time we use Strong Rotation. Thus, some test time rotations have not been observed at training time. Figure 2 shows the test performance of all algorithms (averaged over three runs) as a function of λ. As can be seen, ERM (with no data augmentation) does not generalize to rotated test images and performs poorly. DA-ERM offers significant performance improvement over ERM. When λ is very small all variants of consistency regularization are virtually the same as DA-ERM. DAIR-SQ and KL regularizer outperform other regularizers and are the only two variants that offer improvement over DA-ERM as λ increases. As the label noise level becomes larger, DAIR-SQ is more robust than KL and offers the best performance.

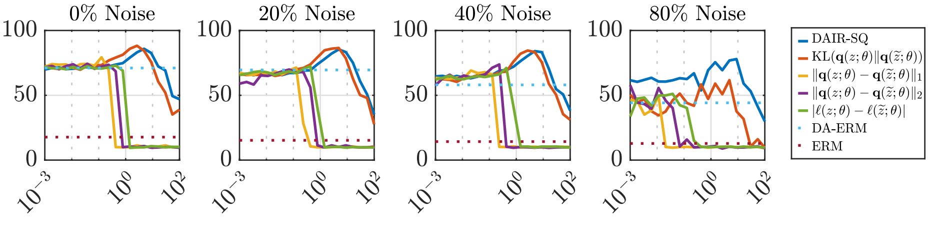

Figure 2: Test accuracy as a function of λ for different noise levels for **Weak Rotation** augmentation.

Besides DAIR-SQ and KL regularizer, it is noteworthy that the other consistency regularization variants did not offer improvement over DA-ERM and they converged to poor local minima with

10% test accuracy (random) for large λ. We were not able to remedy this by tuning of their step size. See Section 2.3 for further justification of this phenomenon. We also observe that the performance of both DAIR-SQ and KL regularizer achieves a sweet spot for some finite λ, i.e., the performance starts to drop for large values of λ. This is not theoretically expected and can be attributed to the practical issues with solving the consistency regularization problem. We further investigate this phenomenon in Appendix B and provide some explanations.

The setup for the second experiment is the same as the first one, except we also use Strong Rotation in training for augmentation, so there is no distribution shift for DA-ERM. As can be seen in Figure 3, data augmentation achieves very good performance in this case and none of the DAIR regularizers offer any improvement beyond data augmentation. We suspect this to be true in general; if the data augmentation is well-devised and optimized the resulting model could become invariant to the desired transformations at test time. This also agrees with findings of Section 2.1, where observed that with *strong* augmentation, DA-ERM could potentially result in similar performance as DAIR. Additional experiments on consistency metric can be found in Appendix E.1.

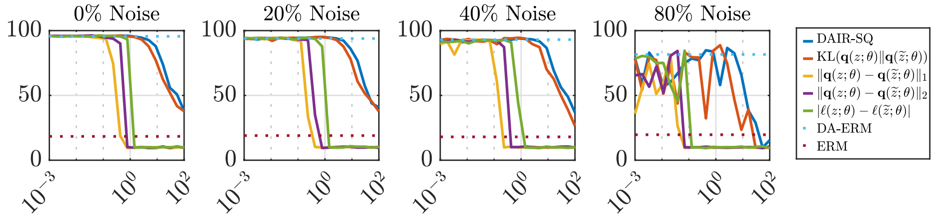

Figure 3: Test accuracy as a function of λ for different noise levels for **Strong Rotation** augmentation.

Colored MNIST (Arjovsky et al., 2019) is a binary classification task built on the MNSIT dataset. Digits 0-4 are labeled 1; whereas digits 5-9 are labeled 0. Additionally, 25% label noise is added, i.e., the labels are flipped with probability 0.25, both at train and test time, capping the achievable test accuracy to 75%. In this dataset, each digit is RGB colored. During training, label 1 is given the color green with probability 0.9 and red with probability 0.1. On the other hand, label 0 is given red color with probability 0.9 and green with probability 0.1. This introduces a high degree of spurious correlation between color and the label. Thus, ERM is expected to significantly overfit to color for predicting the label.

At test time, the correlation with color is reversed for digits. Hence, vanilla ERM is expected to perform worse than 50% coin flip at test time. We explore two data augmentation schemes in this experiment. For the *Adversarial Augmentation (Adv. Aug.)* setup, the augmented images will have their color flipped (from red to green or vice versa) with probability 0.1. For the *Random Augmentation (Rnd. Aug.)* setup, the augmented images are colored uniformly at random. Detailed description of the setup and additional experiments can be found in Appendix D and Appendix E.1, respectively.

Figure 4 suggests that DAIR-SQ and KL consistency regularization achieve $\sim$ 72% test accuracy using both augmentation schemes, outperforming the state-of-the-art 68% test accuracy reported by invariant risk minimization (IRM) (Arjovsky et al., 2019), and almost reaching the 75% cap. We note however that this comparison may be unfair because IRM does not have access to any pairing information between the original and the augmented samples. As we observe in the next section, such information is readily available in several real-world benchmarks and DAIR can exploit it to achieve new state-of-the-art results. We also notice that neither variant of DA-ERM achieves test performance better than 50% coin flip in this experiment, while Adversarial Augmentation seems to fare better than Random Augmentation.

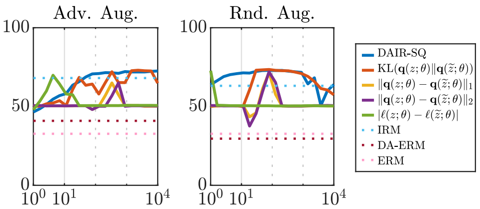

Figure 4: Test accuracy vs λ on Colored MNIST for Adversarial Color augmentation and Random Color augmentation.

Following the experiments, we conclude that DAIR-SQ is more stable and robust than other ones followed by KL divergence consistency regularization. Additionally, DAIR-SQ enjoys the simplicity and computational efficiency, especially when the cardinality of the output is large, e.g., language models where output vector dimension is the same as the vocabulary size. As opposed to KL divergence, DAIR-SQ is also readily applicable to regression with uncountable output.

2.3 Further justification of DAIR-SQ and practical considerations

While we have already compared DAIR-SQ with several consistency regularization alternatives, we want to specifically focus on a closely related DAIR variant called DAIR-L1, i.e., $\mathcal{R}_1(\ell(z;\theta), \ell(\widetilde{z};\theta)) = |\ell(z;\theta) - \ell(\widetilde{z};\theta)|$. As we already observed in Section 2.2, DAIR-L1 either outright failed or was unstable on majority of the experiments we have performed so far. The following lemma further investigates the discrepancy between DAIR-SQ and DAIR-L1:

Lemma 1. *For any non-negative loss function* ℓ,

$$\mathcal{R}_1(z, \widetilde{z}; \theta) - \mathcal{R}_{sq}(z, \widetilde{z}; \theta) = 2\sqrt{\min\{\ell(z;\theta), \ell(\widetilde{z};\theta)\}\mathcal{R}_{sq}(z, \widetilde{z}; \theta)} \geq 0.$$

Thus, $\mathcal{R}_1(z, \widetilde{z}; \theta) \geq \mathcal{R}_{sq}(z, \widetilde{z}; \theta)$ *with equality iff* $\ell(\widetilde{z};\theta) = 0$ *or* $\ell(z;\theta) = 0$ *or* $\ell(\widetilde{z};\theta) = \ell(z;\theta)$.

The proof of Lemma 1 appears in Appendix A. The difference is depicted in Figure 5. This suggests that $\mathcal{R}_{\text{sq}}(z, \widetilde{z}; \theta)$ incurs a much smaller penalty when $\ell(z;\theta)$ is large. On the other hand, when $\ell(z;\theta) \approx 0$ the regularizer is much stronger and almost equivalent to $\mathcal{R}_1$. Why does this matter? At the beginning of training when the network is not yet trained, the loss values on the original samples are large, and the $\mathcal{R}_{\text{sq}}$ regularizer is weak letting the training to proceed towards a good solution for the original samples. As the network is being trained on original samples and their loss is vanishing, the regulairzer starts to force the network to become invariant on the augmented samples.

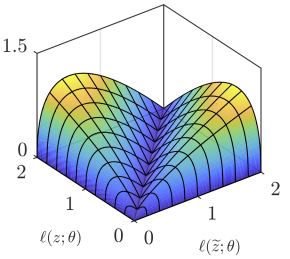

Figure 5: The plot of $\mathcal{R}_1(z, \widetilde{z}; \theta) - \mathcal{R}_{\text{sq}}(z, \widetilde{z}; \theta)$.

We empirically verify this conjecture on Colored MNIST with Adversarial Augmentation. Figure 6 depicts the classification loss and regularization of the first 10 and last 140 iterations. One observes that at the beginning of training, regularization term of DAIR-SQ impacts the training dynamics less while DAIR-L1 starts optimizing the regularizer right away, which dominates the entire training procedure and therefore leads the model to a poor local minimum. The left panel of Figure 6 confirms that the classification loss of DAIR-L1 remains large and unchanged (that of a random classifier).

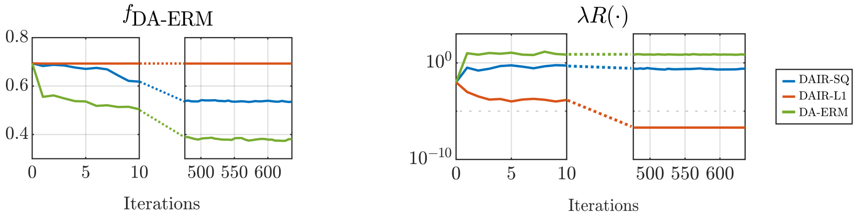

Figure 6: Training DA-ERM loss and (SQ Regularizer) for first 10 and last 140 iterations on Colored MNIST with Adv Aug for DAIR ($\lambda = 100$). The regularizer loss on DA-ERM grows large as it is uncontrolled. DAIR-L1 is optimizing an L1 regularizer, but for unified illustration we evaluate it using (SQ Regularizer).

This same property of DAIR-SQ also weakens the regularizer on training samples with high losses at the later stages of training. These samples are likely noisy, which makes DAIR-SQ more robust to noisy samples, as we already observed in Section 2.2.

2.4 The impact of partial augmentation

We explore the impact of partial augmentation, where we only augment a certain fraction of the training samples. The experiment revisits noiseless Rotated MNIST with weak rotation data augmentation and Colored MNIST with Adversarial augmentation. This experiment emulates situations where an augmentation function is only applicable to certain examples or where augmentation is expensive and we would like to decrease the augmentation cost.

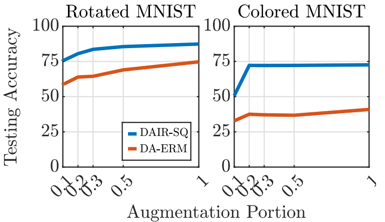

Figure 7: Test accuracy vs fraction of augmented samples on Rotated MNIST.

In Figure 7, we report the experiment results for DA-ERM and DAIR-SQ by applying augmentation only $\{10\%, 20\%, 30\%, 50\%, 100\%\}$ of the training samples, averaged on three runs. In Rotated MNIST experiment, as can be seen, DAIR-SQ with augmentation on only 20-30% of the samples performs similar to full augmentation. On the other hand, DA-ERM is more sensitive to partial augmentation and is subject to a steeper performance drop. This could be viewed as further evidence that DAIR-SQ could reach its best performance using weak augmenter functions. It is also noteworthy that in this example, DAIR-SQ with only 10% partial augmentation still outperforms DA-ERM with 100% augmentation. One can draw similar conclustion in the Colored MNIST experiment as only 10% augmentation gives comparable performance to full augmentation.

3 Experiments on real-world tasks

3.1 Robust Regression: Simultaneous domain shift and label noise

In this experiment, we consider a regression task to minimize the root mean square error (RMSE) of the predicted values on samples from the Drug Discovery dataset. The task is to predict the bioactivities given a set of chemical compounds (binary features). We follow the setup of Li et al. (2021) to introduce random noise to corrupt the targets. Furthermore, similar to Colored MNIST, we add a spurious binary feature to the original setup. At training time, the spurious feature is set to 1 if a particular target is above the median of the all the targets in the training samples, and 0 otherwise. At test time, this condition is reversed leading to poor generalization. We compare using ERM, DA-ERM and DAIR-SQ formulations under 0%, 20% and 40% noise levels on three baselines: $\mathcal{L}_2$ loss, Huber loss, and negatively tilted loss (Li et al., 2021), which is called tilted empirical risk minimization (TERM) and is designed for robust regression. For each of these baselines, we perform data augmentation by randomly assigning the spurious feature as 0 or 1 with equal probability. Finally, we apply the DAIR-SQ regularizer to each of these loss functions with $\lambda = 10$.

Algorithms	Test RMSE (Drug Discovery dataset)									
	0% Noise			20% Noise			40% Noise			Clean
	-	DA-	DAIR	-	DA-	DAIR	-	DA-	DAIR	-
$\mathcal{L}_2$ loss	1.97 (0.00)	1.36 (0.00)	**1.23** (0.00)	4.33 (0.04)	2.52 (0.05)	2.04 (0.06)	5.30 (0.04)	3.47 (0.07)	2.99 (0.09)	1.23 (0.00)
Huber (Huber, 1964)	1.84 (0.00)	1.27 (0.00)	1.24 (0.00)	2.93 (0.05)	1.50 (0.02)	1.39 (0.02)	4.40 (0.07)	2.18 (0.04)	1.70 (0.05)	**1.16** (0.00)
TERM (Li et al., 2021)	1.74 (0.00)	1.26 (0.00)	1.25 (0.00)	1.87 (0.01)	**1.27** (0.01)	**1.27** (0.01)	2.01 (0.02)	**1.33** (0.01)	**1.31** (0.01)	1.23 (0.00)

Table 1: Test RMSE for varying degrees of label noise for ERM, DA-ERM, and DAIR using different losses.

The results of this experiment are reported in Table 1. In the last column of the table we report results on the clean dataset without any spurious features for comparison purposes. As can be seen, without data augmentation all methods fall prey to spurious features and perform poorly, especially as the noise level is increased. It is noteworthy that while TERM is not designed for domain shift, it slightly outperforms the other baselines in the presence of spurious features showing that TERM has some inherent robustness to the domain shift. By adopting data augmentation, testing error decreases but is still quite large as compared to the Clean ERM setup for high values of noise. Notably, DAIR is able to reduce the testing error across all objectives and noise levels with the gap between DAIR and other approaches increasing with the degree of noise. For the 0% noise setup, DAIR is able to almost recover the Clean ERM accuracy for all three objectives. The gains achieved with DAIR are prominent for $\mathcal{L}_2$ and Huber, but marginal for TERM. Finally, data augmentation/DAIR combined with TERM can simultaneously handle domain shift and noisy labels as can be seen in this table.

3.2 Invariant Visual Question Answering

Visual Question Answering (VQA) has diverse applications ranging from visual chatbots to assistants for the visually impaired. In such real-world settings, it is desirable for VQA models to be robust to variations in the input modalities. In this spirit, recent works (Agarwal et al., 2020; Shah et al., 2019; Ray et al., 2019) have studied the robustness and consistency of VQA models under linguistic and visual variations. In this paper, we focus on the InVariant VQA (IV-VQA) dataset which contains semantically edited images corresponding to a subset of the original images from VQA v2 (Goyal et al., 2017). For each image in this subset, IV-VQA contains one or more edited images constructed by removing an object which is irrelevant to answering the question. A robust model should be invariant to such edits by making the same predictions on the edited image.

We choose the attention based SAAA (Kazemi & Elqursh, 2017) model to match the original setup from Agarwal et al. (2020). Using DAIR, we enforce consistency in predictions between the original and edited samples. Wherever the edited image is not available, the DAIR formulation reduces to ERM. We use the standard VQA accuracy along with the consistency metrics proposed in Agarwal et al. (2020) to compare our results against the ERM setup and the DA-ERM approach discussed in Agarwal et al. (2020).

The results are reported in Table 2. We measure the accuracy on the original VQA v2 'val' set and the consistency metrics across edited IV-VQA instances and their corresponding real instances from VQA v2 'val' set. The consistency metrics measure the three types of flips namely, pos → neg, neg → pos and neg → neg. A pos → neg flip indicates that the answer predicted with the original image was correct but was wrong with the corresponding edited image. A neg → neg flip indicates that the answer changes from original to edited image but is wrong for both. The accuracy of DAIR on the VQA v2 'val' set is higher as compared to others, while improving over all baselines by a minimum of **1.3%** under the 'Predictions flipped' metric which is the sum of the three types of flips. This improvement is significant given that the model needs to predict the answer correctly from 3000 candidate answers. While applying DAIR to this task, we observe a trade-off between the VQA accuracy on 'val' and the 'Predictions flipped' percentage controlled by the λ parameter. By increasing λ, the 'Predictions flipped' percentage decreases, and drops to as low as 7-8% when λ is at 10, albeit sacrificing the VQA accuracy by 5-6%. Thus, for moderate values of λ, DAIR is able to maintain the predictive power while enforcing consistency across variations in the visual space.

Algorithm	ERM (%) (Kazemi & Elqursh, 2017)	DA-ERM (%) (Agarwal et al., 2020)	DAIR-SQ (%)
VQA v2 val	57.10	57.30	**57.54**
Predictions flipped	11.84	11.68	**10.37**
pos → neg	4.58	4.40	**3.80**
neg → pos	5.17	5.14	**4.65**
neg → neg	2.08	2.14	**1.91**

Table 2: Accuracy and Consistency metrics on VQA v2 val & IV-VQA test set.

3.3 Training Robust Deep Networks against Adversarial Attacks

In this section, we demonstrate that our regularizer can be applied to train robust neural networks and it achieves comparable or better results than baseline models from state-of-the-art approaches which are specifically designed for this task. In our approach, the augmented examples $\widetilde{z}$ can be generated by a certain strong attack, such as Projected Gradient Descent (PGD) (Madry et al., 2018) or CW (Carlini & Wagner, 2017).We conduct our experiments on CIFAR-10 dataset and compare our approach with several other state-of-the-art baselines.

The performance of our algorithm against FGSM and variants of PGD, is summarized in Table 3, which shows that our results are competitive with the baselines. We report the performance of DAIR-SQ in Table 3 based on the configurations that give the best Clean accuracy (row 3) and the best Robust accuracy against PGD20 (row 6). The trade-off curve shown in Figure 8 suggests that by sweeping the value of λ, DAIR-SQ can achieve a better clean accuracy but a slightly lower PGD20 accuracy, and dominates most of the baseline, while it achieves a similar performance with TRADES. Note that the formulation in TRADES is equivalent to consistency regularization with KL divergence between the logits of the original and adversarial images. As opposed to our setup, the regularizer term in TRADES is also used in solving the maximization problem to generate adversarial images, whereas we only use the original loss for generating the adversarial examples.

We also report the accuracy consistency metric (CM) in this experiment in Table 3. CM captures the consistency of accuracy on PGD20 attack compared to clean examples. We observe that DAIR-SQ outperforms all baselines, which is in line with its best generalization to different attacks.

#	Algorithm	Clean (%)	FGSM (%)	PGD20 (%)	CM (%)
1	PGD Training (Madry et al., 2018)	82.89	55.38	48.40	–
2	APART (Li et al., 2020)	82.45	55.33	48.95	60.05
3	DAIR-SQ ($\lambda = 6$)	83.04	57.57	50.68	62.66
4	TRADES + ATTA (Zheng et al., 2020)	78.98	55.58	52.30	60.56
5	TRADES (Zhang et al., 2019)	81.67	57.78	52.90	63.14
6	DAIR-SQ ($\lambda = 16.7$)	81.29	58.58	53.37	67.51

Table 3: CIFAR-10 test accuracies under no attack (clean), FGSM, and PGD20 attacks, and accuracy consistency metric between original and PGD20 attack.

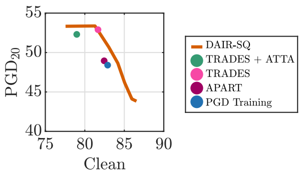

Figure 8: PGD20/Clean Acc. trade-off by sweeping λ.

3.4 Neural Task-Oriented Dialog Modeling

Virtual digital assistants that engage in conversations with human users are rapidly gaining popularity. These devices require the modelling of task-oriented dialog systems that can communicate with users through natural language to accomplish a wide range of tasks. One of the main objectives in task-oriented dialog systems is the Dialog State Tracking (DST), which refers to keeping track of the user goals as the conversation progresses. Among task-oriented dialog datasets, MultiWOZ (Budzianowski et al., 2018) has gained the most popularity owing to the availability of 10k+ realistic dialogs across 8 different domains, and has been improved several times (Wu et al., 2019; Eric et al., 2019; Zang et al., 2020; Han et al., 2021; Qian et al., 2021).

Recently, SimpleTOD (Hosseini-Asl et al., 2020) achieved state-of-the-art results on MultiWOZ using a neural end-to-end modeling approach. However, Qian et al. (2021) observed that the performance of SimpleTOD drops significantly when the test set named entities (which are places in the UK) are replaced with new ones never observed during training (with new entities all based in the US), perhaps due to the memorization of named entities during training. We leverage DAIR-SQ to promote invariance of the dialog policy to named entities in the dialog flow. Here, the data augmentation scheme is a simple one. We replace named entities in the training set with their randomly scrambled version. For example, “cambridge” could be turned into “bmcedrgia.” Details on training data, augmentation schemes and hyper-parameters can be found in Appendix H.

The results are presented in Table 4, where performance is measured in Joint Goal Accuracy (JGA). JGA is a binary metric, and is equal to 1 if the predictions of all dialog states in a turn are correct. As such it is a difficult metric to get right too. As can be seen, both DA-ERM and DAIR outperform SimpleTOD (Hosseini-Asl et al., 2020) on MultiWOZ 2.2 w/ SGD entities (Qian et al., 2021). Perhaps, more surprisingly, DAIR also outperforms SimpleTOD on the original MultiWOZ 2.2 test set with no distribution shift, which we attribute to better robustness to the named entity memorization problem observed by Qian et al. (2021). Finally, we also observe that DAIR significantly improves the JGA consistency metric compared to the DA-ERM baseline.

	MultiWOZ 2.2 Test JGA	MultiWOZ 2.2 Test JGA w/ SGD entities	CM
SimpleTOD (Hosseini-Asl et al., 2020)	0.5483	0.4844	–
SimpleTOD (DA-)	0.5915 (0.0055)	0.5311 (0.0074)	0.8354
SimpleTOD (DAIR)	0.5998 (0.0030)	0.5609 (0.0074)	0.8902

Table 4: Joint Goal Accuracy (JGA) for different approaches on the SimpleTOD model. DAIR achieves state-of-the-art results on the original MultiWOZ 2.2 test set (Zang et al., 2020) and well as the MultiWOZ 2.2 test set w/ named entities replaced with SGD (Qian et al., 2021).

4 Conclusion

In this paper, we proposed a simple yet effective consistency regularization technique, called data augmented invariant regularization (DAIR). DAIR is applicable when data augmentation is used to promote performance invariance across pairs of original and augmented samples, and it enforces the loss to be similar on the original and the augmented samples. As such, DAIR requires access to pairs of original and augmented examples. We also provided motivation and justification for DAIR, and particularly showed that it can recover the optimal solution in a certain regression task where data augmentation alone is insufficient. We also compared DAIR with several other consistency regularizers on several toy problems and showed that it is more stable and results in better performance. We empirically evaluated DAIR in four real-world machine learning tasks, namely robust regression, invariant visual question answering, training robust deep neural networks, and task-oriented dialog modeling. This is a major benefit of DAIR as some of other consistency regularizers cannot be applied broadly. Empirically, DAIR performed well on all these tasks and set new state-of-the-art results in these benchmarks.

Several problems remain open for future research: An in-depth theoretical understanding of the properties of DAIR that lead to its superior empirical performance on broad applications is an important open question. Further, automated hyperparameter tuning techniques for the strength of the regularizer is another avenue for future research. Finally, while we showed that DAIR boosts existing performance metrics, such as accuracy, the interplay of DAIR with other metrics, especially group fairness, is another important area for future research.

REFERENCES

Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards causal VQA: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27:3365–3373, 2014.

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.

Benjamin Bloem-Reddy and Yee Whye Teh. Probabilistic symmetries and invariant neural networks. *J. Mach. Learn. Res.*, 21:90–1, 2020.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017.

Yining Chen, Colin Wei, Ananya Kumar, and Tengyu Ma. Self-training avoids using spurious features under domain shift. *arXiv preprint arXiv:2006.10032*, 2020.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Logan Engstrom, Andrew Ilyas, and Anish Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv:1807.10272*, 2018.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*, 2019.

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.

Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. *arXiv preprint arXiv:2104.09459*, 2021.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

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

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *ICLR*, 2020.

Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. Multiwoz 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks, 2016.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. A simple language model for task-oriented dialogue. *NeurIPS*, 2020.

Peter J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35: 492–518, 1964.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

V. Kazemi and A. Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *ArXiv*, abs/1704.03162, 2017.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (REx). In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5815–5826. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/krueger21a.html`.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018a.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018b.

Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted empirical risk minimization. *ICLR*, 2021.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 624–639, 2018c.

Zichao Li, Liyuan Liu, Chengyu Dong, and Jingbo Shang. Overfitting or underfitting? understand robustness drop in adversarial training, 2020.

Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32:6665–6675, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston. Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*, 2017.

Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8690–8699, 2021.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1406–1415, 2019.

Kun Qian, Ahmad Beirami, Zhouhan Lin, Ankita De, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. Annotation inconsistency and entity bias in MultiWOZ. *The 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, July 2021.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset, 2020.

Arijit Ray, Karan Sikka, Ajay Divakaran, Stefan Lee, and Giedrius Burachas. Sunny and dark outside?! improving answer consistency in vqa through entailed question generation. In *EMNLP/IJCNLP*, 2019.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL `https://aclanthology.org/2020.acl-main.442`.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *2019 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Samarth Sinha and Adji B Dieng. Consistency regularization for variational auto-encoders. *arXiv preprint arXiv:2105.14859*, 2021.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

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

Christopher Tensmeyer and Tony Martinez. Improving invariance and equivariance properties of convolutional neural networks. 2016.

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.

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *arXiv preprint arXiv:1805.12018*, 2018.

Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *arXiv preprint arXiv:2106.04619*, 2021.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. Transferable multi-domain state generator for task-oriented dialogue systems, 2019.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10687–10698, 2020.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

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

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. Multiwoz 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines, 2020.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Haizhong Zheng, Ziqi Zhang, Juncheng Gu, Honglak Lee, and Atul Prakash. Efficient adversarial training with transferable adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13025–13032, 2020.

A Proofs

Proof of Proposition 1. First let us present the DA-ERM solution:

$$\begin{aligned}
f_{\text{DA-ERM}}(w) =& \mathbb{E}\left[(w_1x + w_2y - y)^2 + (w_1x + w_2(ay+n) - y)^2\right] && (1)\\
=& \mathbb{E}\left[w_1^2x^2 + (w_2-1)^2y^2 + 2w_1(w_2-1)xy\right]\\
&+ \mathbb{E}\left[w_1^2x^2 + (w_2a-1)^2y^2 + w_2^2n^2\right]\\
&+ \mathbb{E}\left[2w_1(w_2a-1)xy + 2w_1w_2xn + 2w_2(w_2a-1)yn\right] && (2)\\
=& w_1^2\sigma_x^2 + (w_2-1)^2(\sigma_x^2+\sigma_\varepsilon^2) + 2w_1(w_2-1)\sigma_x^2\\
&+ w_1^2\sigma_x^2 + (w_2a-1)^2(\sigma_x^2+\sigma_\varepsilon^2) + w_2^2\sigma_n^2\\
&+ 2w_1(w_2a-1)\sigma_x^2 && (3)\\
=& (w_1+w_2-1)^2\sigma_x^2 + (w_2-1)^2\sigma_\varepsilon^2\\
&+ (w_1+w_2a-1)^2\sigma_x^2 + (w_2a-1)^2\sigma_\varepsilon^2 + w_2^2\sigma_n^2. && (4)
\end{aligned}$$

Hence, the solution of $w^\star_{\text{DA-ERM}} = \arg\min_w f_{\text{DA-ERM}}(w)$ is given by

$$\begin{aligned}
2w_1^\star + (1+a)w_2^\star - 2 = 0,&\\
(w_1^\star + w_2^\star - 1)\sigma_x^2 + (w_2^\star - 1)\sigma_\varepsilon + a(w_1^\star + w_2^\star a - 1)\sigma_x^2 + a(w_2^\star a - 1)\sigma_\varepsilon^2 + w_2^\star\sigma_n^2 = 0.& \quad (5)
\end{aligned}$$

Subsequently,

$$w^\star_{\text{DA-ERM}} = \begin{pmatrix} \frac{a^2(\sigma_x^2+\sigma_\varepsilon^2)-2a(\sigma_x^2+\sigma_\varepsilon^2)+\sigma_x^2+\sigma_\varepsilon^2+2\sigma_n^2}{a^2(\sigma_x^2+2\sigma_\varepsilon^2)-2a\sigma_x^2+\sigma_x+2(\sigma_\varepsilon^2+\sigma_n^2)} \\ \\ \frac{2(a+1)\sigma_\varepsilon^2}{a^2(\sigma_x^2+2\sigma_\varepsilon^2)-2a\sigma_x^2+\sigma_x^2+2(\sigma_\varepsilon^2+\sigma_n^2)} \end{pmatrix}. \quad (6)$$

$$\begin{aligned}
w^\star_{\text{DAIR}} &= \arg\min_w f_{\text{DAIR}}(w)\\
&= \arg\min_w \mathbb{E}\left[(w_1x + w_2y - y)^2 + (w_1x + w_2(ay+n) - y)^2\right]\\
&+ \left[\lambda(|w_1x + w_2y - y| - |w_1x + w_2(ay+n) - y|)^2\right].
\end{aligned}$$

When $\lambda \to \infty$, we have $w^\star_{\text{DAIR},2} = 0$ and hence:

$$w^\star_{\text{DAIR}} = \begin{pmatrix} 1 \\ \\ 0 \end{pmatrix}.$$

We then evaluate the testing loss assuming the spurious feature is absent, i.e., $\mathbf{x}_{\text{test}} = (x, s = 0)$.

$$\begin{aligned}
\ell_{\text{DAIR}}(\mathbf{x}_{\text{test}}; w^\star_{\text{DAIR}}) &= \mathbb{E}\left[({w^\star_{\text{DAIR}}}^\top \mathbf{x}_{\text{test}} - y)^2\right]\\
&= \mathbb{E}\left[(x - (x+\varepsilon))^2\right]\\
&= \sigma_\varepsilon^2.
\end{aligned}$$

$$\begin{aligned}
\ell_{\text{DA-ERM}}(\mathbf{x}_{\text{test}}; w^\star_{\text{DA-ERM}}) &= \mathbb{E}\left[({w^\star_{\text{DA-ERM}}}^\top \mathbf{x}_{\text{test}} - y)^2\right]\\
&= \mathbb{E}\left[\left(\frac{a^2(\sigma_x^2+\sigma_\varepsilon^2) - 2a(\sigma_x^2+\sigma_\varepsilon^2) + \sigma_x^2 + \sigma_\varepsilon^2 + 2\sigma_n^2}{a^2(\sigma_x^2+2\sigma_\varepsilon^2) - 2a\sigma_x^2 + \sigma_x + 2(\sigma_\varepsilon^2+\sigma_n^2)}x - (x+\varepsilon)\right)^2\right]\\
&= \sigma_\varepsilon^2 + \frac{(a+1)^4\sigma_\varepsilon^4\sigma_x^2}{(a^2(\sigma_x^2+2\sigma_\varepsilon^2) - 2a\sigma_x^2 + \sigma_x + 2(\sigma_\varepsilon^2+\sigma_n^2))^2}\\
&\geq \ell_{\text{DAIR}}.
\end{aligned}$$

Proposition 2. *It is not hard to check that even using the weight decay regularizer $\frac{\gamma}{2}(w_1^2 + w_2^2)$ would not close the gap between the performance of DA-ERM and DAIR. In particular, this regularizer would result in*

$$w^\star_{\textit{DA-ERM-WD}} = \begin{pmatrix} \frac{a^2(\sigma_\varepsilon^2+\sigma_x^2)-2a(\sigma_\varepsilon^2+\sigma_x^2)+2\gamma+\sigma_\varepsilon^2+2\sigma_n^2+\sigma_x^2}{a^2(\gamma(\sigma_\varepsilon^2+\sigma_x^2)+2\sigma_\varepsilon^2+\sigma_x^2)-2a\sigma_x^2+\gamma^2+\gamma(\sigma_\varepsilon^2+\sigma_n^2+\sigma_x^2+2)+2\sigma_\varepsilon^2+2\sigma_n^2+\sigma_x^2} \\ \\ \frac{(a+1)(\gamma(\sigma_\varepsilon^2+\sigma_x^2)+2\sigma_\varepsilon^2)}{a^2(\gamma(\sigma_\varepsilon^2+\sigma_x^2)+2\sigma_\varepsilon^2+\sigma_x^2)-2a\sigma_x^2+\gamma^2+\gamma(\sigma_\varepsilon^2+\sigma_n^2+\sigma_x^2+2)+2\sigma_\varepsilon^2+2\sigma_n^2+\sigma_x^2}) \end{pmatrix},$$

which is not equal to $w^\star = (1, 0)$ unless $\sigma_n^2 \to \infty$ and $\gamma = 0$.

Proof of Proposition 2. The proof follows the same idea of Proposition 1 and therefore it is omitted here.

Proof of Lemma 1. We proceed as follows:

$$\mathcal{R}_1(z, \widetilde{z}; \theta) - \mathcal{R}_{\text{sq}}(z, \widetilde{z}; \theta) = 2\sqrt{\min\{\ell(z;\theta), \ell(\widetilde{z};\theta)\}}\left|\sqrt{\ell(\widetilde{z};\theta)} - \sqrt{\ell(z;\theta)}\right|,$$

We break it into two cases: if $\ell(\widetilde{z};\theta) > \ell(z;\theta)$:

$$\begin{aligned} \mathcal{R}_1(z, \widetilde{z}; \theta) - \mathcal{R}_{\text{sq}}(z, \widetilde{z}; \theta) &= \ell(\widetilde{z};\theta) - \ell(z;\theta) - (\sqrt{\ell(\widetilde{z};\theta)} - \sqrt{\ell(z;\theta)})^2 \\ &= \ell(\widetilde{z};\theta) - \ell(z;\theta) - \ell(\widetilde{z};\theta) - \ell(z;\theta) + 2\sqrt{\ell(\widetilde{z};\theta)}\sqrt{\ell(z;\theta)} \\ &= -2\ell(z;\theta) + 2\sqrt{\ell(\widetilde{z};\theta)}\sqrt{\ell(z;\theta)} \\ &= 2\sqrt{\ell(z;\theta)}(\sqrt{\ell(\widetilde{z};\theta)} - \sqrt{\ell(z;\theta)}). \end{aligned}$$

If $\ell(\widetilde{z};\theta) \leq \ell(z;\theta)$:

$$\begin{aligned} \mathcal{R}_1(z, \widetilde{z}; \theta) - \mathcal{R}_{\text{sq}}(z, \widetilde{z}; \theta) &= \ell(z;\theta) - \ell(\widetilde{z};\theta) - (\sqrt{\ell(\widetilde{z};\theta)} - \sqrt{\ell(z;\theta)})^2 \\ &= \ell(z;\theta) - \ell(\widetilde{z};\theta) - \ell(\widetilde{z};\theta) - \ell(z;\theta) + 2\sqrt{\ell(\widetilde{z};\theta)}\sqrt{\ell(z;\theta)} \\ &= -2\ell(\widetilde{z};\theta) + 2\sqrt{\ell(\widetilde{z};\theta)}\sqrt{\ell(z;\theta)} \\ &= 2\sqrt{\ell(\widetilde{z};\theta)}(\sqrt{\ell(z;\theta)} - \sqrt{\ell(\widetilde{z};\theta)}). \end{aligned}$$

If we combine the two cases, we have:

$$\mathcal{R}_1(z, \widetilde{z}; \theta) - \mathcal{R}_{\text{sq}}(z, \widetilde{z}; \theta) = 2\sqrt{\min\{\ell(z;\theta), \ell(\widetilde{z};\theta)\}}\left|\sqrt{\ell(\widetilde{z};\theta)} - \sqrt{\ell(z;\theta)}\right|.$$

B Practical considerations when DAIR is used in training

In this section, we investigate the reason why we see a sweet spot for the performance of DAIR-SQ as a function of λ. As shown in Figure 2, we see a sweet spot for λ, where the performance takes its maximum and starts to decrease for larger values of λ. There are a few explanations for this performance degradation.

1. It is observed that a large λ requires a relatively longer time for convergence. To show empirically this is true, we added another example in Appendix B.1. Theoretically, this is in line with the classical results in the optimization literature where larger Lipschitz constants (resulting from adding a regularizer) slows down the convergence rate. Thus, as we are training all models for a certain number of epochs, we will end up with underfitting.
2. A larger λ is more likely to guide the optimization trajectory towards a spurious poor local minimum with poor generalization performance, when the optimization trajectory is non-convex. We have experimentally verified this in Section 2.3 (Figure 6) as the reason for the poor performance of DAIR-L1 in Figure 2.
3. With a finite number of samples our regularizer does not necessarily lead to the best possible performance in the infinite sample setting (with weak domain shift). Hence, we might expect to observe the classical approximation-estimation tradeoff. This is especially true in real-world scenarios where one might expect that the difficulty of the example may not necessarily be preserved through data augmentation, and hence forcing the loss to be equal on both samples might be detrimental to the overall performance, which may lead to a practical sweet spot for λ.

We dig into the experiment in Figure 2 specifically and try to understand which case is the responsible for the sweet spot in Figure 2. We extend the number of training epochs from 40 to 160, and report the accuracy for $\lambda \in \{1.43, 8.85, 16.23, 100\}$.[1] Table 5 suggests that, when we increase the number of training epochs, the sweet spot of λ moves from 8.85 to 16.23 and in fact we can achieve an even better performing model with accuracy 89.22 as compared to the previously reported 85.89, while the performance does not change much for the smaller values of λ. We also observe a big performance boost for larger values of λ. This suggests that in this experiment the sweet spot for λ is caused by capping the training epochs to a finite value. Having said that, we believe that we are practically interested in using DAIR with marginal computational overhead over ERM and hence we would expect to observe such sweet spot in performance in practice as $\lambda \to \infty$.

λ	Acc at Epoch 40	Acc at Epoch 160
1.43	79.09	80.19
8.85	**85.89**	86.60
16.23	82.66	**89.22**
100	46.95	69.37

Table 5: Testing accuracy of Rotated MNIST, Weak Augmentaion. We see the accuracy increases as we extend the number of training epochs.

B.1 Additional evidence on growing cost of training with the regularization strength

We also provide further evidence for the growing cost of training with λ on a toy problem where we can reliably measure the gradient norm and ensure convergence. We study the following simple binary logistic classification problem which mirrors the MNIST experiments: at the training time the input is $\mathbf{x}_{\text{train}} = (x, s = 2y - 1 + t_1)$ and the label y, i.e., $z_{\text{train}} = (\mathbf{x}_{\text{train}}, y)$. Here, $x \sim \mathcal{N}(0, \sigma_x^2)$, and $P(y = 1|x) = \frac{1}{1+e^{-x}}$, where t_1 is independent of x and $t_1 \sim \mathcal{N}(0, \sigma_1^2)$. In this example, we intentionally provide feature s which is highly correlated with the label during training. Again, clearly, $w^\star = (1, 0)^\top$, but $w^\star_{\text{ERM}}$ will converge to $(0, 1)^\top$ due to the overfitting to the spurious feature. We introduce an augmenter which generates the augmented example such as $\mathbf{x}_{\text{aug}} = (x, s = 2y - 1 + t_1 + t_2)$ where $t_2 \sim \mathcal{N}(0, \sigma_2^2)$. We use this data augmenter for DAIR training and test on

[1] Note these values comes from $\log_{10}$ sweeping of λ.

$\mathbf{x}_{\text{test}} = (x, s = 1 - 2y)$. We summarize the steps need for convergences and the testing accuracy in Table 6 as well. We can find that the required number of iteration to convergence increases as λ increase.

For this tiny toy example, there is a factor of 10x increase in the required number of iterations when λ is chosen to be 10,000 as opposed to 0.5. Note that this is using ADAM and the gap is significantly larger if we use vanilla gradient descent; as we were not able to even converge in 10^8 steps. This is provided as further evidence for the practical sweet spot for DAIR as $\lambda \to \infty$.

λ	Iterations to Converge
0.5	81.35 ± 6.07
1	91.05 ± 2.53
2	89.10 ± 2.41
5	101.65 ± 2.87
10	107.70 ± 5.77
100	151.75 ± 4.28
1,000	195.85 ± 4.54
10,000	802.60 ± 7.58

Table 6: Iteration needed for the logistic model to converge with different λ. The model is converged when the $\mathcal{L}_2$ norm of the gradient is less than 10^{-7}.

C Model Architecture and Training Parameters for MNIST Experiments

We use a Convolutional Neural Network (CNN) with three convolutional layers followed by two fully connected layers. The last layer output size for Colored MNIST experiments is set to 1, and 10 for the Rotated MNIST experiments. For training we follow a two stage schedule with a learning rate of 0.005 for the first 20 epochs and a learning rate of 0.0005 for the next 20. We choose a batch size of 64 for all experiments. The architectural details and training parameters can be found in Table 7 and Table 8.

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 7: Model Architecture, $C = 1$ for Colored MNIST and $C = 10$ for Rotated MNIST.

Parameter	Value	
Learning Rate	0.005	0.0005
Epochs	First 20	Second 20
Batch-size	64	

Table 8: Training parameter of MNIST experiments.

D Colored MNIST & Rotated MNIST Setup

We apply the proposed loss function (DAIR) on the following two datasets: Colored MNIST and 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.

Our 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 9.

Our 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 10.

Note all the augmented images are generated on the fly. Examples of images from some transformation schemes are shown in Figures 9 to 14.

Scheme	z	Color $\mid y = 0$
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 9: 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.

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

Setup Name	Train	Aug	Test	λ
Adv. Aug.	C1	C2	C3	1000
Rnd. Aug.	C1	C4	C3	100

Table 11: Training procedure of Colored MNIST.

Setup	Train	Aug	Test	λ
Strong Aug.	R1	R5	R2	1
Weak Aug.	R4	R6	R3	10

Table 12: Training procedure of Rotated MNIST

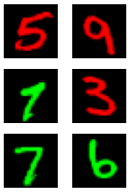

Figure 9: C2

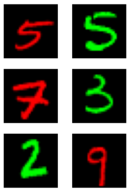

Figure 10: C3

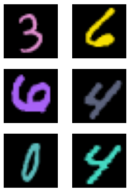

Figure 11: C4

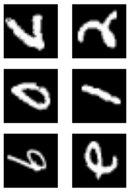

Figure 12: R4

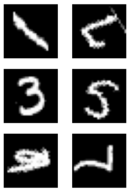

Figure 13: R5

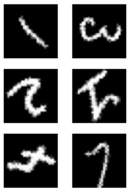

Figure 14: R6

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 9 (Colored MNIST) and Table 10 (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 11 (Colored MNIST) and Table 12 (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 C.

D.1 Colored MNIST

We conduct two sets of experiments for this dataset: Adversarial Augmentation Setup (Table 11) follows the exact same color schemes from the original Colored MNIST Arjovsky et al. (2019). For Random Augmentation Setup, we train the model with the strongest possible augmenter: uniformly random color. The entire procedure is summarized in Table 11.

D.2 Rotated MNIST

We start with the strongest augmenter case. 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 12.

E Additional Results on Colored MNIST & Rotated MNIST

E.1 Colored MNIST

We show additional results on Colored MNIST and Rotated MNIST in Tables 13 and 14. Note that each algorithm has been tuned for best performance. As mentioned in Section 2.2, DAIR outperforms DA-ERM, ERM and other baseline models on classification accuracy. For accuracy consistency, we use the training scheme as the original scheme and the testing scheme as the augmentation scheme. We further compare DAIR with IRM (Arjovsky et al., 2019), DRO (Sagawa et al., 2019), and REx (Krueger et al., 2021). In doing so, we feed all original examples as one environment and all augmented examples as a second environment to these baselines. While we can see that DAIR outperforms all baselines, we caution that the comparison may not be fair in that DAIR exploits pairing information between original and augmented samples, which is not used by the other baselines.

Algorithm	Accuracy	CM
ERM	32.70 ± 0.45	77.76 ± 1.01
DA-ERM	40.91 ± 0.45	84.60 ± 0.60
DAIR	72.58 ± 0.11	99.39 ± 0.11
IRM (Arjovsky et al., 2019)	66.90	–
DRO (Sagawa et al., 2019)	37.40	–
REx (Krueger et al., 2021)	68.70	–

Table 13: Accuracy and Accuracy Consistency Metric (CM) on Colored MNIST with Adversarial Augmentation.

Algorithm	Accuracy	CM
ERM	32.70 ± 0.45	63.50 ± 1.92
DA-ERM	29.61 ± 0.80	88.15 ± 0.18
DAIR	73.10 ± 0.12	99.88 ± 0.01

Table 14: Accuracy and Accuracy Consistency Metric (CM) on Colored MNIST with Random Augmentation.

E.2 Rotated MNIST

We report the accuracy consistency on Rotated MNIST (weak augmentation) in Table 15. The original training scheme here is Scheme R4 (Table 10), i.e., 90° and 270° rotated images, and the augmentation scheme for training is R6 (weak rotation). At test time, we test with R1 (no rotation) and we also use the augmentation scheme of 180° rotation to test the accuracy consistency metric. Note that neither the un-rotated or 180° rotated images have been observed at training time. Hence, the setup is difficult for ERM which struggles to generalize. As can be seen, since the digit 0 is "almost" circularly symmetric, ERM actually does a decent job at classifying 0, however it significantly struggles with all other digits. We see that DAIR outperforms ERM and DA-ERM by a large margin. We observe that digits 6 and 9 are challenging to get right (as one would expect for them to be difficult to tell apart). While we see $2 - 3\%$ drop on the consistency for digits 6 and 9 (when rotating them by 180°), the drop is smaller than expected perhaps due to the fact that the neural network learns to classify these digits based on features that are harder to get for humans.

Digit	ERM		DA-ERM		DAIR	
	Acc.	CM	Acc.	CM	Acc.	CM
0	86.19 ± 01.48	94.95 ± 01.53	95.61 ± 00.66	98.43 ± 00.21	98.44 ± 00.07	99.31 ± 00.15
1	00.15 ± 00.08	11.11 ± 11.11	82.79 ± 03.38	98.54 ± 00.43	96.09 ± 00.71	97.59 ± 01.28
2	29.84 ± 00.51	57.91 ± 02.76	76.68 ± 03.54	82.70 ± 03.27	86.21 ± 00.82	93.21 ± 01.32
3	00.63 ± 00.53	76.47 ± 23.53	78.84 ± 02.60	89.24 ± 01.26	86.60 ± 02.24	94.26 ± 00.36
4	01.97 ± 00.90	23.38 ± 13.49	51.09 ± 03.30	78.15 ± 02.73	79.67 ± 01.26	92.42 ± 00.41
5	05.53 ± 00.32	39.91 ± 04.59	65.02 ± 02.42	84.68 ± 03.71	83.26 ± 02.51	95.11 ± 01.46
6	00.66 ± 00.37	51.79 ± 25.13	67.43 ± 03.82	83.41 ± 05.74	84.79 ± 01.17	92.78 ± 01.71
7	16.67 ± 02.75	18.28 ± 06.65	56.29 ± 07.26	81.67 ± 06.90	78.11 ± 02.10	95.03 ± 01.21
8	10.92 ± 05.47	22.54 ± 05.46	74.50 ± 01.10	89.12 ± 01.69	90.55 ± 01.13	95.35 ± 00.47
9	17.08 ± 07.70	11.56 ± 00.62	69.54 ± 04.18	86.78 ± 01.08	80.84 ± 01.18	93.21 ± 01.39
All	16.85 ± 1.08	64.14 ± 2.69	71.98 ± 1.70	88.28 ± 0.27	86.57 ± 0.55	94.98 ± 0.29

Table 15: Rotated MNIST with 90° or 270° rotated original images and Weak Augmentation during training. The test scheme is un-rotated original images. Consistency metric (CM) is computed between un-roated images and ones with 180° rotation. It can be seen that CM is relatively small for 6 and 9 but the drop is smaller than expected suggesting that CNNs learn from features different from how humans perceive the digits.

F Setup and Additional Results for Visual Question Answering

All the approaches included in this paper use the original VQA v2 'train' split for training, along with the IV-VQA 'train' split for augmentation in the DAIR and DA-ERM(Agarwal et al., 2020) settings. The ERM setup (Kazemi & Elqursh, 2017), represents a vanilla SAAA model trained on the VQA v2 'train' split. For the data augmentation methods, if an image from VQA v2 contains its corresponding edited versions in IV-VQA, we randomly select one of them to serve as an augmentation during training. We modify the official code released by Agarwal et al. (2020) to suit our formulation. All the methods are trained for 40 epochs with a learning rate of 0.001 and a batch size of 48. The baseline approaches that we compare with are trained and evaluated by us, using the same training setup as DAIR.

λ	VQA v2 val (%)	Predictions flipped (%)	pos → neg (%)	neg → pos (%)	neg → neg (%)
0.37	**58.52**	11.92	4.48	5.28	2.17
0.72	58.21	11.28	4.13	5.08	2.07
1.39	57.54	10.37	3.80	4.65	1.91
2.68	56.24	9.68	3.56	4.39	1.73
5.18	54.19	8.75	3.40	3.66	1.69
10	51.32	**7.94**	**3.01**	**3.40**	**1.53**

Table 16: Accuracy-Consistency Tradeoff on VQA v2 val and IV-VQA test set controlled by λ

Table 16 indicates a tradeoff between the accuracy on the VQA v2 'val' set and the consistency metrics. As the λ value increases, the consistency between the predictions increases, while the accuracy on original examples decreases. For instance, A λ value of 10 strongly boosts consistency thus lowering the 'Predictions flipped' percentage to only 7.9% but sacrifices the predictive power causing the accuracy to drop to 51.3%.

G Details on Training Robust Neural Networks

For all algorithms reported in Table 3, we use Pre-Activation ResNet-18 (He et al., 2016), with a last-layer output size of 10 as the classification model. For training the DAIR model, the adversarial examples are generated by $\mathcal{L}_\infty$ based PGD attack with 11 iterations, ε (attack strength) set to 8/255 and attack step size to 2/255. We evaluate all the models against the standard FGSM attack and PGD attack with 20 iterations of same perturbation sizes.

H Details on Neural Task-Oriented Dialog Modeling

We provide details on the benchmark that we used in this experiment. Qian et al. (2021) proposed a new test set for MultiWOZ 2.2, called MultiWOZ 2.2 with SGD entities, where named entities are replaced with those from Schema Guided Dialog dataset (Rastogi et al., 2020) and showed that SimpleTOD (Hosseini-Asl et al., 2020) endures more than 8% performance drop on the new test set. Examples from the dataset are shown in Table 18. To address this problem, we define a new data augmentation scheme for DAIR and DA-ERM by replacing the named entities from the MultiWOZ 2.2 training set with randomly scrambled versions of the named entities. For example, "warkworth house" could be turned into "easrtokow hhrwu" (see Table 18). In all of our experiments, we utilize the SimpleTOD model (Hosseini-Asl et al., 2020) and we apply DAIR to enforce invariance between the named entities in the training examples and the scrambled entities from their corresponding augmented samples. The model is trained with ParlAI (Miller et al., 2017) fine-tuned with the pre-trained BART (Lewis et al., 2019). Training hyper-parameters can be found in Table 17.

Parameter	Value
λ	0.5
Epochs	4
Batchsize	6
Optimizer	AdamW
Learning rate	10^{-5}

Table 17: Hyper-parameters used in training SimpleTOD.

User:	can you help me book a reservation at the warkworth house hotel?
Agent:	yes i could! how many people are staying, and what days would fyou like to stay?
User:	it's just for me, and i'll be staying for three nights starting from tuesday.
DS:	***hotel-bookday:*** *tuesday* ***hotel-bookpeople:*** *1* ***hotel-bookstay:*** *3* ***hotel-name:*** *warkworth house*

User:	can you help me book a reservation at the easrtokow hhrwu hotel?
Agent:	yes i could! how many people are staying, and what days would fyou like to stay?
User:	it's just for me, and i'll be staying for three nights starting from tuesday.
DS:	***hotel-bookday:*** *tuesday* ***hotel-bookpeople:*** *1* ***hotel-bookstay:*** *3* ***hotel-name:*** *easrtokow hhrwu*

User:	can you help me book a reservation at the clarion inn & suites atlanta downtown hotel?
Agent:	yes i could! how many people are staying, and what days would fyou like to stay?
User:	it's just for me, and i'll be staying for three nights starting from tuesday.
DS:	***hotel-bookday:*** *tuesday* ***hotel-bookpeople:*** *1* ***hotel-bookstay:*** *3* ***hotel-name:*** *clarion inn & suites atlanta downtown*

Table 18: Left: sample from the original MultiWOZ dataset. Middle: augmented sample generated by scrambling. Right: synthetic sample with name entities from SGD. Comparing left and the middle example, we are generating new named entities (marked in red) by scrambling. Comparing left and the right example, the only difference is the named entity from different dataset, which is marked in red. Note that the SGD named entities are not exposed to the model during training. Only the original named entities and scrambled named entities from MultiWOZ are used during training.