Learning Unforeseen Robustness from Out-of-distribution Data Using Equivariant Domain Translator

Sicheng Zhu [1] **Bang An** [1] **Furong Huang** [1] **Sanghyun Hong** [2]

Abstract

Current approaches for training robust models are typically tailored to scenarios where data variations are accessible in the training set. While shown effective in achieving robustness to these foreseen variations, these approaches are ineffective in learning *unforeseen* robustness, i.e., robustness to data variations without known characterization or training examples reflecting them. In this work, we learn unforeseen robustness by harnessing the variations in the abundant out-of-distribution data. To overcome the main challenge of using such data, the domain gap, we use a domain translator to bridge it and bound the unforeseen robustness on the target distribution. As implied by our analysis, we propose a two-step algorithm that first trains an equivariant domain translator to map out-of-distribution data to the target distribution while preserving the considered variation, and then regularizes a model's output consistency on the domain-translated data to improve its robustness. We empirically show the effectiveness of our approach in improving unforeseen and foreseen robustness compared to existing approaches. Additionally, we show that training the equivariant domain translator serves as an effective criterion for source data selection.

1. Introduction

A trustworthy machine learning system should provide consistent output despite nuisance transformations in input. For instance, a self-driving car should consistently recognize road objects, regardless of viewpoint changes that do not alter the object's label. This desirable property of a model is measured by robustness — a hallmark feature exhibited by humans and numerous other creatures (Tacchetti et al., 2018). Training a model to be robust not only improves its trustworthiness but may also improve its in-distribution (Zhou et al., 2022) and out-of-distribution (OOD) generalization (Hendrycks et al., 2020), potentially by expanding the labeled region (Wei et al., 2021).

Recognizing the importance of robustness, previous work has proposed several methods to train robust models (a.k.a. robustness interventions), including training on augmented data (Sohn et al., 2020), consistency regularization (Xie et al., 2020), adversarial training (Madry et al., 2018), and architecture modifications (Zhang, 2019). These methods effectively improve robustness against *foreseen* data variations — those that can be characterized by known transformation functions or observable in pairs of training examples before and after the transformation, such as noise corruption (Hendrycks & Dietterich, 2019) and spatial transformations (Engstrom et al., 2019).

Nevertheless, *unforeseen* robustness remains challenging to achieve, with existing methods either unable or struggling to learn it. This issue is particularly problematic given that in many datasets, only specific synthetic data variations are foreseen, while others, including most natural variations, are not. As a result, models remain vulnerable to these unforeseen data variations, such as changes in viewpoint (Koh et al., 2021) or time (Shankar et al., 2021).

This work introduces a method to learn unforeseen robustness. Notably, the data variation unforeseen from a given training set is often observable as pairs of transformed examples in the abundant out-of-distribution data, such as simulations. Leveraging this observation, we propose to learn unforeseen robustness from out-of-distribution data, using an equivariant domain translator to bridge the domain gap while preserving the variation, as illustrated in Figure 1.

Contributions: *First*, we formulate the problem of learning unforeseen robustness from out-of-distribution data (§3.1) and identify the difficulties in extending existing approaches to solve this problem (Figure 1).

Second, recognizing that the primary challenge of this problem stems from the domain gap, we analyze the problem

[1]University of Maryland, College Park [2]Oregon State University. Correspondence to: Sicheng Zhu <sczhu@umd.edu>, Bang An <bangan@umd.edu>, Furong Huang <furongh@umd.edu>, Sanghyun Hong <sanghyun.hong@oregonstate.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

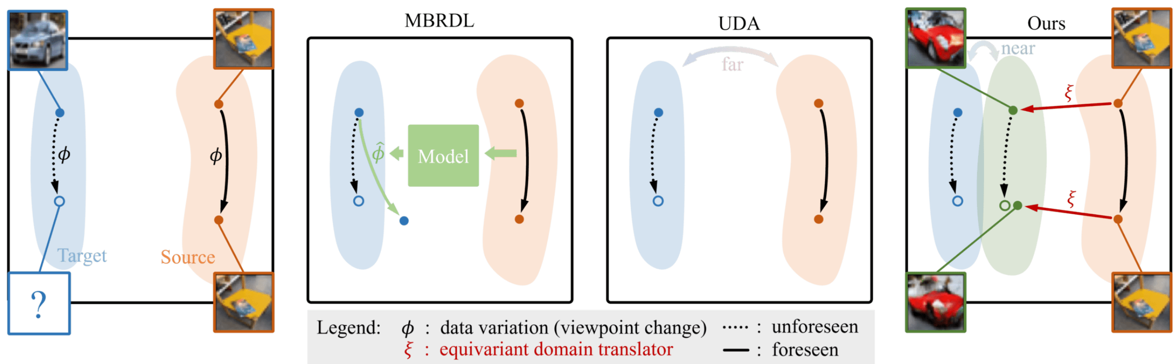


Figure 1. Illustration of our method and two existing methods *extended* to our setting. Our goal here is to learn the unforeseen (3D viewpoint change) robustness on the target dataset (CIFAR-10, depicted in blue). To this end, we find some out-of-distribution source data that contains example pairs exhibiting this variation (Objectron, a set of video clips showing viewpoint changes, depicted in orange) and learn robustness from them. **MBRDL** (model-based robust deep learning, Robey et al. (2020)) learns an auxiliary model to capture the variation in source data and then applies it to augment the target data. However, the auxiliary model encounters difficulties in generalizing across large domain gaps and in modeling complex data variations with multi-modal distributions. **UDA** (unsupervised data augmentation, Xie et al. (2020)) learns robustness directly on the source data. However, robustness learned from source data does not generalize well to the target due to the domain gap. In contrast, **our method** trains an equivariant domain translator to make source data resemble the target while preserving the variation, and then learns robustness from the translated data (depicted in green). The paired images outlined in green are generated by our trained domain translator.

with an auxiliary domain translator bridging the gap (§3.2). By considering a domain translator, i.e., a map from the input space to itself, we establish an upper bound for the robustness loss on the target distribution in terms of variations on the source distribution. In particular, this bound can be tightened by a domain translator that has two properties: *equivariant*, meaning that transforming an example first and then domain-translating it yields similar output as domain-translating the example first and then transforming it, and *accurate*, meaning that the domain-translated source distribution closely aligns with the target distribution in terms of the Wasserstein-1 distance.

Third, we propose a two-step algorithm for solving the problem based on our prior analysis (§4). The first step trains an equivariant and accurate domain translator. To make it accurate, we train the translator under the supervision of a Lipschitz-regularized domain discriminator, following WGAN (Arjovsky et al., 2017). To make it equivariant, we offer three optional regularization losses and choose one depending on our knowledge of the transformation function and the transformation parameter associated with each pair of transformed examples. The second step uses consistency regularization on the domain-translated source data to improve a model's robustness.

Fourth, we empirically evaluate our method for image classification tasks on a combination of seven source datasets, two target datasets, and two types of data variations (§5). We first verify that our method indeed learns equivariant and accurate domain translators. Then, we show the effectiveness of our method in learning unforeseen robustness compared to other baselines, and further support it by ablation studies. As a by-product, we also show that the training result of the equivariant domain translator correlates strongly (R=0.91) with the robustness benefit of a certain source dataset, indicating its usefulness as a source dataset selection criterion.

Fifth, we apply our method to two real-world tasks to demonstrate its practical significance. First, we learn the 3D viewpoint change robustness on CIFAR-10 by harnessing variations in video clips and show the improvement using surrogate transformations. Second, we show that our method can leverage out-of-distribution data to further improve the foreseen robustness on the target, effectively serving as a generalized and improved unsupervised data augmentation method. Our method achieves better improvements in robustness, in-distribution generalization, and out-of-distribution generalization compared to the previous method (Xie et al., 2020). We will make our code publicly available at https://github.com/schzhu/unforeseen-robustness.

2. Related Work

Semi-supervised consistency regularization. A large body of work uses consistency regularization for semi-supervised learning (Sohn et al. (2020)), achieving state-of-the-art results in generalization. The key idea is to do supervised learning on the labeled data while regularizing the model to predict consistently on the unlabeled data, which potentially expands the labeled region and thus improves generalization (Wei et al., 2021). Despite the various goals previous work has, such as improving generalization (Sohn et al., 2020) or improving adversarial robustness (Zhang et al., 2019; Alayrac et al., 2019; Carmon et al., 2019; Deng et al., 2021), there is no work, to our knowledge, that learns unforeseen robustness from out-of-distribution (OOD) data. Indeed, the OOD data with potentially disjoint label sets in our setting pose a unique challenge that invalidates many common techniques such as pseudo-labeling. To harness OOD data, previous work assumes some overlaps of label sets (i.e., open-set setting, see Saito et al. (2021)) and then filters out "irrelevant" data (Xie et al., 2020; Huang et al., 2022). In contrast, overlapping label sets are not necessary for learning robustness in our setting, so we can make use of any OOD data with the desired variation.

Model-based data augmentation. Another line of work uses generative models to capture class-agnostic data variations in the dataset and then apply the trained model to do input-conditioned data augmentation for better robustness and generalization (Antoniou et al., 2017; Robey et al., 2020; Zhou et al., 2022). Modeling the variation directly from OOD data and then applying the model to the target data encounters two major difficulties. First, while the class-agnostic data variations by assumption generalize across classes and domains, the generative model capturing them may not, confining previous work to train and apply the model on the same or similar dataset. If the domain gap is large, this method can even hurt the generalization of downstream classifier. In contrast, our domain translator is trained on and applies only to the existing OOD examples, thus avoiding this issue. Second, using a GAN-based generative model to capture highly multimodal natural variations faces intrinsic challenges (Tanielian et al., 2020; Salmona et al., 2022). Indeed, prior work showed its limitation to capture geometric transformations like rotation (Zhou et al., 2022). Our method addresses this challenge by relying on the ground-truth variations from the source data, resulting in target-like rotated images as shown in the experiment.

Neural style transfer. Our approach to using a domain translator that maps source images to approximate the target distribution, is related to neural style transfer (Gatys et al., 2015; Johnson et al., 2016; Huang et al., 2018; Isola et al., 2017; Zhu et al., 2017). The similar image-to-image translation process allows us to take advantage of this rich literature and adapt various off-the-shelf network architectures to implement our domain translator. However, the goals differ. Neural style transfer aims at transferring the style of a source image to a target one while preserving some content or the underlying label. In contrast, our domain translator does not need to preserve the content or label but requires equivariance to the data variation.

3. Problem Analysis

In this section, we first formulate the problem of learning unforeseen robustness by harnessing variations on source data, which has rarely been addressed before. To quantify the potential robustness achievable through learning from source data, we then establish an upper bound for the robustness loss on the target distribution in terms of variations on the source.

3.1. Problem: Robustness from Variations on Source

In this problem, we are given some target examples $\{\boldsymbol{x}_i\}$ sampled from the *target data distribution* $\mathbb{P}$ on the input space $\mathcal{X}$. We consider $\mathcal{X}$ to be $\mathbb{R}^d$ since we focus on image data. In addition, we are given some source examples $\{\boldsymbol{u}_i\}$ sampled from the *source data distribution* $\mathbb{Q}$ on $\mathcal{X}$. We do learning over a family of models $\{f : \mathcal{X} \to \mathbb{R}^k\}$ which map examples in $\mathcal{X}$ to k-dimensional output vectors. For classification tasks, we consider the model's logit output (before softmax) as the model output.

Data variation. We consider data variations that can be represented by some (possibly unknown) data transformation function $\phi : \mathcal{T} \times \mathcal{X} \to \mathcal{X}$, where $\mathcal{T}$ is the set of possible transformation parameters. Some examples are noise corruption, group actions with $\mathcal{T}$ being a group (e.g. flipping), and 3D viewpoint changes projected to the 2D pixel space (given that ϕ models the stochasticity). As we focus on random data transformation, we also consider some transformation parameter distribution $\mathbb{T}$ on $\mathcal{T}$. We assume that the data variation is unforeseen, meaning that we neither know the data transformation function nor have transformed target example pairs $\{(\boldsymbol{x}_i, \phi_{\boldsymbol{t}_i}(\boldsymbol{x}_i))\}$. Instead, given the source examples $\{\boldsymbol{u}_i\}$, we have finite (e.g., variations extracted from video clips) or infinite (e.g., simulated data) transformed versions $\{\phi_{\boldsymbol{t}_{ij}}(\boldsymbol{u}_i)\}$, where $\boldsymbol{t}_{ij}$ is sampled from $\mathbb{T}$.

Robustness. We consider model robustness to random data transformations. To measure the consistency of two model outputs, we use some loss function $\ell : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}_{\geq 0}$ that satisfies the triangle inequality $\ell(\boldsymbol{v}, \boldsymbol{v}'') \leq \ell(\boldsymbol{v}, \boldsymbol{v}') + \ell(\boldsymbol{v}', \boldsymbol{v}''), \forall \boldsymbol{v} \in \mathbb{R}^k$. Examples of such loss functions include the zero-one loss $\ell_{0\text{-}1}(\boldsymbol{v}, \boldsymbol{v}') = \mathbf{1}\{\arg\max_i \boldsymbol{v}_i \neq \arg\max_i \boldsymbol{v}'_i\}$, the ℓ_p loss $\ell_p(\boldsymbol{v}, \boldsymbol{v}') = \|\boldsymbol{v} - \boldsymbol{v}'\|_p$ for some $p \geq 1$, and certain f-divergences like the square root of JS-divergence (Endres & Schindelin, 2003). Given such a loss function, we define the following robustness loss.

Definition 3.1 (Robustness loss)**.** *Let ϕ be some transformation function and $\mathbb{T}$ be the distribution of transformation parameters. Then the robustness loss of a model f on the data distribution $\mathbb{P}$ is defined as*

$$L_\phi(f, \mathbb{P}) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}, \boldsymbol{t} \sim \mathbb{T}} \left[\ell\big(f(\boldsymbol{x}), f(\phi_{\boldsymbol{t}}(\boldsymbol{x}))\big) \right] \tag{3.1}$$

Note that the robustness loss is label-agnostic, making it well-defined on domains with different label sets. Similar notions of robustness also appear in the literature (e.g., Hendrycks & Dietterich (2019) and Zhou et al. (2022)).

Goal. Given the target examples $\{\boldsymbol{x}_i\}$, and the source examples $\{\boldsymbol{u}_i\}$ with their transformed versions $\{\phi_{\boldsymbol{t}_j}(\boldsymbol{u}_i)\}$, our goal is to learn a model f that minimizes the robustness loss on the target distribution $L_\phi(f, \mathbb{P})$ in addition to some other primary task loss (e.g., classification loss). We refer to this problem as learning robustness on the target data from variations on the source data.

For classification tasks, the significance of minimizing the robustness loss is that a small robustness loss along with a small classification loss $\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P},\boldsymbol{t}\sim\mathbb{T}}\left[\ell_{0\text{-}1}(\boldsymbol{y}, f(\boldsymbol{x}))\right]$ are sufficient to guarantee a small robust classification loss $\mathbb{E}_{\boldsymbol{x}\sim\mathbb{P},\boldsymbol{t}\sim\mathbb{T}}\left[\ell_{0\text{-}1}(\boldsymbol{y}, f(\phi_{\boldsymbol{t}}(\boldsymbol{x})))\right]$, where $\boldsymbol{y}$ is the ground-truth label of $\boldsymbol{x}$ (Zhang et al., 2019).

3.2. Robustness Guarantee with Domain Translator

Directly minimizing the robustness loss on the target distribution $L_\phi(f, \mathbb{P})$ requires transformed target examples pairs $\{\boldsymbol{x}_i, \phi_{\boldsymbol{t}_i}(\boldsymbol{x}_i)\}$ to estimate the expectation. However, we lack these pairs and are unaware of the transformation function needed to sample them. In this case, the following proposition shows the feasibility of leveraging the available source examples with the help of a domain translator.

To simplify notation, we use $\bar{\ell}_f : \mathcal{X} \to \mathbb{R}$ to denote the function $\bar{\ell}_f(\boldsymbol{x}) := \mathbb{E}_{t\sim\mathbb{T}}[\ell(f(\boldsymbol{x}), f(\phi_{\boldsymbol{t}}(\boldsymbol{x})))]$, which intuitively measures the robustness loss of the model at a given example. Given some (measurable) function $\xi : \mathcal{X} \to \mathcal{X}$, we use $\xi_\# \mathbb{Q}$ to denote the push-forward probability distribution[1] of $\mathbb{Q}$ on $\mathcal{X}$. We use W_1 to denote Wasserstein-1 distance.

Proposition 3.2. *We assume that $\bar{\ell}_f$ is Lipschitz uniformly over all models f, with a (possibly infinite) Lipschitz constant $\|\bar{\ell}\|_L$. Then for any (measurable) function $\xi : \mathcal{X} \to \mathcal{X}$, the following holds:*

$$L_\phi(f, \mathbb{P}) \le I_1 + I_2 + I_3, \tag{3.2}$$

$$\begin{aligned} \textit{where} \quad I_1 &= \mathop{\mathbb{E}}_{u\sim\mathbb{Q},t\sim\mathbb{T}}\left[\ell\big(f(\xi(\boldsymbol{u})), f(\xi \circ \phi_{\boldsymbol{t}}(\boldsymbol{u}))\big)\right], \\ I_2 &= \mathop{\mathbb{E}}_{u\sim\mathbb{Q},t\sim\mathbb{T}}\left[\ell\big(f(\xi \circ \phi_{\boldsymbol{t}}(\boldsymbol{u})), f(\phi_{\boldsymbol{t}} \circ \xi(\boldsymbol{u}))\big)\right], \\ I_3 &= \|\bar{\ell}\|_L W_1(\mathbb{P}, \xi_\# \mathbb{Q}). \end{aligned}$$

This proposition, proved in Appendix A.1, upper-bounds the robustness loss on the target distribution by three terms illustrated in Figure 2. We can intuitively interpret ξ as a domain translator which translates a given source example into another example that "looks like" the target examples[2]. Below, we remark on two properties of the domain translator.

Equivariant domain translator minimizes I_2. Note that any domain translator ξ satisfying $\xi \circ \phi_t(\boldsymbol{u}) = \phi_t \circ \xi(\boldsymbol{u})$

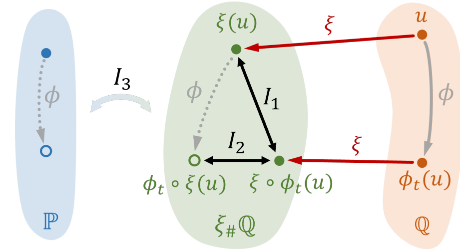


Figure 2. Illustration of our proposition. Here ξ represents the domain translator. Term I_1 measures the model's consistency loss on the domain-translated example pairs. Term I_2 measures the model's consistency loss on a ground-truth transformed example and its approximated version generated by the domain translator, which can be minimized if the domain translator is equivariant. Term I_3 measures how well the push-forward distribution matches the target distribution.

(almost surely with respect to $\mathbb{P}\times\mathbb{T}$) is sufficient to minimize the term I_2 for *any* model f (assuming $\bar{\ell}_f$ has bounded range). Particularly, such ξ satisfying $\xi \circ \phi_t(\boldsymbol{u}) = \phi_t \circ \xi(\boldsymbol{u})$ is said to be *equivariant* if t belongs to a group and ϕ_t be the group action (Cohen & Welling, 2016). Nevertheless, we abuse the notion and refer to any ξ approximately satisfying this property (measured by some loss) as being equivariant.

Accurate domain translator minimizes I_3. Note that any domain translator ξ pushing the source distribution to match the target distribution accurately such that $W_1(\mathbb{P}, \xi_\#\mathbb{Q}) = 0$ is sufficient to minimize the term I_2 to zero for *any* model f (assuming bounded $\|\bar{\ell}\|_{\mathrm{L}}$). We refer to any ξ approximately satisfying this property as being accurate.

The above two remarks imply that we can learn an equivariant and accurate domain translator to minimize I_2 and I_3 regardless of the model f, which motivates our two-step algorithm in the next section. We empirically demonstrate the existence of such domain translators for certain datasets and leave further existence discussion to Appendix A.2.

4. The Two-Step Algorithm

Based on Proposition 3.2, we propose a two-step algorithm for learning unforeseen robustness from out-of-distribution data. We first describe step one, which trains an equivariant and accurate domain translator. To encourage equivariance, we offer three optional regularization losses, which can be chosen based on available knowledge. Then we describe step two, where we learn the desired robust model using the trained domain translator. Figure 3 depicts the algorithm.

4.1. Step One: Training Domain Translator

To begin, we provide the training objective for the equivariant domain translator, assuming that we know the transformation function characterizing the considered data variation. This scenario corresponds to the unsupervised data augmen-

[1] We state the definition in Appendix A.1.

[2] Compared to the style transfer work, such translation is unpaired and does not need to preserve the underlying concept class.

tation problem (Xie et al., 2020).

$$\min_{\xi} \ W_1(\mathbb{P}, \xi_{\#}\mathbb{Q}) + \lambda \underset{\boldsymbol{u}\sim\mathbb{Q}}{\mathbb{E}} \underset{\boldsymbol{t}\sim\mathbb{T}}{\mathbb{E}} [\ell(\xi \circ \phi_{\boldsymbol{t}}(\boldsymbol{u}), \phi_{\boldsymbol{t}} \circ \xi(\boldsymbol{u}))] \tag{4.1}$$

where the first term minimizes I_3, encouraging accurate domain translation, and the second term minimizes I_2, encouraging equivariance. The hyperparameter λ balances the two objectives.

In our implementation, we adopt the encoder-decoder architecture commonly used in style transfer literature as our domain translator ξ. To optimize the first term, we follow WGAN (Arjovsky et al., 2017) and train the domain translator ξ under the supervision of an auxiliary domain discriminator that has regularized Lipschitz constant. To estimate and optimize the second term, we proceed as follows: we randomly sample one transformation parameter for each source example, apply domain translation followed by transformation to get $\phi_{\boldsymbol{t}} \circ \xi(\boldsymbol{u})$, and apply transformation followed by domain translation to get $\xi \circ \phi_{\boldsymbol{t}}(\boldsymbol{u})$. We encourage the domain translator to generate examples such that the two terms are similar in terms of the ℓ_2 loss.

In addition to WGAN, some recent work suggests that score-based generative models also implicitly minimize the Wasserstein distance (Kwon et al., 2022). Nevertheless, we defer further exploration of alternative implementations, such as image-to-image diffusion models (Tumanyan et al., 2022; Bansal et al., 2022), to future work.

4.2. Methods for Encouraging Equivariance

Directly encouraging the equivariance (the second term in Eq. 4.1) for the domain translator ξ requires knowing the data transformation function ϕ. However, when learning unforeseen robustness, we only have some transformed source example pairs $\{(\boldsymbol{u}_i, \phi_{\boldsymbol{t}_i}(\boldsymbol{u}_i))\}$ but lack knowledge about the underlying transformation function. This poses a challenge to encouraging equivariance since we cannot transform a domain-translated example $\xi(\boldsymbol{u})$ to get $\phi_{\boldsymbol{t}} \circ \xi(\boldsymbol{u})$ in Eq. 4.1. To address this issue, we provide three optional methods for encouraging equivariance based on different assumptions about the available knowledge.

First, in some special cases where we know the transformation function, such as when using our algorithm to do unsupervised data augmentation, we can directly encourage equivariance by minimizing the equivariance loss in Eq. 4.1.

Second, if we do not know the transformation function but know the transformation parameters $\{\boldsymbol{t}_i\}$ used to generate the pairs of transformed examples $\{(\boldsymbol{u}_i, \phi_{\boldsymbol{t}_i}(\boldsymbol{u}_i))\}$, such as when learning unforeseen robustness from some simulated data, we can empirically encourage equivariance to the transformation $\phi_{\boldsymbol{t}}$ by training a predictor that predicts the transformation parameters $\{\boldsymbol{t}_i\}$ based on the model's output. This method is proposed by some recent work (Qi et al., 2019; Lee et al., 2021; Dangovski et al., 2022).

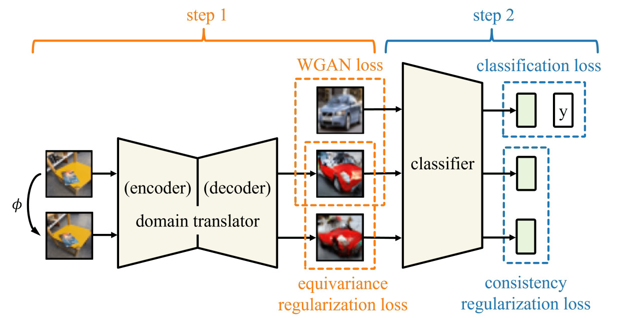


Figure 3. Overview of our two-step algorithm for learning unforeseen robustness from out-of-distribution data. First, we train an equivariant and accurate domain translator using the equivariance regularization loss and the WGAN loss. Second, we use the trained domain translator to translate all out-of-distribution data into target-like ones while preserving the considered variations. We then train a robust model using consistency regularization on the domain-translated data.

Third, in cases where both the transformation function and transformation parameters are unknown, we propose an alternative method to encourage equivariance. The main idea is to encourage a learnable feature extractor to extract the same encoded information about the transformation parameter $\{\boldsymbol{t}_i\}$ from both the original source example pairs $\{(\boldsymbol{u}_i, \phi_{\boldsymbol{t}_i}(\boldsymbol{u}_i))\}$ and the domain translated source example pairs $\{(\xi(\boldsymbol{u}_i), \xi \circ \phi_{\boldsymbol{t}_i}(\boldsymbol{u}_i))\}$. The intuition becomes more evident when we consider fixing the feature extractor using some hard-coded or pretrained model, such as an optical flow estimator. In such cases, we encourage the extraction of the same encoded transformation parameter from the two pairs, similar to the second method of predicting the transformation parameter. A more detailed description of this method appears in Appendix B.1.

4.3. Step Two: Training Robust Model

Our goal in this problem is to improve the robustness of a model while performing some primary task. To illustrate this, we consider the classification task with a given classification loss $L_{\text{classifier}}$. Building upon the prior training of the domain translator, which minimizes I_2 and I_3 in Proposition 3.2, we proceed to train a robust classifier f to minimize I_1 and $L_{\text{classifier}}$ while freezing the translator.

For notation simplicity, we write I_1 as a functional of f and ξ. We use ξ^* to denote the trained domain translator, and use ξ_{id} to denote the identity domain translator, which maps any example to itself (perfectly equivariant but not accurate). Then, the training objective is

$$\min_{f} \quad L_{\text{classifier}}(f) + \lambda_1 I_1(f, \xi^*) + \lambda_2 I_1(f, \xi_{\text{id}}), \tag{4.2}$$

where λ_1 and λ_2 are weight hyperparameters. We include the last term in the objective, which is essentially consistency regularization on the source data, since we observe that this often produces the best result. In fact, the UDA method (Xie et al., 2020) can be viewed as a special case of our method, with $\lambda_1 = 0$ and $\lambda_2 = 1$.

5. Empirical Evaluation

In this section, we empirically verify the effectiveness of our two-step algorithm in learning unforeseen robustness. Different from Section 6, this section only considers synthetic data variations since they enable reliable robustness evaluation on the target dataset. To begin, we show that our equivariance-encouraging method effectively trains equivariant domain translators. Then, we compare the two-step algorithm with two existing methods extended to our setting and provide an ablation study. Lastly, we show that the training of the equivariant domain translator, as a by-product, also serves as a criterion for selecting suitable source datasets for learning robustness. Our experimental settings are as follows:

Datasets. We use CIFAR-10 and CIFAR-100 as our target datasets, while selecting the source dataset from a range of options including SVHN, STL-10, CIFAR-100, MNIST, CelebA, and Caltech-256. Note that some source datasets, such as MNIST or CelebA, are visually distinct from the target datasets, mirroring real-world scenarios where the considered data variation is only available from extremely out-of-distribution data.

Data variations. We use two types of synthetic data variations: (1) RandAugment (Cubuk et al., 2020), which includes a diverse range of 14 random transformations, spanning from geometric transformations to color space changes, and their random combinations, The variety of these transformations allows us to evaluate our algorithm's ability to preserve such variations; (2) Random rotation, as a supplementary evaluation due to its well-defined nature. Despite its simplicity, modeling random rotation using model-based methods has proven to be a challenging task (Zhou et al., 2022). To simulate the scenario of learning unforeseen robustness, we refrain from accessing the transformation function or transformed target example pairs during training. Instead, we only use the transformation function during testing to evaluate the learned robustness.

Other settings. To implement the domain translator, we adopt the encoder-decoder architecture borrowed from CycleGAN (Zhu et al., 2017), which comprises two down-sampling convolutional layers, two residual blocks for latent propagation, and two up-sampling convolutional layers. We use ResNet18 (He et al., 2016) to implement the classifier. We use cross-entropy loss for classification, KL-divergence for consistency regularizing, and mean-squared-error (MSE)

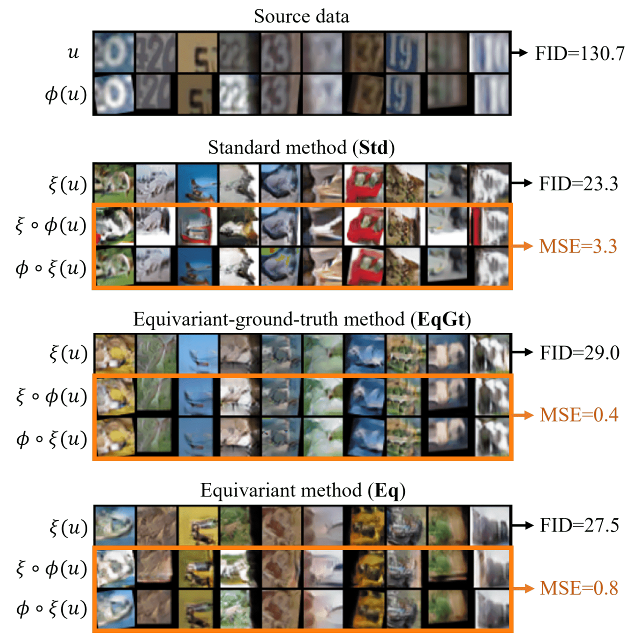


Figure 4. Quantitative and visualization results of domain translators trained using three different equivariance-encouraging methods. Each row of images is labeled on its left to indicate its meaning. Images are column-aligned, with the inputs being the corresponding source images from the first row. All three domain translators translate the source data to closely resemble the target, as indicated by the low FID compared to the original source data. However, domain translators without equivariance regularization (Std) fail to preserve variations well, as highlighted by the examples in the orange box and high MSE loss. Compared to EqGt, our equivariant regularization method Eq achieves comparable equivariance loss without accessing the transformation function or transformation parameters, demonstrating its effectiveness.

loss for measuring the equivariance of the domain translator (the second term in Eq. 4.1). Unless otherwise stated, we set $\lambda = 1$ in Eq. 4.1 and $\lambda_1 = \lambda_2 = 0.5$ in Eq. 4.2. In this section, we do not apply data augmentation on the source or target to avoid entanglement with the considered data variation. We defer more setup details to Appendix C, and most results on random rotation and CIFAR-100 to Appendix D.

5.1. Training Equivariant Domain Translator

We first test if the equivariance-encouraging methods provided in Section 4.2 can train equivariant domain translators. Specifically, we compare three methods: (1) The baseline method, denoted as `Std` (standard), which does not apply any equivariance regularization by setting $\lambda = 0$ in Eq. 4.1; (2) The first optional method, denoted as `EqGt` (equivariant-ground-truth), which encourages equivariance using the ground-truth data transformation function; (3) The third optional method, denoted as `Eq` (equivariant), which does not require access to the transformation function or transformation parameters.

We use the Fréchet Inception Distance (FID, Heusel et al. (2017)) as our evaluation metric, since it provides a reliable measure of how well the domain translator translates the source data to resemble the target, given the difficulty in directly estimating the W_1 distance. Figure 4 showcases some results of training the domain translators with the three methods, using CIFAR-10 as the target and SVHN as the source dataset. We present several results below and defer additional results with different datasets, including domain translation under real-world illumination changes with limited data (Murmann et al., 2019), to Appendix D.

Our method trains accurate translators. Despite the significant difference between SVHN and CIFAR-10, as indicated by a direct FID calculation of 130.7, all three domain translators are effective in translating SVHN to resemble CIFAR-10, achieving FIDs less than 30. This shows that our method trains accurate domain translators.

Our method trains equivariant translators. Both `EqGt` and `Eq` achieve much lower equivariance loss (0.4 and 0.8, respectively) than `Std` (3.3) while maintaining similar FIDs, demonstrating their capacity to preserve the data variations while still generate target-like output. The better equivariance is also visually observable from the images highlighted within the orange boxes. Notably, the images in the first and seventh columns depict rotation transforms, which are challenging for model-based methods to capture (Zhou et al., 2022) but are well preserved by `EqGt` and `Eq`. In particular, `Eq` preserves the various transformations in RandAugment almost as effectively as `EqGt`, yet without knowing ground-truth transformation functions or transformation parameters, underscoring its effectiveness and generality.

5.2. Learning Robust Classifiers

With the trained domain translator, we proceed to train robust classifiers against unforeseen variations. We compare our two-step algorithm with three baseline methods: MBRDL (Robey et al., 2020), UDA (Xie et al., 2020), and empirical risk minimization (ERM). ERM trains the classifier without any robustness intervention. We describe the implementation details of these methods in Appendix C. Unless otherwise specified, we use the domain translator trained using the equivariance-encouraging method `Eq`.

We evaluate the classifiers using three metrics: (1) *Robust accuracy* (R), which measures the probability of a model preserving its prediction under input variations; (2) *Robust Classification accuracy* (RC), which measures the probability of a model predicting the correct label under input variations; (3) *Standard accuracy* (S), which measures the probability of a model predicting the correct label.

Our algorithm learns unforeseen robustness. Despite the stark dissimilarity between SVHN and CIFAR-10, Table 1 shows that both our algorithm and UDA can harness the variations on SVHN to improve the robust classification accuracy on CIFAR-10 by 4.1% and 3.2%, respectively, indicating the feasibility of learning unforeseen robustness from out-of-distribution data. Moreover, our algorithm improves the standard accuracy whereas UDA sometimes hurts it. Using SVHN, our algorithm increases the standard accuracy by 0.9% over ERM, whereas UDA falls short by 0.8%. Meanwhile, MBRDL underperforms ERM in all three metrics, indicating the importance of the learning methods. Our analysis of MBRDL in Appendix D.5 suggests that it may fail due to the domain gap and the difficulty in modeling complex variations in RandAugment.

Our algorithm consistently outperforms UDA. Since the consistency regularization in UDA and our algorithm intro-

Table 1. Results of classifiers trained using different methods and source datasets. The target dataset is CIFAR-10 without data augmentation, and the variation is RandAugment. RC, R, and S denote robust classification accuracy, robust accuracy, and standard accuracy, respectively. For reference, we include the oracle method that applies consistency regularization directly on the target dataset. Our method achieves the best robustness and accuracy.

		Robustness		**Accuracy**
Method	**Src**	**RC** (%)	**R** (%)	**S** (%)
ERM	/	79.1 ± 0.2	82.5 ± 0.2	89.0 ± 0.2
MBRDL	SVHN	68.7 ± 0.4	77.4 ± 0.3	78.9 ± 0.3
UDA	SVHN	82.3 ± 0.2	85.5 ± 0.3	88.2 ± 0.3
Ours	SVHN	**83.2** ± 0.3	**86.7** ± 0.3	**89.9** ± 0.2
MBRDL	STL10	72.1 ± 0.4	78.8 ± 0.3	82.9 ± 0.3
UDA	STL10	85.8 ± 0.3	89.5 ± 0.2	89.9 ± 0.3
Ours	STL10	**87.8** ± 0.2	**91.5** ± 0.3	**91.0** ± 0.3
Oracle	/	91.7 ± 0.1	94.8 ± 0.2	93.3 ± 0.1

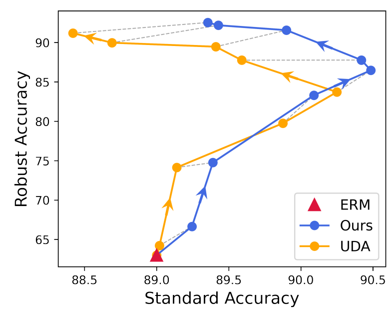


Figure 5. Robust vs. standard accuracy of classifiers trained with different weights of consistency regularization. The source is STL10, the target is CIFAR-10, and the data variation is random rotation. We gradually increase (denoted by the arrow) the weight from 0 to 5, producing different classifiers whose results are denoted by dots. The pair of dots connected by a gray dashed line have the same weight setting. Our method outperforms UDA in each weight setting and achieves better Pareto-optimal.

Table 2. Ablation study of the two-step algorithm, varying whether to use the source dataset (Src) and the training method of the domain translator (DT). Using the equivariant domain translator plays a key role in learning unforeseen robustness.

Src	DT	SVHN		STL-10	
		RC (%)	**S** (%)	**RC** (%)	**S** (%)
✓	`EqGt`	83.7 (↑ 0.5)	89.5	88.1 (↑ 0.3)	91.2
✓	`Eq`	83.2	89.9	87.8	91.0
✓	`Std`	82.8 (↓ 0.4)	88.5	86.2 (↓ 1.6)	90.6
✓	×	82.3 (↓ 0.9)	88.2	85.8 (↓ 2.0)	89.9
×	×	79.1 (↓ 4.1)	89.0	79.1 (↓ 8.7)	89.0

duces an extra weight hyperparameter, we further vary that weight for a comprehensive comparison and show results in Figure 5. For both methods, we observe two stages as the regularization weight increases. In the first stage, increasing the weight improves both robust and standard accuracy. In the second, however, increasing the weight improves robust accuracy but hurts standard accuracy, leading to a robustness-accuracy trade-off. Nevertheless, our method outperforms UDA across all weight settings and achieves better Pareto-optimal in the second stage.

Equivariant domain translator is the key. Table 2 shows the ablation study result for the two-step algorithm. Compared to ERM (last row), harnessing variations from the source dataset (top four rows) improves the target robustness significantly. Furthermore, both `EqGt` and `Eq` outperform `Std` and the one not using the domain translator (fourth row), indicating the importance of using an equivariant domain translator. Among the top two rows, `Eq` shows comparable robust classification and standard accuracy to `EqGt`, indicating that our equivariance-encouraging method trains equivariant domain translators that are equally helpful for downstream classification.

5.3. Source Dataset Selection

When learning unforeseen robustness, we cannot use cross-validation to select suitable source datasets to learn from due to the lack of target data variations. For example, it is hard to determine whether using SVHN or CelebA would better improve the robustness to RandAugment on CIFAR-10. In this case, we show that the training result of an equivariant domain translator can serve as a selection criterion. Specifically, given the target dataset, the variation, and a source dataset, we evaluate three available selection criteria: (1) `DT-Eq-FID`, which trains an `Eq` domain translator and then computes the FID between the target dataset and the domain-translated source dataset. (2) `DT-Std-FID`, similar to `DT-Eq-FID`, but uses an `Std`-trained domain translator. (3) `Naive-FID`, which directly computes the FID between the target and the source datasets. For all three criteria, we select source datasets with lower FIDs.

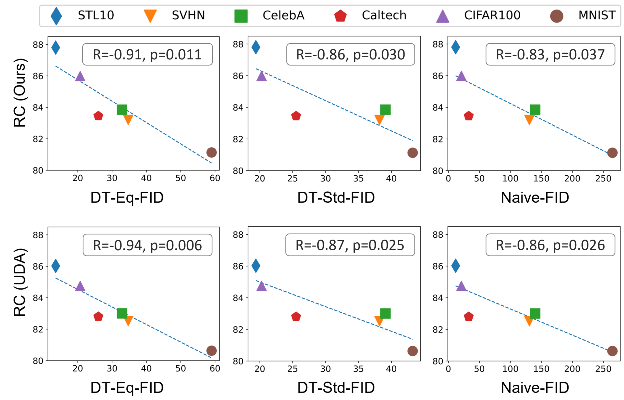


Figure 6. Correlation results for three source dataset selection criteria, with our method (first row) or UDA (second row) training the classifier on the corresponding source dataset. Each point represents a source dataset, where the x-coordinate is the score given by the criterion and the y-coordinate is the actual robust classification accuracy of the resulting classifier. We measure the Pearson correlation (R) and the p-value (p). `DT-Eq-FID`, which is based on our equivariant domain translator training, shows the strongest correlation among the three even when for UDA.

Equivariant DT can select suitable source. In Figure 6, we evaluate the three criteria for selecting source datasets for our method and UDA. `DT-Eq-FID`, based on our equivariant domain translator training, shows the strongest correlation among the three (R=-0.91 for our method, R=-0.94 for UDA) with the resulting classifier's robustness, indicating its effectiveness as a general source dataset selection criterion. It favors CelebA over SVHN for learning robustness to RandAugment on CIFAR-10, which corroborates our results, whereas the other two criteria do not. `DT-Eq-FID` also has two desired properties compared to `DT-Std-FID` and `Naive-FID`. It is sensitive to the considered data variation, while `DT-Std-FID` and `Naive-FID` are not, enabling it to explain why the same source dataset can have different benefits for different data variations. It also depends on the order of the source and target datasets, enabling it to explain why SVHN as the source and CIFAR-10 as the target gives a worse result than the other way around.

6. Applications

Now, we apply our algorithm to two real-world tasks to show its practical significance. First, we train robust CIFAR-10 classifiers to unforeseen 3D-viewpoint changes. Then, we harness out-of-distribution data to further improve foreseen robustness on the target data, resulting in improved in-distribution and out-of-distribution generalization.

6.1. Learning Unforeseen Robustness in Real-world

To evaluate the effectiveness of learning unforeseen real-world robustness, we choose Objectron (Ahmadyan et al.,

Table 3. Robust classification accuracy on CIFAR10 under six geometric data transformations, which serves as a surrogate for the 3D-viewpoint-change robustness. Our method best learns the unforeseen robustness to this natural variation.

Variations	**ERM** (%)	**UDA** (%)	**Ours** (%)
Affine	69.2 ± 0.5	69.7 (↑ 0.5)	70.9 (↑ **1.7**)
Rotate	83.3 ± 0.3	83.4 (↑ 0.1)	84.5 (↑ **1.2**)
Perspective	61.6 ± 0.6	54.8 (↓ 6.8)	63.2 (↑ **1.6**)
Crop	85.5 ± 0.1	85.4 (↓ 0.1)	86.2 (↑ **0.7**)
Elastic transform	85.9 ± 0.3	86.4 (↑ 0.5)	87.3 (↑ **1.4**)
Fisheye	43.7 ± 1.2	33.9 (↓ 9.8)	43.8 (↑ **0.1**)
Plate Spline	81.6 ± 0.3	81.4 (↓ 0.2)	82.8 (↑ **1.2**)

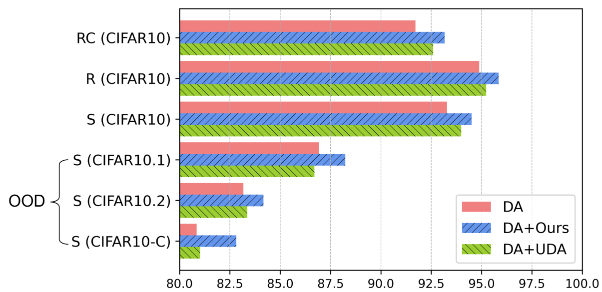


Figure 7. For foreseen variations, using our method in addition to data augmentation (DA) further improves robustness (RC and R), ID generalization (S), and OOD generalization (S on three OOD test sets). Compared to UDA, our method not only better improves robustness and in-distribution generalization, but also benefits OOD generalization while UDA cannot, demonstrating its superiority as an unsupervised data augmentation method.

2021) as the source data. Objectron contains video clips reflecting 3D viewpoint changes in the real world. We construct the transformed pairs by randomly selecting an anchor frame and its adjacent frames. We set $\lambda_1 = 1$ and $\lambda_2 = 0$ for our algorithm. As we cannot directly compute the viewpoint change robustness on CIFAR-10 (target), we select six common geometric transformations as a surrogate. We compare our algorithm with UDA, which is the most effective baseline in our evaluation. Results in Table 3 show that our algorithm achieves comprehensive improvements in robustness to all the surrogate transformations, outperforming UDA in the same tasks. Domain-translated images are shown in Figure 12. We leave learning unforeseen robustness from simulated data to future work.

6.2. Improving Unsupervised Data Augmentation

Moreover, we test if our algorithm can improve foreseen robustness and serve as a generalized and improved unsupervised data augmentation method. Following the setting used by prior work (Xie et al., 2020), we train CIFAR-10 classifiers with RandAugment variations and then use our method to further improve the robustness with STL-10. To test the out-of-distribution generalization, we use CIFAR10.1 (Recht et al., 2019), CIFAR10.2 (Lu et al., 2020), and CIFAR-10-C (Hendrycks & Dietterich, 2019).

Figure 7 shows our results. We further improve the robustness and in-domain generalization, doubling the improvement brought by UDA in the same setting. In addition, we improve the accuracy on three out-of-distribution datasets by 1.5%, 1.2%, and 2.4%, respectively, whereas UDA barely helps. This result demonstrates our method's superiority for unsupervised data augmentation using foreseen variations.

7. Conclusion

This paper introduces a new approach to learning robustness that broadens the scope of existing robustness interventions. Unlike previous methods confined to a limited range of data variations, our approach harnesses the variations observed in some source data to learn the robustness on the target data, thus expanding the spectrum of robustness types that can be effectively learned.

Limitations. The most evident limitation of our approach is the additional computational burden introduced by training the domain translator. Appendix B.2 provides an analysis of the computational cost compared to existing methods. This is mainly due to the need for training different domain translators for different source data. Therefore, we hope that future work can develop a foundation model for image-to-image translation, allowing our method to achieve almost cost-free domain translation by simply fine-tuning with the addition of equivariance regularization.

Another drawback of our approach is the need to find suitable source data for training, which is not always readily available as existing datasets are not constructed with our specific problem in mind. However, many datasets, particularly some simulation datasets (e.g., for autonomous driving), have the capability to exhibit real-world variations. Hence, we encourage the community to consider incorporating validation data with various variations when constructing datasets, both for learning and evaluating robustness.

Acknowledgments

Zhu, An, and Huang are supported by National Science Foundation NSF-IIS-FAI program, DOD-ONR-Office of Naval Research, DOD Air Force Office of Scientific Research, DOD-DARPA-Defense Advanced Research Projects Agency Guaranteeing AI Robustness against Deception (GARD), Adobe, Capital One and JP Morgan faculty fellowships.

References

Ahmadyan, A., Zhang, L., Wei, J., Ablavatski, A., and Grundmann, M. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7818–7827, 2021. 8, 15

Alayrac, J.-B., Uesato, J., Huang, P.-S., Fawzi, A., Stanforth, R., and Kohli, P. Are labels required for improving adversarial robustness? *Advances in Neural Information Processing Systems*, 32, 2019. 3

Antoniou, A., Storkey, A. J., and Edwards, H. Data augmentation generative adversarial networks. *CoRR*, 2017. 3

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017. 2, 5, 16

Bansal, A., Borgnia, E., Chu, H.-M., Li, J. S., Kazemi, H., Huang, F., Goldblum, M., Geiping, J., and Goldstein, T. Cold diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*, 2022. 5

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. 17

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. S. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019. 3

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021. 14

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In Gordon, G., Dunson, D., and Dudík, M. (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223, Fort Lauderdale, FL, USA, Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/coates11a.html. 15

Cohen, T. and Welling, M. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pp. 2990–2999. PMLR, Jun 2016. URL http://proceedings.mlr.press/v48/cohenc16.html. 4

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. Randaugment: Practical automated data augmentation with a reduced search space. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18613–18624. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d85b63ef0ccb114d0a3bb7b7d808028f-Paper.pdf. 6

Dangovski, R., Jing, L., Loh, C., Han, S., Srivastava, A., Cheung, B., Agrawal, P., and Soljacic, M. Equivariant self-supervised learning: Encouraging equivariance in representations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gKLAAfiytI. 5

Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 15

Deng, Z., Zhang, L., Ghorbani, A., and Zou, J. Improving adversarial robustness via unlabeled out-of-domain data. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2845–2853. PMLR, Apr 2021. URL https://proceedings.mlr.press/v130/deng21b.html. 3

Endres, D. M. and Schindelin, J. E. A new metric for probability distributions. *IEEE Transactions on Information theory*, 49(7):1858–1860, 2003. 3

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. Exploring the landscape of spatial robustness. In *ICML*, 2019. 1

Gatys, L., Ecker, A. S., and Bethge, M. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28, 2015. 3

Griffin, G., Holub, A., and Perona, P. Caltech 256. 15

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. 6

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm. 1, 3, 9, 15

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., and Gilmer, J. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv:2006.16241 [cs, stat]*, Aug 2020. URL http://arxiv.org/abs/2006.16241. arXiv: 2006.16241. 1

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7, 17

Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 172–189, 2018. 3, 17

Huang, Z., Xia, X., Shen, L., Han, B., Gong, M., Gong, C., and Liu, T. Harnessing out-of-distribution examples via augmenting content and style. (arXiv:2207.03162), Jul 2022. doi: 10.48550/arXiv.2207.03162. URL http://arxiv.org/abs/2207.03162. arXiv:2207.03162 [cs]. 3

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017. 3

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pp. 694–711. Springer, 2016. 3

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. Wilds: A benchmark of in-the-wild distribution shifts. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5637–5664. PMLR, Jul 2021. URL https://proceedings.mlr.press/v139/koh21a.html. 1

Koralov, L. and Sinai, Y. G. *Theory of probability and random processes*. Springer Science & Business Media, 2007. 13

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Technical report*, 2009. 15

Kwon, D., Fan, Y., and Lee, K. Score-based generative modeling secretly minimizes the wasserstein distance. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=oPzICxVFqVM. 5

Le, Y. and Yang, X. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 15

Lee, H., Lee, K., Lee, K., Lee, H., and Shin, J. Improving transferability of representations via augmentation-aware self-supervision. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=U34rQjnImpM. 5

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 15

Lu, S., Nott, B., Olson, A., Todeschini, A., Vahabi, H., Carmon, Y., and Schmidt, L. Harder or different? a closer look at distribution shift in dataset reproduction. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2020. 9, 15

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb. 1

Murmann, L., Gharbi, M., Aittala, M., and Durand, F. A multi-illumination dataset of indoor object appearance. In *2019 IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 7, 18

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 15

Pope, P., Zhu, C., Abdelkader, A., Goldblum, M., and Goldstein, T. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XJk19XzGq2J. 14

Qi, G.-J., Zhang, L., Chen, C. W., and Tian, Q. Avt: Unsupervised learning of transformation equivariant representations by autoencoding variational transformations. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8129–8138, Oct 2019. doi: 10.1109/ICCV.2019.00822. 5, 14

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018. 15

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5389–5400. PMLR, Jun 2019. URL `https://proceedings.mlr.press/v97/recht19a.html`. 9

Robey, A., Hassani, H., and Pappas, G. J. Model-based robust deep learning: Generalizing to natural, out-of-distribution data. *arXiv:2005.10247 [cs, stat]*, Nov 2020. URL `http://arxiv.org/abs/2005.10247`. arXiv: 2005.10247. 2, 3, 7, 17

Saito, K., Kim, D., and Saenko, K. Openmatch: Open-set semi-supervised learning with open-set consistency regularization. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=77cNKCCjgw`. 3

Salmona, A., Bortoli, V. D., Delon, J., and Desolneux, A. Can push-forward generative models fit multimodal distributions? In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=Tsy9WCO_fK1`. 3, 14

Shankar, V., Dave, A., Roelofs, R., Ramanan, D., Recht, B., and Schmidt, L. Do image classifiers generalize across time? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9661–9669, 2021. 1

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pp. 596–608. Curran Associates, Inc., 2020. 1, 2

Tacchetti, A., Isik, L., and Poggio, T. A. Invariant recognition shapes neural representations of visual input. *Annual Review of Vision Science*, 4(1):403–422, 2018. doi: 10.1146/annurev-vision-091517-034103. 1

Tanielian, U., Issenhuth, T., Dohmatob, E., and Mary, J. Learning disconnected manifolds: a no gan's land. In *International Conference on Machine Learning*, pp. 9418–9427. PMLR, 2020. 3

Tumanyan, N., Bar-Tal, O., Bagon, S., and Dekel, T. Splicing vit features for semantic appearance transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10748–10757, 2022. 5

Villani, C. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021. 14

Wei, C., Shen, K., Chen, Y., and Ma, T. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=rC8sJ4i6kaH`. 1, 2

Xie, Q., Dai, Z., Hovy, E., Luong, T., and Le, Q. Unsupervised data augmentation for consistency training. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6256–6268. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/44feb0096faa8326192570788b38c1d1-Paper.pdf`. 1, 2, 3, 5, 6, 7, 9, 18

Zhang, H., Yu, Y., Jiao, J., Xing, E., Ghaoui, L. E., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482. PMLR, Jun 2019. URL `https://proceedings.mlr.press/v97/zhang19p.html`. 2, 4, 17

Zhang, R. Making convolutional networks shift-invariant again. In *ICML*, 2019. 1

Zhou, A., Tajwar, F., Robey, A., Knowles, T., Pappas, G. J., Hassani, H., and Finn, C. Do deep networks transfer invariances across classes? In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=Fn7i_r5rR0q`. 1, 3, 6, 7, 15, 19

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017. 3, 6

Appendix

Table of Contents

A. Additional Analysis

A.1. Proof of Proposition 3.2

Before giving the proof, we first state the definition of push-forward distribution, which appears in many textbooks (see, e.g., Koralov & Sinai (2007)).

Definition A.1 (Push-forward distribution)**.** *Given a probability space* $(\Omega, \mathcal{F}, \mathbb{P})$*, a measurable space* $(\tilde{\Omega}, \tilde{\mathcal{F}})$*, and a measurable mapping* $\xi : \Omega \to \tilde{\Omega}$*, the push-forward distribution of* $\mathbb{P}$ *on the* σ*-algebra* $\tilde{\mathcal{F}}$ *is defined by*

$$\xi_{\#}\mathbb{Q}(A) = \mathbb{P}(\xi^{-1}(A)) \quad \textit{for } A \in \tilde{\mathcal{F}},$$

where $\xi^{-1}(A) := \{\omega \in \Omega : \xi(\omega) \in A\}$ *denotes the pre-image of a measurable set* A*.*

The proof follows from the assumptions that the loss ℓ satisfies the triangle inequality and $\bar{\ell}_f$ is Lipschitz uniformly over all models f. Since we are working on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ with functions implemented by neural networks (with continuous activation functions) and common losses, we omit the measurability issue.

Proof. First, since ℓ is non-negative, by Tonelli's theorem, we have

$$L_\phi(f, \mathbb{P}) := \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}, \boldsymbol{t}\sim\mathbb{T}} \left[\ell\big(f(\boldsymbol{x}), f(\phi_{\boldsymbol{t}}(\boldsymbol{x}))\big)\right] = \mathop{\mathbb{E}}_{x\sim\mathbb{P}} \left[\bar{\ell}_f(x)\right],$$

where $\bar{\ell}_f(\boldsymbol{x}) := \mathbb{E}_{t\sim\mathbb{T}}[\ell\big(f(\boldsymbol{x}), f(\phi_{\boldsymbol{t}}(\boldsymbol{x}))\big)]$.

Then, since $\bar{\ell}_f$ is uniformly Lipschitz with a Lipschitz constant $\|\bar{\ell}\|_{\mathrm{L}}$, by Kantorovich-Rubenstein duality theorem (see, e.g., (Villani, 2021)), we have

$$\mathop{\mathbb{E}}_{x\sim\mathbb{P}}\left[\bar{\ell}_f(x)\right] - \mathop{\mathbb{E}}_{x\sim\xi_{\#}\mathbb{Q}}\left[\bar{\ell}_f(x)\right] \leq \|\bar{\ell}\|_{\mathrm{L}}\, W_1(\mathbb{P},\xi_{\#}\mathbb{Q}).$$

Thirdly, since $\xi_{\#}\mathbb{Q}$ is the push-forward distribution of $\mathbb{Q}$ through the mapping ξ, by change of measure, we have

$$\mathop{\mathbb{E}}_{x\sim\xi_{\#}\mathbb{Q}}\left[\bar{\ell}_f(x)\right] = \mathop{\mathbb{E}}_{u\sim\mathbb{Q}}\left[\bar{\ell}_f(\xi(u))\right].$$

Lastly, since ℓ satisfies the triangle inequality, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{u\sim\mathbb{Q}}\left[\bar{\ell}_f(\xi(u))\right] &= \mathop{\mathbb{E}}_{u\sim\mathbb{Q}}\mathop{\mathbb{E}}_{t\sim\mathbb{T}}\left[f(\xi(\boldsymbol{u})), f(\phi_{\boldsymbol{t}}\circ\xi(\boldsymbol{u})))\right]\\ &\leq \mathop{\mathbb{E}}_{u\sim\mathbb{Q}}\mathop{\mathbb{E}}_{t\sim\mathbb{T}}\left[f(\xi(\boldsymbol{u})), f(\xi\circ\phi_{\boldsymbol{t}}(\boldsymbol{u})))\right] + \mathop{\mathbb{E}}_{u\sim\mathbb{Q}}\mathop{\mathbb{E}}_{t\sim\mathbb{T}}\left[f(\xi\circ\phi_{\boldsymbol{t}}(\boldsymbol{u})), f(\phi_{\boldsymbol{t}}\circ\xi(\boldsymbol{u})))\right]\end{aligned}$$

Rearranging terms completes the proof. □

A.2. Discussion about the Existence of Equivariant and Accurate Domain Translators

We discuss some of our conjectures about the existence here and leave the complete characterization to future work. Since we use continuous maps to instantiate ξ, we conjecture that the equivariant domain translator does not exist if the support of the source data distribution, after being expanded by the transformation, has a smaller intrinsic dimension (see, e.g., Pope et al. (2021); Salmona et al. (2022)) than that of the target. Indeed, we empirically observe that for some source and target datasets such as SVHN to CIFAR-10, training the domain translator yields a trade-off between the equivariance and the approximate performance, but such trade-off mitigates if we swap the source and target datasets. Interestingly, this existence issue seems to enable us to use the training result of an *equivariant* domain translator as the source selection criterion.

B. Algorithm Details

B.1. Detailed Method to Encourage Equivariance for Domain Translator

Figure 8 illustrates our proposed method and discusses the intuition behind it. Compared to previous work, our method only requires the transformed source example pairs and their domain-translated counterparts. To use it in training the domain translator, we replace the second term in Eq. 4.1 with the cosine similarity loss shown in the figure. We simultaneously train the domain translator, projector, and predictor, to minimize the loss. This method applies to any data transformation that can be represented as $\phi_{\boldsymbol{t}}$, without needing to modify the architecture or hyperparameters. When we know the transformation's type (e.g., motion changes across video frames), we may also hard-code the predictor accordingly (e.g., use an optical-flow estimator) for better performance.

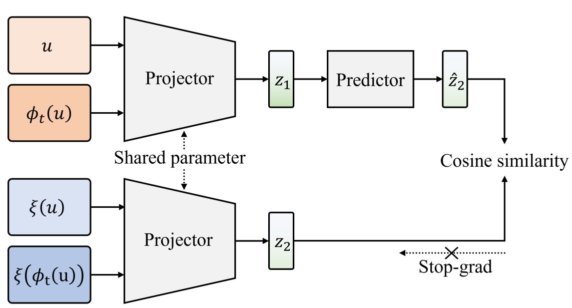


Figure 8. Illustration of our proposed method for encouraging the equivariance of the domain translator, requiring neither the transformation function $\phi_{\boldsymbol{t}}$ nor its parameter $\boldsymbol{t}$. Here, the projector, whose architecture refers to Qi et al. (2019), takes as input the original example $\boldsymbol{u}$ and its transformed version $\phi_{\boldsymbol{t}}(\boldsymbol{u})$ and outputs a vector $\boldsymbol{z}_1$. The intuition is that $\boldsymbol{z}_1$ may contain the encoded transformation parameter, which is exactly the case when the projector is a hard-coded model like an optical flow estimator. If the domain translator is equivariant, then the domain-translated pair $\xi(\boldsymbol{u})$ and $\xi(\phi_{\boldsymbol{t}}(\boldsymbol{u}))$ should also contain the same encoded transformation parameter. Thus, we encourage z_1 and z_2 to be similar, which is implemented with a predictor to prevent degeneration (referring to SimSiam (Chen & He, 2021)).

B.2. Computational Complexity Analysis

We report computational complexity in Table 4 and 5. When training the classifier, our method requires approximately 24% more time compared to UDA and MBRDL. When only using the trained domain translator, the required time is similar to that of UDA and MBRDL. We note that our method can potentially be accelerated by pre-translating all source examples and implementing proper parallelization techniques.

Table 4. Computational complexity of training the auxiliary modules

Method	Complexity			GPU seconds per epoch (batch size 256)
	encoder-decoder	discriminator	projector-predictor	
UDA	n/a	n/a	n/a	n/a
MBRDL	4, 2	2,2	n/a	79s
Ours	3, 1	1, 1	2, 1	75s
WGAN (with encoder-decoder)	2, 1	1, 1	n/a	64s

Table 5. Computational complexity of training the classifier

	Complexity		GPU seconds per epoch (batch size 256)
	encoder-decoder	classifier	
UDA	n/a	3, 1	78s
MBDL	1, 0	1+k, 1 (we choose k=1)	80s
Ours (translate source online, $\lambda_2 = 0$)	1, 0	3, 1	83s
Ours (translate source online, $\lambda_2 \neq 0$)	1, 0	5, 1	97s

B.3. Algorithm Pseudocode

We show the pseudocode of training the domain translator and classifier in Algorithm 1 and 2.

C. Detailed Experimental Setup

Datasets. In Section 5, We use CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) as target datasets and SVHN (Netzer et al., 2011), STL-10 (Coates et al., 2011), CIFAR-100, MNIST (Deng, 2012), CelebA (Liu et al., 2015), and Caltech-256 (Griffin et al.). When training domain translators, we only use unlabeled images from the source and target. In Section 6, we use Objectron (Ahmadyan et al., 2021) as the source dataset to learn 3D-viewpoint-change robustness. Objectron is a collection of short, object-centric video clips. We randomly sample several frames from each clip as the anchor images and randomly sample frames in a range of 10 frames as the 3D-viewpoint changed images. We use such pairs to do 3D-viewpoint change consistency regularization. To evaluate the out-of-distribution generalization of classifiers trained on CIFAR-10, we use CIFAR-10.1 (Recht et al., 2018), CIFAR-10.2 (Lu et al., 2020), and CIFAR-10-C (Hendrycks & Dietterich, 2019) as the ood datasets. CIFAR-10.1 and CIFAR-10.2 are sampled from TinyImageNet (Le & Yang, 2015) with the same classes of CIFAR-10. CIFAR-10-C is a collection of a corrupted version of CIFAR-10 under 15 types of corruption.

Data variations. In Section 5, we use RandAugment and random rotation as the variations. RandAugment contains 14 candidate transformation functions: "ShearX", "ShearY", "TranslateX", "TranslateY", "Rotate", "Brightness", "Color", "Contrast", "Sharpness", "Posterize", "Solarize", "AutoContrast", "Equalize", and "Identity". When using RandAugment, a composition of two randomly selected functions are applied to the images. For random rotation, we use $[-30°, 30°]$ random rotation. Although the rotation is simply defined, it cannot be modeled by existing model-based methods that use MUNIT-like architectures (Zhou et al., 2022). In Section 6, we consider 3D-viewpoint change as the unforeseen variation. We randomly select two nearby frames from one video clip as the two 3D-views of one object. Since we could not evaluate the model robustness to 3D-viewpoint change on the target data (CIFAR-10), we use six proxy transformations to estimate the 3D-viewpoint robustness. Proxy transformations are geometric transformations that do warping on images, which include "Random Affine", "Random Rotate", "Random Perspective", " Random Crop", " Random Fisheye", "Random Thin Plate Spline"[3].

[3]Implementation follows https://kornia.readthedocs.io/en/latest/augmentation.module.html

Algorithm 1 Training the domain translator (PyTorch-style pseudocode)

Input : domain translator ξ, discriminator $disc$, projector $proj$, and predictor $pred$
target data $\{x_i\}_{i=1}^N$, source data $\{(u_i, u_i')\}_{i=1}^M$ (consists of transformed pairs),
batch size B, coefficient λ for equivariance regularization
Output : trained domain translator ξ^*

```
randomly initialize all modules
for epoch in range(max_training_epochs) do
    for a target batch X_B = {x_i}_{i=1}^B in all target data do
        randomly sample a source batch {(u_i, u'_i)}_{i=1}^B
        ( denote U_B = {u_i}_{i=1}^B, U'_B = {u'_i}_{i=1}^B, UU'_B = {(u_i, u'_i)}_{i=1}^B, ξ(U_B) = {ξ(u_i)}_{i=1}^B )

        # training discriminator disc
        translate the source batch to get ξ(U_B)
        loss_disc = disc(ξ(U_B)).mean() − disc(X_B).mean()
        update disc according to loss_disc
        clip the parameters in disc (following WGAN)

        # get accuracy loss for domain translator ξ
        translate the source batch to get ξ(U_B)
        loss_acc = −disc(ξ(U_B)).mean()

        # get equivariance loss for domain translator ξ
        translate the source batch to get UU'_B
        p_1 = proj(UU'_B)
        p_2 = proj(ξ(UU'_B)
        loss_eq = −cosine_similarity (p_1.detach(), pred(p_2)) .mean()

        # training domain translator ξ
        loss_ξ = loss_acc + λ · loss_eq
        update ξ according to loss_eq

    end
end
```

Evaluation Metrics. We evaluate the trained classifiers with three metrics[4]: the *robust accuracy*, denoted as R, measures the probability of a model preserving its output under input variations, the *robust classification accuracy*, denoted as RC, measures the probability of a model predicting the correct label under input variations, the *standard accuracy*, denoted as S, measures the probability of a model predicting the correct label. During testing, we randomly sample 20 transformed versions for each example to estimate the expectation of robust accuracy and robust classification accuracy.

C.1. Our Method

We use Wasserstein GAN (Arjovsky et al., 2017) to train a domain translator where the inputs of the generator (i.e. domain translator) are source images and the outputs are encouraged to be similar to the target images. We use the encoder-decoder model architecture for implementing the domain translator (i.e. generator), which consists of two convolutional layers for down-sampling, two residual blocks for latent propagation, and two other convolutional layers for up-sampling. The discriminator then distinguishes the real target data from the fake ones translated from the source data. We train generator and discriminator with adversarial training following WGAN where we use 0.01 as the clip value of the discriminator's weight. For training equivariant domain translator, we use the mean-squared-error (MSE) loss for the equivariance regularization term (the second term in Eq. 4.1). We set $\lambda = 1$ in Eq. 4.1.

[4]Each of them can be viewed as one minus the corresponding loss (instantiated with zero-one loss) defined in Section 3.1.

Algorithm 2 Training the classifier (PyTorch-style pseudocode)

Input : trained domain translator ξ^*, classifier f
target data $\{x_i\}_{i=1}^N$, source data $\{(u_i, u_i')\}_{i=1}^M$ (consists of transformed pairs),
batch size B, coefficient λ_1 and λ_2 for weighing the trained and the identity
domain translator.

Output : trained classifier f^*

randomly initialize all modules

for *epoch in range(max_training_epochs)* **do**

  for *a target batch* $\{(x_i, label_i)\}_{i=1}^B$ *in all target data* **do**

    randomly sample a source batch $\{(u_i, u_i')\}_{i=1}^B$
    (denote $X_B = \{x_i\}_{i=1}^B$, $Label_B = \{label_i\}_{i=1}^B$,$U_B = \{u_i\}_{i=1}^B$, $U_B' = \{u_i'\}_{i=1}^B$,
    $UU_B' = \{(u_i, u_i')\}_{i=1}^B$, $\xi(U_B) = \{\xi(u_i)\}_{i=1}^B$)

    # get classification loss
    $l_{\text{classify}} = cross\_entropy(f(X_B), Label_B)$

    # get consistency loss under trained domain translator ξ^*
    translate the source batch to get $\xi^*(U_B)$ and $\xi^*(U_B')$ # online translation
    $p_1 = softmax(f(\xi^*(U_B')))$
    $p_2 = softmax(f(\xi^*(U_B)))$
    $loss_{\text{trained}} = kl\_divergence(p_1,\ p_2.detach())$

    # get consistency loss under identity map
    translate the source batch to get UU_B'
    $p_1 = softmax(f(U_B'))$
    $p_2 = softmax(f(U_B))$
    $loss_{\text{identity}} = kl\_divergence(p_1,\ p_2.detach())$

    # training classifier f
    $loss = loss_{\text{classify}} + \lambda_1 \cdot loss_{\text{trained}} + \lambda_2 \cdot loss_{\text{identity}}$
    update f according to $loss$

  end

end

For the robust classifier, we use ResNet18 as the architecture. Since the zero-one loss is difficult to optimize directly, we follow the common practice of using the surrogate loss (Bartlett et al., 2006). We use the cross-entropy loss for training the classifier, including the robustness regularization term I_1, similar to Zhang et al. (2019). The MSE loss and the L^1 norm loss are two common training objectives that measure the difference between two images in the pixel space. They are used as the reconstruction loss in VAE, CycleGAN, Diffusion Model, etc. We also tried the L^1 loss for the equivariance regularization term but did not observe substantial difference. In all our experiments, we use cross-entropy loss as the surrogate loss for training and regularizing the classifier. We set $\lambda_1 = \lambda_2 = 0.5$ in Eq. 4.2. Since accurately estimating the W_1 distance for multi-dimensional non-Gaussian distributions is difficult, we use the Fréchet inception distance (FID, see Heusel et al. (2017)) to evaluate how well the domain translator pushes forward the source data to approximate the target data.

C.2. MBRDL

MBRDL (model-based robust deep learning, (Robey et al., 2020)) learns a model to simulate the natural variation. In their paper, the variation model is learned and applied to the same domain. Their method can easily extend to scenarios where variations are unforeseen in the target domain but is available in the source domain. In this paper, we first learn a variation simulator with the source data where transformed pairs are used for learning variations. We use MUNIT (Huang et al., 2018) as the variation simulator following settings in (Robey et al., 2020). MUNIT is first designed for style transfer, here (Robey et al., 2020) use it for input transformation. Then, we apply the variation simulator directly to the target data to do data variation and train robust classifiers with a consistency regularization loss addition to the classification loss.

C.3. UDA

UDA (unsupervised data augmentation, (Xie et al., 2020)) improves the model's robustness against variations with consistency regularization on unlabeled data. Although the unlabeled data is very similar to the target data and has foreseen variations in their paper, we can directly use their method in our case. We see source data as the unlabeled data and do consistency regularization on it while training the classifier on the target data. It's easy to see that, UDA is a simple version of our method where $\lambda_1 = 0$ in Eq. 4.2. In our experiments of UDA, we set $\lambda_2 = 1$.

D. Additional Results

D.1. Visualizing the Results of Equivariant Domain Translator

We show the outputs of our domain translators in Figure 10, 11 and 12. Results demonstrate that our method can effectively translate the source data to be target-like. The trained domain translator also well-preserve the variations including random rotation, RandAugment, and 3D-viewpoint change. Therefore, we are able to do consistency regularization with the target-like images and the transformed version of them, so that to train a robust classifier under unforeseen variations. We notice that domain translators trained with different source dataset have different performances. As discussed in Section 5.3, the source dataset's distance to the target dataset correlates with the performance. Additionally, if the source dataset is much "simpler" than the target one, such as MNIST and SVHN, it is very difficult for the domain translator to cover the whole manifold of the target distribution, and to preserve complex variations such as RandAugment (especially the color change) on MNIST. One interesting future work is to take the intrinsic dimension of the dataset into consideration.

In addition, we evaluate the capability of the equivariant domain translator to preserve more real-world variations using limited data. To this end, we train the equivariant domain translator to preserve illumination changes from the Multi-illumination dataset (comprising 1015 images, Murmann et al. (2019)) to the labeled training set of STL-10 (comprising 5k images). Figure 9 shows some domain-translated images for visual evaluation.

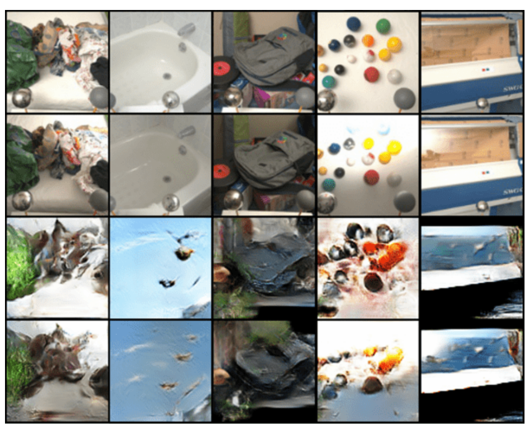


Figure 9. Visualization of some domain-translated images from the Multi-illumination dataset (comprising 1015 images) to the STL10's training set (comprising 5k images), demonstrating the preservation of illumination changes using limited data.

D.2. Results on CIFAR-100

Table 6 shows the results on CIFAR-100 where we use SVHN, STL10 and CIFAR-10 as the source data. Data variation is the RandAugment. We get consistent results where our method excels over other methods in robustness and accuracy.

D.3. Additional Baselines

Incorporating additional baselines, although not originally designed to address our specific problem, can stimulate deeper insights into the problem setting. In Table 7, we consider contrastive self-supervised learning and additionally evaluate 1)

(a) SVHN as the source dataset. Random rotation as the variation.

(b) STL10 as the source dataset. Random rotation as the variation.

Figure 10. Results of our method with random rotation as the input variation. We use CIFAR-10 as the target dataset. z denotes the source data, ϕ denotes the variation, i.e. random rotation, and ξ denotes Eq, the domain translator trained with the heuristic method. By comparing $\xi(z)$ with CIFAR-10 data, results indicate that our method can effectively translate the source data to be target-like. By comparing between $\xi \circ \phi(z)$ and $\phi \circ \xi(z)$, which are expected to be similar, our domain translators well-preserve the variations.

contrastive pretraining on the source, and 2) contrastive learning on the source as regularization which uses the SimCLR loss on the source as auxiliary regularization for training the classifier since it encourages invariance. We evaluate the two methods on CIFAR-10 and CIFAR-100. The contrastive pretraining, under our hyperparameter setting (temperature=0.2, latent dimension=128, two-layer projection head), does not show a significant difference from ERM, so we only report the result of contrastive learning as regularization here. Table 7 shows the result on CIFAR-10 under RandAugment with different regularization weights, which corresponds to Table 1 in the paper. The result is averaged over three independent runs.

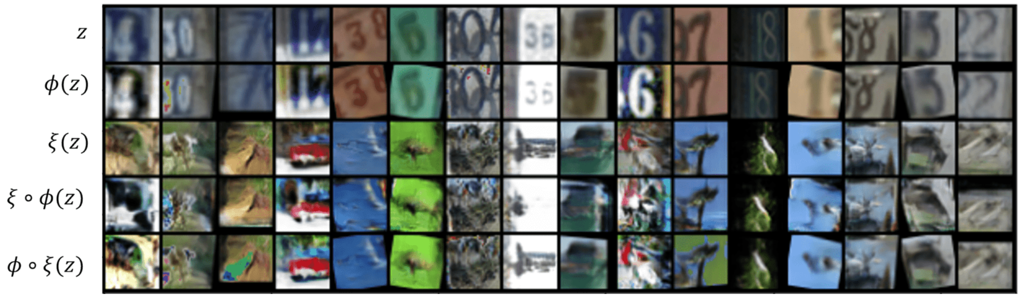
Using the optimal regularization weight (0.1), SimCLR improves all three metrics over ERM (and MBRDL). The robustness benefit, however, is less than that brought by UDA and our method. Interestingly, the SimCLR has better standard accuracy than UDA (+1.6% on SVHN and +0.3% on STL10), suggesting that while SimCLR cannot provide the same level of robustness as UDA, the projection head and the InfoNCE loss better benefits the standard accuracy.

D.4. Sensitivities to Source Sample Size

We further investigate whether the number of source datapoints is a significant factor. In Table 8 we show some initial results. Our empirical findings suggest that 1) when the source sample size is "comparable" with the target sample size (greater than 25k or 50% of the target training sample size), there is no noticeable change in the final robust classification accuracy. 2) As the source sample size decreases below 50% of the target sample size, the robust classification accuracy gradually drops. Nonetheless, our method still offers a small benefit over ERM even when the source sample size is as small as one batch size (256).

D.5. Problems of MBRDL

Figure 13 and 14 shows the performance of the variation simulator learned by MBRDL. We can see that the MBRDL suffers from two problems. Firstly, it is hard to learn a good variation simulator. As (Zhou et al., 2022) observed and as shown in



(a) SVHN as the source dataset. RandAugment as the variation.

(b) STL10 as the source dataset. RandAugment as the variation.

(c) CelebA as the source dataset. RandAugment as the variation.

(d) MNIST as the source dataset. RandAugment as the variation.

Figure 11. Results of our method with RandAugment as the input variation. We use CIFAR-10 as the target dataset. z denotes the source data, ϕ denotes the variation, i.e. RandAugment, and ξ denotes Eq, the domain translator trained with the heuristic method. By comparing $\xi(z)$ with CIFAR-10 data, results indicate that our method can effectively translate the source data to be target-like. By comparing between $\xi \circ \phi(z)$ and $\phi \circ \xi(z)$, which are expected to be similar, our domain translators well-preserve the variations in most cases.

Figure 12. Results of our method with 3D-viewpoint change. We use CIFAR-10 as the target dataset and Objectron as the source dataset. Here, z denotes the source data, ϕ denotes the variation, i.e. 3D-viewpoint change, and ξ denotes Eq, the domain translator trained with the heuristic method. By comparing $\xi(z)$ with CIFAR-10 data, results indicate that our method can effectively translate the source data to be target-like. $\xi \circ \phi(z)$ shows that the domain translator well-preserves the 3D-viewpoint change. For example, in the fourth column, two cars generated by $\xi(z)$ and $\xi \circ \phi(z)$ well-preserve the viewpoint change that exits in two chair images (i.e. z and $\phi(z)$).

Table 6. Results of classifiers trained using different methods and source datasets. The target dataset is CIFAR-100 w/o data augmentation and the data variation is RandAugment. We show here for reference the oracle method that does consistency regularization directly on the target dataset.

		Robustness		**Accuracy**
Method	**Src**	**RC** (%)	**R** (%)	**S** (%)
ERM	/	48.8 ± 0.1	57.2 ± 0.2	62.9 ± 0.3
MBRDL	SVHN	36.9 ± 0.4	55.3 ± 0.5	52.4 ± 0.3
UDA	SVHN	51.7 ± 0.2	61.6 ± 0.2	63.2 ± 0.4
EDT (Ours)	SVHN	**53.2** ± 0.3	**63.4** ± 0.2	**64.1** ± 0.3
MBRDL	STL10	39.6 ± 0.3	56.1 ± 0.3	56.1 ± 0.2
UDA	STL10	55.9 ± 0.3	67.1 ± 0.2	64.1 ± 0.3
EDT (Ours)	STL10	**58.3** ± 0.3	**70.0** ± 0.3	**65.1** ± 0.3
MBRDL	CIFAR-10	39.6 ± 0.4	58.4 ± 0.3	56.2 ± 0.3
UDA	CIFAR-10	56.5 ± 0.2	68.3 ± 0.2	63.8 ± 0.3
EDT (Ours)	CIFAR-10	**59.0** ± 0.2	**71.2** ± 0.3	**64.5** ± 0.2
Oracle	/	70.9 ± 0.2	82.1 ± 0.2	73.6 ± 0.2

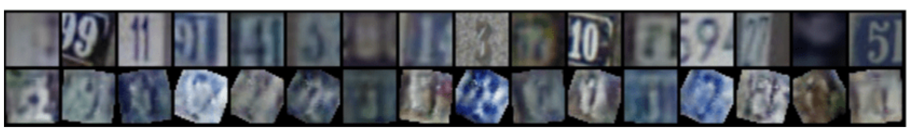
Figure 13, brightness change and color change are easy to learn but geometric transformations such as rotation are hard to learn. The complex variations such as RandAugment are even harder. Secondly, the learned variation simulator has poor generalization ability. Figure 14 (a) and (c) show that the variation simulator which is trained on the source data performs well on the source data. However, (b) and (d) show that the variation simulator performs badly when directly applied to the target data, resulting in blurred images or content-changed images. We suspect that it is because the variation is very hard to learn and it is even harder to learn a variation simulator that is disentangled from the source data. The problems get severe when the target domain and the source domain are far from each other. This explains why MBRDL hurts the robustness and accuracy in our experiments.

Table 7. Performance of SimCLR on SVHN and STL10 datasets

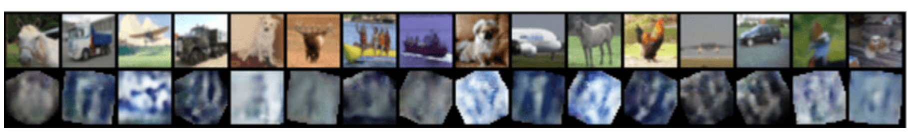
Method	Source	Weight	Robust Classification Accuracy	Robust Accuracy	Standard Accuracy
SimCLR	SVHN	0.001	81.2 ± 0.2	84.7 ± 0.2	89.4 ± 0.1
SimCLR	SVHN	0.01	**81.8 ± 0.1**	**85.5 ± 0.1**	**89.6 ± 0.2**
SimCLR	SVHN	0.1	80.8 ± 0.1	84.6 ± 0.1	89.5 ± 0.3
SimCLR	SVHN	1	80.4 ± 0.1	84.4 ± 0.2	89.1 ± 0.0
SimCLR	STL10	0.001	79.7 ± 0.2	82.8 ± 0.3	89.4 ± 0.0
SimCLR	STL10	0.01	82.1 ± 0.2	85.4 ± 0.3	89.6 ± 0.1
SimCLR	STL10	0.1	**84.0 ± 0.1**	**87.8 ± 0.1**	**90.3 ± 0.4**
SimCLR	STL10	1	81.6 ± 0.8	85.5 ± 0.9	89.8 ± 0.3

Table 8. Performance on SVHN and STL10 datasets with varying training sizes.

src	0 (ERM)	256	1,024	4,096	16,384	65,536	All
SVHN	79.1 ± 0.2	80.1 ± 0.4	80.2 ± 0.5	80.8 ± 0.3	82.6 ± 0.3	–	83.2 ± 0.3 (73,257)
STL10	79.1 ± 0.2	81.0 ± 0.5	81.7 ± 0.4	82.0 ± 0.3	84.8 ± 0.2	87.9 ± 0.3	87.8 ± 0.2 (100,000)

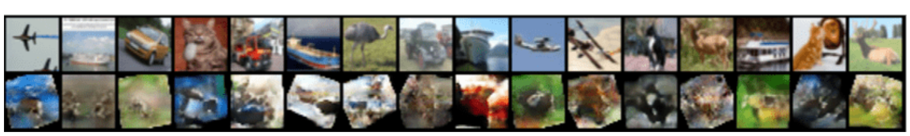


(a) Apply rotation simulator learned on SVHN to SVHN.



(b) Apply rotation simulator learned on SVHN to CIFAR10.

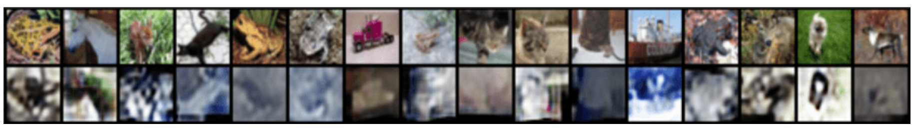
(c) Apply rotation simulator learned on STL10 to STL10.



(d) Apply rotation simulator learned on STL10 to CIFAR10.

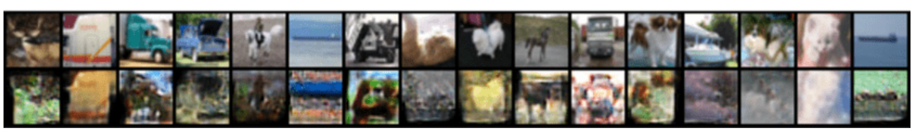
Figure 13. Results of MBRDL with random rotation as the input variation. In every subfigure, the first line shows the original images and the second line shows the transformed ones using the learned variation simulator.

(a) Apply RandAugment simulator learned on SVHN to SVHN.



(b) Apply RandAugment simulator learned on SVHN to CIFAR10.

(c) Apply RandAugment simulator learned on STL10 to STL10.



(d) Apply RandAugment simulator learned on STL10 to CIFAR10.

Figure 14. Results of MBRDL with RandAugment as the input variation. In every subfigure, the first line shows the original images and the second line shows the transformed ones using the learned variation simulator.