Domain Invariant Adversarial Learning

Matan Levi *matanle@post.bgu.ac.il*
Department of Computer Science
Ben-Gurion University of the Negev

Idan Attias *idanatti@post.bgu.ac.il*
Department of Computer Science
Ben-Gurion University of the Negev

Aryeh Kontorovich *karyeh@bgu.ac.il*
Department of Computer Science
Ben-Gurion University of the Negev

Reviewed on OpenReview: *https://openreview.net/forum?id=U8uJAUMzj9*

Abstract

The phenomenon of adversarial examples illustrates one of the most basic vulnerabilities of deep neural networks. Among the variety of techniques introduced to surmount this inherent weakness, adversarial training has emerged as the most effective strategy for learning robust models. Typically, this is achieved by balancing robust and natural objectives. In this work, we aim to further optimize the trade-off between robust and standard accuracy by enforcing a domain-invariant feature representation. We present a new adversarial training method, *Domain Invariant Adversarial Learning* (DIAL), which learns a feature representation that is both robust and domain invariant. DIAL uses a variant of Domain Adversarial Neural Network (DANN) on the natural domain and its corresponding adversarial domain. In the case where the source domain consists of natural examples and the target domain is the adversarially perturbed examples, our method learns a feature representation constrained not to discriminate between the natural and adversarial examples, and can therefore achieve a more robust representation. DIAL is a generic and modular technique that can be easily incorporated into any adversarial training method. Our experiments indicate that incorporating DIAL in the adversarial training process improves both robustness and standard accuracy.

1 Introduction

Deep learning models have achieved impressive success on a wide range of challenging tasks. However, their performance was shown to be brittle in the face of *adversarial examples*: small, imperceptible perturbations in the input that drastically alter the classification (Carlini & Wagner, 2017a;b; Goodfellow et al., 2014; Kurakin et al., 2016b; Moosavi-Dezfooli et al., 2016; Szegedy et al., 2013; Tramèr et al., 2017; Dong et al., 2018; Tabacof & Valle, 2016; Xie et al., 2019b; Rony et al., 2019). The problem of designing reliable robust models has gained significant attention in the arms race against adversarial examples. Adversarial training (Szegedy et al., 2013; Goodfellow et al., 2014; Madry et al., 2017; Zhang et al., 2019b) has been proposed as one of the most effective approaches to defend against such examples, and can be described as solving the following min-max optimization problem:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{x':\|x'-x\|_p\leq\epsilon} L\left(x', y; \theta\right)\right],$$

Our source code is available at `https://github.com/matanle51/DIAL`

where x' is the ϵ-bounded perturbation in the ℓ_p norm and L is the loss function. Different unrestricted attacks methods were also suggested, such as adversarial deformation, rotations, translation and more (Brown et al., 2018; Engstrom et al., 2018; Xiao et al., 2018; Alaifari et al., 2018; Gilmer et al., 2018).

The resulting min-max optimization problem can be hard to solve in general. Nevertheless, in the context of ϵ-bounded perturbations, the problem is often tractable in practice. The inner maximization is usually approximated by generating adversarial examples using projected gradient descent (PGD) (Kurakin et al., 2016a; Madry et al., 2017). A PGD adversary starts with randomly initialized perturbation and iteratively adjust the perturbation while projecting it back into the ϵ-ball:

$$x_{t+1} = \Pi_{\mathbb{B}_\epsilon(x_0)} \left(x_t + \alpha \cdot \text{sign}(\nabla_{x_t} L(G(x_t), y)) \right),$$

where x_0 is the natural example (with or without random noise), and $\Pi_{\mathbb{B}_\epsilon(x)}$ is the projection operator onto the ϵ-ball, G is the network, and α is the perturbation step size. As was shown by Athalye et al. (2018), PGD-based adversarial training was one of the few defenses that were not broken under strong attacks.

That said, the gap between robust and natural accuracy remains large for many tasks such as CIFAR-10 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009). Generally speaking, Tsipras et al. (2018) suggested that robustness may be at odds with natural accuracy, and usually the trade-off is inherent. Nevertheless, a growing body of work aimed to improve the standard PGD-based adversarial training introduced by Madry et al. (2017) in various ways such as improved adversarial loss functions and regularization techniques (Kannan et al., 2018; Wang et al., 2019b; Zhang et al., 2019b), semi-supervised approaches(Carmon et al., 2019; Uesato et al., 2019; Zhai et al., 2019), adversarial perturbations on model weights (Wu et al., 2020), utilizing out of distribution data (Lee et al., 2021) and many others. We refer to related work for a more extensive literature review.

Our contribution. In this work, we propose a novel approach to regulating the tradeoff between robustness and natural accuracy. In contrast to the aforementioned works, our method enhances adversarial training by enforcing a feature representation that is invariant across the natural and adversarial domains. We incorporate the idea of Domain-Adversarial Neural Networks (DANN) (Ganin & Lempitsky, 2015; Ganin et al., 2016) directly into the adversarial training process. DANN is a representation learning approach for domain adaptation, designed to ensure that predictions are made based on invariant feature representation that cannot discriminate between source and target domains. This technique is modular and can be easily incorporated into any standard adversarial training algorithm. Intuitively, the tasks of adversarial training and of domain-invariant representation have a similar goal: given a source (natural) domain X and a target (adversarial) domain X', we hope to achieve $g(X) \approx g(X')$, where g is a feature representation function (i.e., neural network). As we present in section 3.3, our work is also theoretically motivated by the domain adaptation generalization bounds.

In a comprehensive battery of experiments on MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011), CIFAR-10 (Krizhevsky et al., 2009) and CIFAR-100 (Krizhevsky et al., 2009) datasets, we demonstrate that by enforcing domain-invariant representation learning using DANN simultaneously with adversarial training, we gain a significant and consistent improvement in both robustness and natural accuracy compared to other state-of-the-art adversarial training methods, under Auto-Attack (Croce & Hein, 2020) and various strong PGD (Madry et al., 2017), and CW (Carlini & Wagner, 2017b) adversaries in white-box and black-box settings. Additionally, we evaluate our method using unforeseen "natural" corruptions (Hendrycks & Dietterich, 2018), unforeseen adversaries (e.g., ℓ_1, ℓ_2), transfer learning, and perform ablation studies. Finally, we offer a novel score function for quantifying the robust-natural accuracy trade-off.

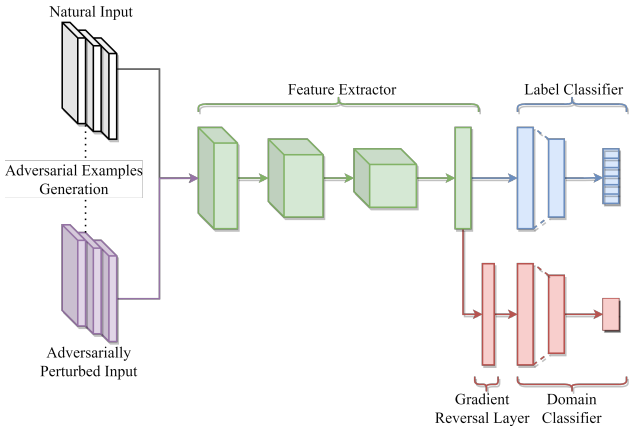


Figure 1: Illustration of the proposed architecture to enforce domain invariant representation. The feature extractor and label classifier form the a regular DNN architecture that can be used for the main natural task. The domain classifier is incorporated alongside the label classifier. The reversal gradient layer multiplies the gradient by a negative number during the back-propagation.

2 Related work

2.1 Defense methods

A variety of theoretically principled (Raghunathan et al., 2018a; Sinha et al., 2017; Raghunathan et al., 2018b; Wong et al., 2018; Wong & Kolter, 2018; Gowal et al., 2018) and empirical defense approaches (Bai et al., 2021) were proposed to enhance robustness since the discovery of adversarial examples. Theoretically principled methods focus on certifying robustness to adversarial perturbations under a given norm, using variety of techniques such as randomized smoothing (Cohen et al., 2019). However, empirical defence methods in general, and in particular adversarial training, still yield preferable results.

Among the empirical defence techniques, we can find: *adversarial regularization*—adding various regularization terms to the loss functions to enhance robustness (e.g., encouraging logits for clean and adversarial examples to be similar) (Kurakin et al., 2016a; Madry et al., 2017; Zhang et al., 2019b; Wang et al., 2019b; Kannan et al., 2018; Jin et al., 2022), *curriculum-based adversarial training*—taking incremental approach when learning PGD adversaries in the goal of improving generalization on clean data while still preserving robustness (e.g., gradually increasing the number of PGD iterations to avoid overfitting the adversarial examples) (Cai et al., 2018; Zhang et al., 2020; Wang et al., 2019a), *ensemble adversarial training*—where clean data is augmented with adversarial examples generated from different target models instead of a single model (Tramèr et al., 2017; Pang et al., 2019; Yang et al., 2020), *adversarial training with adaptive attack budget*—where we change the perturbation budget to prevent over-confident predictions and achieve better exploration of the manifold (Ding et al., 2018; Cheng et al., 2020), *semi-supervised and unsupervised adversarial training*—several methods theoretically and empirically demonstrated how unlabeled data can reduce the sample complexity gap between standard training and adversarial training (Carmon et al., 2019; Uesato et al., 2019; Zhai et al., 2019), *robust self and pre-training*—other works integrate self-supervised pretraining tasks such as Selfie, Rotation and Jigsaw together with adversarial examples (Jiang et al., 2020; Chen et al., 2020), *efficient adversarial training*—due to the high cost of adversarial training, efficient methods aim to keep the favorable performance of adversarial training while reducing the computational and time costs (Shafahi et al., 2019; Wong et al., 2020; Andriushchenko & Flammarion, 2020; Zhang et al., 2019a), and many other techniques such as adversarial training based on feature scatter (Zhang & Wang, 2019), adversarially robust distillation (Goldblum et al., 2020), hypersphere embedding (Pang et al., 2020b), and augmenting adversarial examples by interpolation (Lee et al., 2020). In an additional research direction, researchers suggested to add new dedicated building blocks to the network architecture for improved robustness (Xie & Yuille, 2019; Xie et al., 2019a; Liu et al., 2020). Liu et al. (2020) hypothesised that different adversaries belong to different domains, and suggested gated batch normalization which is trained with multiple perturbation types. Guo et al. (2020) focused on

searching robust architectures against adversarial examples. Others works presented improved robustness by combining data augmentation techniques and generated data (Rebuffi et al., 2021a;b), where the latest is also the current state-of-the-art in robustness.

Our work belongs to the the family of adversarial regularization techniques, for which we elaborate on common and best performing methods, and highlight the differences compared to our method.

Madry et al. (2017) proposed a technique, commonly referred to as Adversarial Training (AT), to minimize the cross entropy loss on adversarial examples generated by PGD (without using the natural examples). Zhang et al. (2019b) suggested to decompose the prediction error for adversarial examples as the sum of the natural error and boundary error, and provided differentiable upper bounds on both terms. Motivated by this decomposition, they suggested a technique called TRADES that uses the Kullback-Leibler (KL) divergence as a regularization term that will push the decision boundary away from the data. They do so by applying the KL-divergence on the logits of clean examples and their adversarial counterparts. Wang et al. (2019b) suggested that misclassified examples have a significant impact on final robustness, and proposed a technique called MART that differentiate between correctly classified and miss-classified examples during training by weighting the KL-divergence between the clean and adversarial logits using the probability of the classifier on the correct label.

Another area of research aims at revealing the connection between the loss weight landscape and adversarial training (Prabhu et al., 2019; Yu et al., 2018; Wu et al., 2020). Specifically, Wu et al. (2020) identified a correlation between the flatness of weight loss landscape and robust generalization gap. They proposed the Adversarial Weight Perturbation (AWP) mechanism that is integrated into existing adversarial training methods and generates adversarial perturbations on both the inputs and the network weights. More recently, this approach was formalized from a theoretical standpoint by Tsai et al. (2021). However, this method forms a double-perturbation mechanism that perturbs both inputs and weights, which may incur a significant increase in calculation overhead. We demonstrate how DIAL improves results also when combined with AWP, named DIAL_{AWP}. In Section 3.4 we elaborate about the loss functions of the different compared methods.

A related approach to ours, called ATDA, was presented by Song et al. (2018). They proposed to add several constrains to the loss function in order to enforce domain adaptation: correlation alignment and maximum mean discrepancy (Borgwardt et al., 2006; Sun & Saenko, 2016). While the objective is similar, using ideas from domain adaptation for learning better representation, we address it in two different ways. Our method fundamentally differs from Song et al. (2018) since we do not enforce domain adaptation by adding specific constrains to the loss function. Instead, we let the network learn the domain invariant representation directly during the optimization process, as suggested by Ganin & Lempitsky (2015); Ganin et al. (2016). Moreover, Song et al. (2018) focused mainly of Fast Gradient Sign Method (FGSM) attack, which is a one step variant of PGD attack. We empirically demonstrate the superiority of our method in Section 4. In a concurrent work, Qian et al. (2021) utilized the idea of exploiting local and global data information, and suggested to generate the adversarial examples by attacking an additional domain classifier.

2.2 Theoretical analysis of robust generalization

Several works investigated the sample complexity requires the ensure adversarial generalization compared to the non-adversarial counterpart. Schmidt et al. (2018) has shown that there exists a distribution (mixture of Gaussians) where ensuring robust generalization necessarily requires more data than standard learning. This has been furthered investigated in a distribution-free models via the Rademacher complexity, VC dimension, and fat-shattering dimension (Yin et al., 2019; Attias et al., 2019; Khim & Loh, 2018; Awasthi et al., 2020; Cullina et al., 2018; Montasser et al., 2019; Tsai et al., 2021; Attias et al., 2022; Attias & Hanneke, 2022) and additional settings (Diochnos et al., 2018; Carmon et al., 2019).

3 Domain Invariant Adversarial Learning approach

In this section, we introduce our Domain Invariant Adversarial Learning (DIAL) approach for adversarial training. The source domain is the natural dataset, and the target domain is generated using adversarial attack on the natural domain. We aim to learn a model that has low error on the source (natural) task (e.g.,

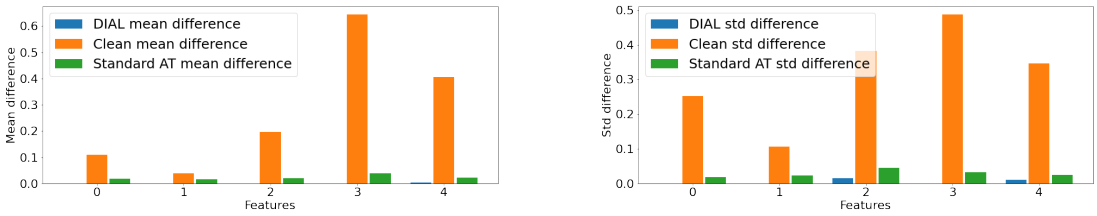
classification) while ensuring that the internal representation cannot discriminate between the natural and adversarial domains. In this way, we enforce additional regularization on the feature representation, which enhances robustness.

3.1 The benefits of invariant representation to adversarial examples

The motivation behind the proposed method is to enforce an invariant feature representation to adversarial perturbations. Given a natural example x and its adversarial counterpart x', if the domain classifier manages to distinguish between them, this means that the perturbation has induced a significant difference in the feature representation. We impose an additional loss on the natural and adversarial domains in order to discourage this behavior.

We demonstrate that the feature representation layer does not discriminate between natural and adversarial examples, namely $G_f(x;\theta_f) \approx G_f(x';\theta_f)$. Figure 2 presents the scaled mean and standard deviation (std) of the absolute differences between the natural examples from test and their corresponding adversarial examples on different features from the feature representation layer. Smaller differences in the mean and std imply a higher domain invariance — and indeed, DIAL achieves near-zero differences almost across the board. Moreover, DIAL's feature-level invariance almost consistently outperforms the naturally trained model (model trained without adversarial training), and the model trained using standard adversarial training techniques (Madry et al., 2017). We provide additional features visualizations in Appendix H.

Recently, other communities also discovered the benefits of adopting analogous architectures to DANN, such as the contrastive learning community which used similar architecture to improve representation learning (Dangovski et al., 2021; Wang et al., 2021)



(a) Mean difference comparison

(b) Standard deviation difference comparison

Figure 2: We visualize the (a) Mean and (b) standard deviation (std) differences comparison between three models: (1) Naturally trained model (without adversarial training), named Clean. (2) Model trained using standard adversarial training, named standard AT, and (3) Model trained using our method, DIAL. We visualize five random features from the features layer. Each bar represent the difference between the means/std of the natural examples and the mean/std of their corresponding adversarial examples on this same feature.

3.2 Model architecture and regularized loss function

Let us define the notation for our domain invariant robust architecture and loss. Let $G_f(\cdot;\theta_f)$ be the feature extractor neural network with parameters θ_f. Let $G_y(\cdot;\theta_y)$ be the label classifier with parameters θ_y, and let $G_d(\cdot;\theta_d)$ be the domain classifier with parameters θ_d. That is, $G_y(G_f(\cdot;\theta_f);\theta_y)$ is essentially the standard model (e.g., wide residual network (Zagoruyko & Komodakis, 2016)), while in addition, we have a domain classification layer to enforce a domain invariant on the feature representation. An illustration of the architecture is presented in Figure 1.

Given a training set $\{(x_i, y_i)\}_{i=1}^n$, the natural loss is defined as:

$$\mathcal{L}_{\text{nat}}^y = \frac{1}{n}\sum_{i=1}^n \text{CE}(G_y(G_f(x_i;\theta_f);\theta_y), y_i).$$

We consider two basic forms of the robust loss. One is the standard cross-entropy (CE) loss between the predicted probabilities and the actual label, which we refer to later as DIAL_{CE}. The second is the

Kullback-Leibler (KL) divergence between the adversarial and natural model outputs (logits), i.e., the class probabilities between the natural examples and their adversarial counterparts, which we refer to as DIAL_{KL}.

$$\mathcal{L}_{\text{rob}}^{\text{CE}} = \frac{1}{n}\sum_{i=1}^{n} \text{CE}(G_y(G_f(x_i';\theta_f);\theta_y), y_i),$$
$$\mathcal{L}_{\text{rob}}^{\text{KL}} = \frac{1}{n}\sum_{i=1}^{n} \text{KL}(G_y(G_f(x_i';\theta_f);\theta_y) \parallel G_y(G_f(x_i;\theta_f);\theta_y)).$$

where $\{(x_i', y_i)\}_{i=1}^{n}$ are the generated corresponding adversarial examples. Next, we define source domain label d_i as 0 (for natural examples) and target domain label $d_i^{'}$ as 1 (for adversarial examples). Then, the natural and adversarial domain losses are defined as:

$$\mathcal{L}_{\text{nat}}^{d} = \frac{1}{n}\sum_{i=1}^{n} \text{CE}(G_d(G_f(x_i;\theta_f);\theta_d), d_i),$$
$$\mathcal{L}_{\text{adv}}^{d} = \frac{1}{n}\sum_{i=1}^{n} \text{CE}(G_d(G_f(x_i^{'};\theta_f);\theta_d), d_i').$$

We can now define the full domain invariant robust loss:

$$\text{DIAL}_{\text{CE}} = \mathcal{L}_{\text{nat}}^{y} + \lambda\mathcal{L}_{\text{rob}}^{\text{CE}} - r(\mathcal{L}_{\text{nat}}^{d} + \mathcal{L}_{\text{adv}}^{d}),$$
$$\text{DIAL}_{\text{KL}} = \mathcal{L}_{\text{nat}}^{y} + \lambda\mathcal{L}_{\text{rob}}^{\text{KL}} - r(\mathcal{L}_{\text{nat}}^{d} + \mathcal{L}_{\text{adv}}^{d}).$$

The goal is to *minimize* the loss on the natural and adversarial classification while *maximizing* the loss for the domains. The *reversal-ratio* hyper-parameter r is inserted into the network layers as a gradient reversal layer (Ganin & Lempitsky, 2015; Ganin et al., 2016) that leaves the input unchanged during forward propagation and reverses the gradient by multiplying it with a negative scalar during the back-propagation. The reversal-ratio parameter is initialized to a small value and is gradually increased to r, as the main objective converges. This enforces a domain-invariant representation as the training progress: a larger value enforces a higher fidelity to the domain. A comprehensive algorithm description can be found in Appendix A.

3.3 Theoretical Analysis through the Lens of Generalization Bounds

Our method refers to the natural and adversarial examples as two distinct domains, where the source domain consists of natural examples and the target domain is the adversarially perturbed examples. Given this assumption, we can adapt the generalization bounds from Mansour et al. (2009) to theoretically justify our approach.

Preliminaries. Let H be a set of functions mapping X to Y and let $\mathcal{L} : Y \times Y \to \mathbb{R}_+$ be a loss function over Y. The $\mathcal{L}_Q(f, g)$ loss functional is defined as the expected loss for any two functions $f, g : X \to Y$ in H, and any distribution Q over X:

$$\mathcal{L}_Q(f, g) = \mathbb{E}_{x\sim Q}[L(f(x), g(x))].$$

We also denote $f_Q : X \to Y$ as the target function on examples drawn from Q. That is, the error of a function $h \in H$ is defined as $\mathcal{L}_Q(h, f_Q) = \mathbb{E}_{x\sim Q}[L(h(x), f_Q(x))]$.

We define the discrepancy distance disc_L between two distributions Q_1 and Q_2 over X by:

$$\text{disc}_L(Q_1, Q_2) = \max_{h,h'\in H} |\mathcal{L}_{Q_1}(h, h') - \mathcal{L}_{Q_2}(h, h')|.$$

Reduction to domain adaptation. Given source (natural) distribution D, we define the target (adversarial) distribution D_{adv}, such that every pair (x, y) in the support of D is mapped to (z, y), where $z \in \mathbb{B}_\epsilon(x)$ and $D_{\text{adv}}(z, y) = D(x, y)$.

Let $h_D^\star = \arg\min_{h\in H} \mathcal{L}_D(h, f_D)$ be the optimal natural function, and let $h_{D_{\text{adv}}}^\star = \arg\min_{h\in H} \mathcal{L}_{D_{\text{adv}}}(h, f_{D_{\text{adv}}})$ be the optimal robust function. Additionally, we assume that the labeling function $f_{D_{\text{adv}}}$ is the same as f_D.

Under these assumptions, we reduce our problem to a domain adaptation one, which consists of selecting a hypothesis $h \in H$ with a small expected loss according to the target distribution.

We can now adapt Theorem 8 in (Mansour et al., 2009) to our case and bound the adversarial loss by:

$$\begin{aligned}\mathcal{L}_{D_{\text{adv}}}(h, f_{D_{\text{adv}}}) \leq & \overbrace{\mathcal{L}_{D_{\text{adv}}}(h^\star_{D_{\text{adv}}}, f_{D_{\text{adv}}})}^{\text{adv}} + \overbrace{\mathcal{L}_D(h, h^\star_D)}^{\text{natural}} \\ & + \underbrace{\mathcal{L}_D(h^\star_D, h^\star_{D_{\text{adv}}})}_{\text{trade-off}} + \underbrace{\text{disc}_L(D_{\text{adv}}, D)}_{\text{discrepancy}}.\end{aligned} \tag{1}$$

The bound on the adversarial loss consists of four terms: *first term* - approximation error of H for adversarial distribution. *second term* - estimation error on natural examples for the output function on the true distribution (compared to the optimal $h \in H$). *third term* - trade-off error (depends on H, D, D_{adv}, and not the algorithm output) that represents the fraction of points for which the optimal $h \in H$ on adversarial examples are wrong, compared to the optimal $h \in H$ on clean examples. *forth term* - the discrepancy distance between D and D_{adv}.

Therefore, the adversarial loss bounded in equation 1 depends, in addition to the natural and adversarial losses, on the discrepancy distance, disc_L, between the two distributions. The discrepancy distance will decrease as the two distributions D and D_{adv} will be similar to one another.

Our adversarial training method, DIAL, minimizes the adversarial and natural losses, and simultaneously learns an invariant feature representation between the adversarial and natural domains. This representation ensures that adversarial examples and their natural counterparts will be invariant to the domain they were taken from, and therefore makes the two distributions D and D_{adv} similar to each other. *Altogether, DIAL also minimizes the discrepancy distance, which in turn leads to minimizing the upper bound on the adversarial error.*

3.4 Related work loss function comparison

In Table 1 we further illustrate the loss functions of the different methods. For $\text{TRADES}_{\text{AWP}}$, the process involves running TRADES loss function, and then running weight perturbation as defined in equation (10) in (Wu et al., 2020). These loss functions can be compared to our proposed loss functions DIAL_{CE} and DIAL_{KL} which are detailed in section 3.2, where the main difference lies in the new natural and adversarial domain losses. We colored in red the unique terms of our method.

4 Experiments

In this section we conduct comprehensive experiments to emphasise the effectiveness of DIAL, including evaluations under white-box and black-box settings, robustness to unforeseen adversaries, robustness to unforeseen corruptions, transfer learning, and ablation studies. Finally, we present a new measurement to test the balance between robustness and natural accuracy, which we named F_1-robust score.

4.1 A case study on SVHN and CIFAR-100

In the first part of our analysis, we conduct a case study experiment on two benchmark datasets: SVHN (Netzer et al., 2011) and CIFAR-100 Krizhevsky et al. (2009). We follow common experiment settings as in Rice et al. (2020); Wu et al. (2020). We used the PreAct ResNet-18 (He et al., 2016) architecture on which we integrate a domain classification layer. The adversarial training is done using 10-step PGD adversary with perturbation size of 0.031 and a step size of 0.003 for SVHN and 0.007 for CIFAR-100. The batch size is 128, weight decay is $7e^{-4}$ and the model is trained for 100 epochs. For SVHN, the initial learinnig rate is set to 0.01 and decays by a factor of 10 after 55, 75 and 90 iteration. For CIFAR-100, the initial learning rate is set to 0.1 and decays by a factor of 10 after 75 and 90 iterations. Results are averaged over 3 restarts while omitting one standard deviation (which is smaller than 0.2% in all experiments). As can be seen by the results in Tables 2 and 3, DIAL presents consistent improvement in robustness (e.g., 5.75% improved robustness on SVHN against AA) compared to the standard AT while also improving the natural accuracy. More results are presented in Appendix B.

Table 1: Loss function comparison. Let (x,y) be the natural example and its corresponding label. Let x' be the adversarial example generated from x. CE refers to the cross-entropy loss function, KL refers to the KL-divergence loss function, and BCE is the boosted cross-entropy loss function. $\mathcal{L}_{CORAL}$ and $\mathcal{L}_{MMD}$ correspond to the correlation alignment and maximum mean discrepancy (Borgwardt et al., 2006; Sun & Saenko, 2016), respectively. $\mathcal{L}_{margin}$ minimize the intra-class variations and maximize the inter-class variations (Song et al., 2018). λ is a hyper parameters to control the ratio between different losses, and r is the reversal ratio hyper parameter. Let $G_f(\cdot;\theta_f)$ be the feature extractor neural network with parameters θ_f. Let $G_y(\cdot;\theta_y)$ be the label classifier with parameters θ_y, and let $G_d(\cdot;\theta_d)$ be the domain classifier with parameters θ_d. That is, $G(\cdot;\theta) = G_y(G_f(\cdot;\theta_f);\theta_y)$ is essentially the standard model definition (e.g., wide residual network).We define source domain label d as 0 (for natural examples) and target domain label $d^{'}$ as 1 (for adversarial examples). For convenience, we present the loss function on a single example.

Method	Loss function
AT	$\mathrm{CE}(G(x';\theta),y)$
TRADES	$\mathrm{CE}(G(x;\theta),y) + \lambda \cdot \mathrm{KL}(G(x';\theta) \parallel G(x;\theta))$
MART	$\mathrm{BCE}(G(x';\theta),y) + \lambda \cdot \mathrm{KL}(G(x';\theta) \parallel G(x;\theta)) \cdot (1 - G(x;\theta)_y)$
ATDA	$\mathrm{CE}(G(x';\theta),y) + \mathrm{CE}(G(x;\theta),y) + \mathcal{L}_{CORAL} + \mathcal{L}_{MMD} + \mathcal{L}_{margin}$
$\mathrm{DIAL_{CE}}$	$\mathrm{CE}(G(x;\theta),y) + \lambda \cdot \mathrm{CE}(G(x';\theta),y) - r(\mathrm{CE}(G_d(G_f(x;\theta_f);\theta_d),d) + \mathrm{CE}(G_d(G_f(x';\theta_f);\theta_d),d'))$
$\mathrm{DIAL_{KL}}$	$\mathrm{CE}(G(x;\theta),y) + \lambda \cdot \mathrm{KL}(G(x';\theta) \parallel G(x;\theta)) - r(\mathrm{CE}(G_d(G_f(x;\theta_f);\theta_d),d) + \mathrm{CE}(G_d(G_f(x';\theta_f);\theta_d),d'))$

Table 2: Robustness against white-box, black-box attacks and Auto-Attack (AA) on SVHN. Black-box attacks are generated using naturally trained surrogate model. Natural represents the naturally trained (non-adversarial) model.

		White-box				Black-Box				
Defense Model	Natural	PGD^{20}	PGD^{100}	PGD^{1000}	CW^{∞}	PGD^{20}	PGD^{100}	PGD^{1000}	CW^{∞}	AA
NATURAL	96.85	0	0	0	0	0	0	0	0	0
AT	89.90	53.23	49.45	49.23	48.25	86.44	86.28	86.18	86.42	45.25
$\mathrm{DIAL_{KL}}$ (Ours)	90.66	**58.91**	**55.30**	**55.11**	**53.67**	87.62	87.52	87.41	87.63	**51.00**
$\mathrm{DIAL_{CE}}$ (Ours)	**92.88**	55.26	50.82	50.54	49.66	**89.12**	**89.01**	**88.74**	**89.10**	46.52

Table 3: Robustness against white-box, black-box attacks and Auto-Attack (AA) on CIFAR100. Black-box attacks are generated using naturally trained surrogate model. Natural represents the naturally trained (non-adversarial) model.

		White-box				Black-Box				
Defense Model	Natural	PGD^{20}	PGD^{100}	PGD^{1000}	CW^{∞}	PGD^{20}	PGD^{100}	PGD^{1000}	CW^{∞}	AA
NATURAL	79.30	0	0	0	0	0	0	0	0	0
AT	56.73	29.57	28.45	28.39	26.6	55.52	55.29	55.26	55.40	24.12
$\mathrm{DIAL_{KL}}$ (Ours)	58.47	**31.19**	**30.50**	**30.42**	**26.91**	57.16	56.81	56.80	57.00	**25.87**
$\mathrm{DIAL_{CE}}$ (Ours)	**60.77**	27.87	26.66	26.61	25.98	**59.48**	**59.06**	**58.96**	**59.20**	23.51

4.2 Performance comparison on CIFAR-10

In this part, we evaluate the performance of DIAL compared to other well-known methods on CIFAR-10. We follow the same experiment setups as in Madry et al. (2017); Wang et al. (2019b); Zhang et al. (2019b). When experiment settings are not identical between tested methods, we choose the most commonly used settings,

and apply it to all experiments. This way, we keep the comparison as fair as possible and avoid reporting changes in results which are caused by inconsistent experiment settings (Pang et al., 2020a). To show that our results are not caused because of what is referred to as *obfuscated gradients* (Athalye et al., 2018), we evaluate our method with same setup as in our defense model, under strong attacks (e.g., PGD^{1000}) in both white-box, black-box settings, Auto-Attack (Croce & Hein, 2020), unforeseen "natural" corruptions (Hendrycks & Dietterich, 2018), and unforeseen adversaries. To make sure that the reported improvements are not caused by *adversarial overfitting* (Rice et al., 2020), we report best robust results for each method on average of 3 restarts, while omitting one standard deviation (which is smaller than 0.2% in all experiments). Additional results for CIFAR-10 as well as comprehensive evaluation on MNIST can be found in Appendix D and E.

Table 4: Robustness against white-box, black-box attacks and Auto-Attack (AA) on CIFAR-10. Black-box attacks are generated using naturally trained surrogate model. Natural represents the naturally trained (non-adversarial) model.

		White-box			Black-Box			
Defense Model	Natural	PGD^{20}	PGD^{100}	CW^{∞}	PGD^{20}	PGD^{100}	CW^{∞}	AA
NATURAL	95.43	0	0	0	0	0	0	0
TRADES	84.92	56.60	55.56	54.20	84.08	83.89	83.91	53.08
MART	83.62	58.12	56.48	53.09	82.82	82.52	82.80	51.10
AT	85.10	56.28	54.46	53.99	84.22	84.14	83.92	51.52
ATDA	76.91	43.27	41.13	41.01	75.59	75.37	75.35	40.08
DIAL_{KL} (Ours)	85.25	**58.43**	**56.80**	**55.00**	84.30	84.18	84.05	**53.75**
DIAL_{CE} (Ours)	**89.59**	54.31	51.67	52.04	**88.60**	**88.39**	**88.44**	49.85
DIAL_{AWP} (Ours)	**85.91**	**61.10**	**59.86**	**57.67**	**85.13**	**84.93**	**85.03**	**56.78**
$\text{TRADES}_{\text{AWP}}$	85.36	59.27	59.12	57.07	84.58	84.58	84.59	56.17

CIFAR-10 setup. We use the wide residual network (WRN-34-10) (Zagoruyko & Komodakis, 2016) architecture. Sidelong this architecture, we integrate a domain classification layer. To generate the adversarial domain dataset, we use a perturbation size of $\epsilon = 0.031$. We apply 10 of inner maximization iterations with perturbation step size of 0.007. Batch size is set to 128, weight decay is set to $7e^{-4}$, and the model is trained for 100 epochs. Similar to the other methods, the initial learning rate was set to 0.1, and decays by a factor of 10 at iterations 75 and 90. See Appendix C for additional details.

Table 5: Black-box attack using the adversarially trained surrogate models on CIFAR-10.

Surrogate (source) model	Target model	robustness %
TRADES	DIAL_{CE}	**67.77**
DIAL_{CE}	TRADES	65.75
MART	DIAL_{CE}	**70.30**
DIAL_{CE}	MART	64.91
AT	DIAL_{CE}	**65.32**
DIAL_{CE}	AT	63.54
ATDA	DIAL_{CE}	**66.77**
DIAL_{CE}	ATDA	52.56

White-box/Black-box robustness. As reported in Table 4 and Appendix E, our method achieves better robustness compared to the other methods. Specifically, in the white-box settings, our method improves robustness over Madry et al. (2017) and TRADES by 2% while keeping higher natural accuracy. We also observe better natural accuracy of 1.65% over MART while also achieving better robustness over all attacks. Moreover, our method presents significant improvement of up to 15% compared to the the domain invariant

method suggested by Song et al. (2018) (ATDA). When incorporating AWP, our method improves the results of $\text{TRADES}_{\text{AWP}}$ by almost 2%. When tested on black-box settings, DIAL_{CE} presents a significant improvement of more than 4.4% over the second-best performing method, and up to 13%. In Table 5, we also present the black-box results when the source model is taken from one of the adversarially trained models. In addition to the improvement in black-box robustness, DIAL_{CE} also manages to achieve better clean accuracy of more than 4.5% over the second-best performing method.

4.2.1 Robustness to Unforeseen Attacks and Corruptions

Unforeseen Adversaries. To further demonstrate the effectiveness of our approach, we test our method against various adversaries that were not used during the training process. We attack the model under the white-box settings with ℓ_2-PGD, ℓ_1-PGD, ℓ_∞-DeepFool and ℓ_2-DeepFool (Moosavi-Dezfooli et al., 2016) adversaries using Foolbox (Rauber et al., 2017). We applied commonly used attack budget with 20 and 50 iterations for PGD and DeepFool, respectively. Results are presented in Table 6. As can be seen, our approach gains an improvement of up to 4.73% over the second best method under the various attack types and an average improvement of 3.7% over all threat models.

Table 6: Robustness on CIFAR-10 against unseen adversaries under white-box settings.

Threat Model	Attack Constraints	DIAL_{KL}	DIAL_{CE}	AT	TRADES	MART	ATDA
ℓ_2-PGD	$\epsilon = 0.5$	76.05	**80.51**	76.82	76.57	75.07	66.25
	$\epsilon = 0.25$	80.98	**85.38**	81.41	81.10	80.04	71.87
ℓ_1-PGD	$\epsilon = 12$	74.84	**80.00**	76.17	75.52	75.95	65.76
	$\epsilon = 7.84$	78.69	**83.62**	79.86	79.16	78.55	69.97
ℓ_2-DeepFool	overshoot=0.02	84.53	**88.88**	84.15	84.23	82.96	76.08
ℓ_∞-DeepFool	overshoot=0.02	68.43	**69.50**	67.29	67.60	66.40	57.35

Unforeseen Corruptions. We further demonstrate that our method consistently holds against unforeseen "natural" corruptions, consists of 18 unforeseen diverse corruption types proposed by Hendrycks & Dietterich (2018) on CIFAR-10, which we refer to as CIFAR10-C. The CIFAR10-C benchmark covers noise, blur, weather, and digital categories. As can be shown in Figure 3, our method gains a significant and consistent improvement over all the other methods. Our method leads to an average improvement of 4.7% with minimum improvement of 3.5% and maximum improvement of 5.9% compared to the second best method over all unforeseen attacks. See Appendix F for the full experiment results.

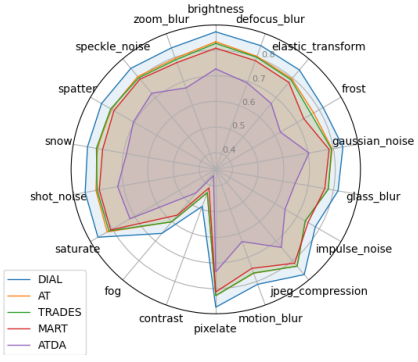


Figure 3: Accuracy comparison over all unforeseen corruptions.

4.2.2 Transfer Learning

Recent works (Salman et al., 2020; Utrera et al., 2020) suggested that robust models transfer better on standard downstream classification tasks. In Table 7 we demonstrate the advantage of our method when applied for transfer learning across CIFAR10 and CIFAR100 using the common linear evaluation protocol. see Appendix G for detailed settings.

Table 7: Transfer learning results comparison.

		Target	
Source	Defence Model	CIFAR10	CIFAR100
CIFAR10	DIAL	-	**28.57**
	AT		26.95
	TRADES		25.40
CIFAR100	DIAL	**73.68**	-
	AT	71.41	
	TRADES	71.42	

4.2.3 Modularity and Ablation Studies

We note that the domain classifier is a modular component that can be integrated into existing models for further improvements. Removing the domain head and related loss components from the different DIAL formulations results in some common adversarial training techniques. For DIAL_{KL}, removing the domain and related loss components results in the formulation of TRADES. For DIAL_{CE}, removing the domain and related loss components results in the original formulation of the standard adversarial training, and for DIAL_{AWP} the removal results in $\text{TRADES}_{\text{AWP}}$. Therefore, the ablation studies will demonstrate the effectiveness of combining DIAL on top of different adversarial training methods.

We investigate the contribution of the additional domain head component introduced in our method. Experiment configuration are as in 4.2, and robust accuracy is based on white-box PGD^{20} on CIFAR-10 dataset. We remove the domain head from both DIAL_{KL}, DIAL_{AWP}, and DIAL_{CE} (equivalent to $r = 0$) and report the natural and robust accuracy. We perform 3 random restarts and omit one standard deviation from the results. Results are presented in Figure 4. All DIAL variants exhibits stable improvements on both natural accuracy and robust accuracy. DIAL_{CE}, DIAL_{KL}, and DIAL_{AWP} present an improvement of 1.82%, 0.33%, and 0.55% on natural accuracy and an improvement of 2.5%, 1.87%, and 0.83% on robust accuracy, respectively. This evaluation empirically demonstrates the benefits of incorporating DIAL on top of different adversarial training techniques.

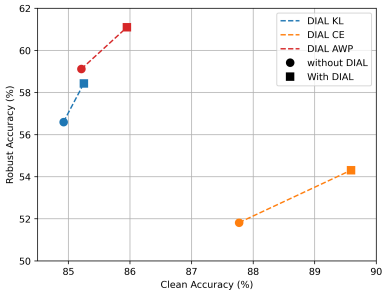
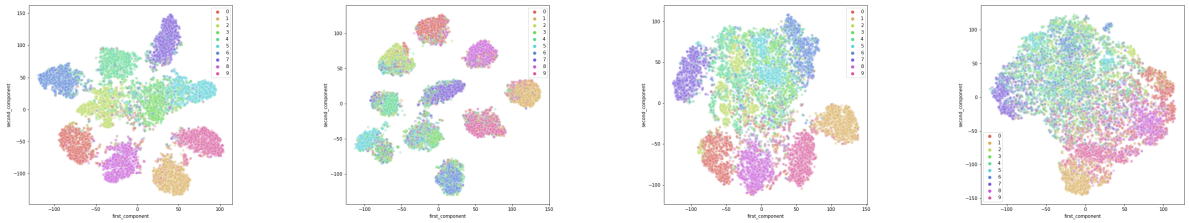


Figure 4: Ablation studies for DIAL_{KL}, DIAL_{CE}, and DIAL_{AWP} on CIFAR-10. Circle represent the robust-natural accuracy without using DIAL, and square represent the robust-natural accuracy when incorporating DIAL.

4.2.4 Visualizing DIAL

To further illustrate the superiority of our method, we visualize the model outputs from the different methods on both natural and adversarial test data. Figure 5 shows the embedding received after applying t-SNE (Van der Maaten & Hinton, 2008) with two components on the model output for our method and for TRADES. DIAL seems to preserve strong separation between classes on both natural test data and adversarial test data. Additional illustrations for the other methods are attached in Appendix H.



(a) **DIAL** on natural logits (b) **DIAL** on adversarial logits (c) **TRADES** on natural logits (d) **TRADES** on adversarial logits

Figure 5: t-SNE embedding of model output (logits) into two-dimensional space for DIAL and TRADES using the CIFAR-10 natural test data and the corresponding PGD^{20} generated adversarial examples.

4.3 Balanced measurement for robust-natural accuracy

One of the goals of our method is to better balance between robust and natural accuracy under a given model. For a balanced metric, we adopt the idea of F_1-score, which is the harmonic mean between the precision and recall. However, rather than using precision and recall, we measure the F_1-score between robustness and natural accuracy, using a measure we call the $\mathbf{F_1}$**-robust** score.

$$F_1\text{-robust} = \frac{\text{true\_robust}}{\text{true\_robust} + \frac{1}{2}(\text{false\_robust} + \text{false\_natural})}, \tag{2}$$

where true_robust are the adversarial examples that were correctly classified, false_robust are the adversarial examples that were miss-classified, and false_natural are the natural examples that were miss-classified. Results are presented in Table 8 and demonstrate that our method achieves the best F_1-robust score in both settings, which supports our findings from previous sections.

Table 8: F_1-robust measurement using PGD^{20} attack in white and black box settings on CIFAR-10.

	TRADES	MART	AT	ATDA	DIAL_{CE}	DIAL_{KL}	DIAL_{AWP}	$\text{TRADES}_{\text{AWP}}$
White-box	0.659	0.666	0.657	0.518	0.660	**0.675**	**0.698**	0.682
Black-box	0.844	0.831	0.845	0.761	**0.890**	0.847	**0.854**	0.849

5 Conclusion and Future Work

In this paper, we investigated the hypothesis that domain invariant representation can be beneficial for robust learning. With this idea in mind, we proposed a new adversarial learning method, called *Domain Invariant Adversarial Learning* (DIAL) that incorporates domain adversarial neural network into the adversarial training process. The proposed method, DIAL, is theoretically motivated by the domain adaptation generalization bounds. DIAL is generic and can be combined with any network architecture and any adversarial training technique in a wide range of tasks. Additionally, since the domain classifier does not require the class labels, we argue that additional unlabeled data can be leveraged in future work. Our evaluation process included

strong adversaries, unforeseen adversaries, unforeseen corruptions, transfer learning tasks, and ablation studies. Using the extensive empirical analysis, we demonstrate the significant and consistent improvement obtained by DIAL in both robustness and natural accuracy.

6 Broader Impact Statement

Adversarial examples illustrate a fundamental vulnerability of deep neural networks, and manage to break state-of-the art DNNs in various fields. As these DNNs are deployed in critical systems, such as autonomous vehicles and facial recognition systems, it becomes crucial to build models which are robust against such attacks. These kind of system cannot tolerate attacks that can cost in human lives. For this reason, we proposed DIAL to improve models' robustness against adversarial attacks. We hope that it will help in building more secure models for real-world applications. DIAL is comparable to the state-of-the-art methods we tested in terms of training times and other resources. That said, this work is not without limitations: adversarial training is still a computationally expensive procedure that requires extra computations compared to standard training, with the concomitant environmental costs. Even though incorporating our method introduced improved standard accuracy, adversarial training still degrades the standard accuracy. Moreover, models are trained to be robust using well known threat models such as the bounded ℓ_p norms. However, once a model is deployed, we cannot control the type of attacks it faces from sophisticated adversaries. Thus, the general problem is still very far from being fully solved.

References

Rima Alaifari, Giovanni S Alberti, and Tandri Gauksson. Adef: an iterative algorithm to construct adversarial deformations. *arXiv preprint arXiv:1804.07729*, 2018.

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *arXiv preprint arXiv:2007.02617*, 2020.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pp. 274–283. PMLR, 2018.

Idan Attias and Steve Hanneke. Adversarially robust learning of real-valued functions. *arXiv preprint arXiv:2206.12977*, 2022.

Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for robust learning. In *Algorithmic Learning Theory*, pp. 162–183. PMLR, 2019.

Idan Attias, Steve Hanneke, and Yishay Mansour. A characterization of semi-supervised adversarially-robust pac learnability. *arXiv preprint arXiv:2202.05420*, 2022.

Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. On the rademacher complexity of linear hypothesis sets. *arXiv preprint arXiv:2007.11045*, 2020.

Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021.

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22 (14):e49–e57, 2006.

Tom B Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano, and Ian Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018.

Qi-Zhi Cai, Min Du, Chang Liu, and Dawn Song. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14, 2017a.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017b.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019.

Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 699–708, 2020.

Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit Dhillon, and Cho-Jui Hsieh. Cat: Customized adversarial training for improved robustness. *arXiv preprint arXiv:2002.06789*, 2020.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. Pac-learning in the presence of adversaries. In *Advances in Neural Information Processing Systems*, pp. 230–241, 2018.

Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic. Equivariant self-supervised learning: Encouraging equivariance in representations. In *International Conference on Learning Representations*, 2021.

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

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*, 2018.

Dimitrios Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody. Adversarial risk and robustness: General definitions and implications for the uniform distribution. In *Advances in Neural Information Processing Systems*, pp. 10359–10368, 2018.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. 2018.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

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.

Justin Gilmer, Ryan P Adams, Ian Goodfellow, David Andersen, and George E Dahl. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.

Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3996–4003, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Minghao Guo, Yuzhe Yang, Rui Xu, Ziwei Liu, and Dahua Lin. When nas meets robustness: In search of robust architectures against adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 631–640, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018.

Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. In *NeurIPS*, 2020.

Gaojie Jin, Xinping Yi, Wei Huang, Sven Schewe, and Xiaowei Huang. Enhancing adversarial training with second-order statistics of weights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15273–15283, 2022.

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

Justin Khim and Po-Ling Loh. Adversarial risk bounds for binary classification via function transformation. *arXiv preprint arXiv:1810.09519*, 2, 2018.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016a.

Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016b.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Saehyung Lee, Hyungyu Lee, and Sungroh Yoon. Adversarial vertex mixup: Toward better adversarially robust generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 272–281, 2020.

Saehyung Lee, Changhwa Park, Hyungyu Lee, Jihun Yi, Jonghyun Lee, and Sungroh Yoon. Removing undesirable feature contributions using out-of-distribution data. *arXiv preprint arXiv:2101.06639*, 2021.

Aishan Liu, Shiyu Tang, Xianglong Liu, Xinyun Chen, Lei Huang, Zhuozhuo Tu, Dawn Song, and Dacheng Tao. Towards defending multiple adversarial perturbations via gated batch normalization. *arXiv preprint arXiv:2012.01654*, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory*, pp. 2512–2530, 2019.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pp. 4970–4979. PMLR, 2019.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*, 2020a.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Jun Zhu, and Hang Su. Boosting adversarial training with hypersphere embedding. *arXiv preprint arXiv:2002.08619*, 2020b.

Vinay Uday Prabhu, Dian Ang Yap, Joyce Xu, and John Whaley. Understanding adversarial robustness through loss landscape geometries. *arXiv preprint arXiv:1907.09061*, 2019.

Zhuang Qian, Shufei Zhang, Kaizhu Huang, Qiufeng Wang, Rui Zhang, and Xinping Yi. Improving model robustness with latent distribution locally and globally. *arXiv preprint arXiv:2107.04401*, 2021.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018a.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Semidefinite relaxations for certifying robustness to adversarial examples. *arXiv preprint arXiv:1811.01057*, 2018b.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning*, 2017. URL `http://arxiv.org/abs/1707.04131`.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021a.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, 34, 2021b.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.

Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4322–4330, 2019.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *arXiv preprint arXiv:2007.08489*, 2020.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Mądry. Adversarially robust generalization requires more data. *arXiv preprint arXiv:1804.11285*, 2018.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2, 2017.

Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Improving the generalization of adversarial training with domain adaptation. *arXiv preprint arXiv:1810.00740*, 2018.

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

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Pedro Tabacof and Eduardo Valle. Exploring the space of adversarial images. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 426–433. IEEE, 2016.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Yu-Lin Tsai, Chia-Yi Hsu, Chia-Mu Yu, and Pin-Yu Chen. Formalizing generalization and robustness of neural networks to weight perturbations. *arXiv preprint arXiv:2103.02200*, 2021.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Robert Stanforth, Alhussein Fawzi, and Pushmeet Kohli. Are labels required for improving adversarial robustness? *arXiv preprint arXiv:1905.13725*, 2019.

Francisco Utrera, Evan Kravitz, N Benjamin Erichson, Rajiv Khanna, and Michael W Mahoney. Adversarially-trained deep nets transfer better. *arXiv preprint arXiv:2007.05869*, 2020.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Yifei Wang, Zhengyang Geng, Feng Jiang, Chuming Li, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Residual relaxation for multi-view representation learning. *Advances in Neural Information Processing Systems*, 34:12104–12115, 2021.

Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, volume 1, pp. 2, 2019a.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019b.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2018.

Eric Wong, Frank R Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. *arXiv preprint arXiv:1805.12514*, 2018.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33, 2020.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018.

Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. *arXiv preprint arXiv:1906.03787*, 2019.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 501–509, 2019a.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2730–2739, 2019b.

Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. Dverge: diversifying vulnerabilities for enhanced robust generation of ensembles. *arXiv preprint arXiv:2009.14720*, 2020.

Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *International Conference on Machine Learning*, pp. 7085–7094. PMLR, 2019.

Fuxun Yu, Chenchen Liu, Yanzhi Wang, Liang Zhao, and Xiang Chen. Interpreting adversarial robustness: A view from decision surface in input space. *arXiv preprint arXiv:1810.00144*, 2018.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. *arXiv preprint arXiv:1905.00877*, 2019a.

Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. *Advances in Neural Information Processing Systems*, 32:1831–1841, 2019.

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*, pp. 7472–7482. PMLR, 2019b.

Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International Conference on Machine Learning*, pp. 11278–11287. PMLR, 2020.

Algorithm 1 Domain Invariant Adversarial Learning

Input: Source data $S = \{(x_i, y_i)\}_{i=1}^{n}$ and network architecture G_f, G_y, G_d
Parameters: Batch size m, perturbation size ϵ, pgd attack step size τ, adversarial trade-off λ, initial reversal ratio r, and step size α
Init: Y_0 and Y_1 source and target domain vectors filled with 0 and 1 respectively
Output: Robust network $G = (G_f, G_y, G_d)$ parameterized by $\hat{\theta} = (\theta_f, \theta_y, \theta_d)$ respectively
repeat
  Fetch mini-batch $X_s = \{x_j\}_{j=1}^{m}$, $Y_s = \{y_j\}_{j=1}^{m}$
  # Generate adversarial target domain batch X_t
  for $j = 1, \ldots, m$ (in parallel) **do**
    $x'_j \leftarrow PGD(x_j, y_j, \epsilon, \tau)$
    $X_t \leftarrow X_t + x'_j$
  end for
  ℓ_s^y, $\ell_t^y \leftarrow \mathrm{CE}(G_y(G_f(X_s)), Y_s)$, $\mathrm{CE}(G_y(G_f(X_t)), Y_s)$
  ℓ_s^d, $\ell_t^d \leftarrow \mathrm{CE}(G_d(G_f(X_s)), Y_0)$, $\mathrm{CE}(G_d(G_f(X_t)), Y_1)$
  $\ell \leftarrow \ell_s^y + \lambda \ell_t^y - r(\ell_s^d + \ell_t^d)$
  $\hat{\theta} \leftarrow \hat{\theta} - \alpha \nabla_{\hat{\theta}}(\ell)$
until stopping criterion is not met

A Domain Invariant Adversarial Learning Algorithm

Algorithm 1 describes a pseudo-code of our proposed $\mathrm{DIAL}_{\mathrm{CE}}$ variant. As can be seen, a target domain batch is not given in advance as with standard domain-adaptation task. Instead, for each natural batch we generate a target batch using adversarial training. The loss function is composed of natural and adversarial losses with respect to the main task (e.g., classification), and from natural and adversarial domain losses. By maximizing the losses on the domain we aim to learn a feature representation which is invariant to the natural and adversarial domain, and therefore more robust.

B additional results on CIFAR-100 and SVHN

Table 9: Robustness against white-box, black-box attacks and Auto-Attack (AA) on SVHN. Black-box attacks are generated using naturally trained surrogate model and applied to the best performing robust models.

		White-box				Black-Box				
Defense Model	Natural	PGD^{20}	PGD^{100}	PGD^{1000}	CW^{∞}	PGD^{20}	PGD^{100}	PGD^{1000}	CW^{∞}	AA
TRADES	90.35	57.10	54.13	54.08	52.19	86.89	86.73	86.57	86.70	49.5
$\mathrm{DIAL}_{\mathrm{KL}}$ (Ours)	90.66	**58.91**	**55.30**	**55.11**	**53.67**	87.62	87.52	87.41	87.63	**51.00**
$\mathrm{DIAL}_{\mathrm{CE}}$ (Ours)	**92.88**	55.26	50.82	50.54	49.66	**89.12**	**89.01**	**88.74**	**89.10**	46.52

Table 10: Robustness against white-box, black-box attacks and Auto-Attack (AA) on CIFAR100. Black-box attacks are generated using naturally trained surrogate model and applied to the best performing robust models.

		White-box				Black-Box				
Defense Model	Natural	PGD^{20}	PGD^{100}	PGD^{1000}	CW^{∞}	PGD^{20}	PGD^{100}	PGD^{1000}	CW^{∞}	AA
TRADES	58.24	30.1	29.66	29.64	25.97	57.05	56.71	56.67	56.77	24.92
$\mathrm{DIAL}_{\mathrm{KL}}$ (Ours)	58.47	**31.19**	**30.50**	**30.42**	**26.91**	57.16	56.81	56.80	57.00	**25.87**
$\mathrm{DIAL}_{\mathrm{CE}}$ (Ours)	**60.77**	27.87	26.66	26.61	25.98	**59.48**	**59.06**	**58.96**	**59.20**	23.51

C CIFAR-10 Additional Experimental Setup details

Additional defence setup. For being consistent with other methods, the natural images are padded with 4-pixel padding with 32-random crop and random horizontal flip. Furthermore, all methods are trained using SGD with momentum 0.9. For DIAL_{KL}, we balance the robust loss with $\lambda = 6$ and the domains loss with $r = 4$. For DIAL_{CE}, we balance the robust loss with $\lambda = 1$ and the domains loss with $r = 2$. For DIAL-AWP, we used the same learning rate schedule used in Wu et al. (2020), where the initial 0.1 learning rate decays by a factor of 10 after 100 and 150 iterations. For black-box attacks, we used two types of surrogate models (1) naturally trained surrogate model, with natural accuracy of 95.61% and (2) surrogate model trained using one of the adversarial training methods.

D Benchmarking the State-of-the-art on MNIST

Defence setup. We use the same CNN architecture as used in Zhang et al. (2019b) which consists of four convolutional layers and three fully-connected layers. Sidelong this architecture, we integrate a domain classification layer. To generate the adversarial domain dataset, we use a perturbation size of $\epsilon = 0.3$. We apply 40 iterations of inner maximization with perturbation step size of 0.01. Batch size is set to 128 and the model is trained for 100 epochs. Similar to the other methods, the initial learning rate was set to 0.01, and decays by a factor of 10 after 55 iterations, 75 and 90 iterations. All the models in the experiment are trained using SGD with momentum 0.9. For our method, we balance the robust loss with $\lambda = 6$ and the domains loss with $r = 0.1$.

White-box/Black-box robustness. We evaluate all defense models using PGD^{40}, PGD^{100}, PGD^{1000} and CW_∞ (ℓ_∞ version of Carlini & Wagner (2017b) attack optimized by PGD-100) with step size 0.01. We constrain all attacks by the same perturbation $\epsilon = 0.3$. For our black-box setting, we use a naturally trained surrogate model with natural accuracy of 99.51%. As reported in Table 11, our method achieves improved robustness over the other methods under the different attack types, while preserving the same level of natural accuracy, and even surpassing the naturally trained model. We should note that in general, the improvement margin on MNIST is more moderate compared to CIFAR-10, since MNIST is an easier task than CIFAR-10 and the robustness range is already high to begin with. Additional results are available in Appendix E.

Table 11: Robustness against white-box, black-box attacks and Auto-Attack (AA) on MNIST. Black-box attacks are generated using naturally trained surrogate model and applied to the best performing robust models.

		White-box			Black-Box			
Defense Model	Natural	PGD^{40}	PGD^{100}	CW^∞	PGD^{40}	PGD^{100}	CW^∞	AA
TRADES	99.48	96.07	95.52	95.69	98.12	97.86	98.21	92.79
MART	99.38	96.99	96.11	95.98	98.16	97.96	98.28	93.30
AT	99.41	96.01	95.49	95.78	98.05	97.73	98.20	88.50
ATDA	98.72	96.82	96.26	96.31	97.74	97.28	97.76	93.31
DIAL_{KL} (Ours)	99.46	97.05	96.06	96.17	98.14	97.83	98.14	**93.68**
DIAL_{CE} (Ours)	**99.52**	**97.61**	**96.91**	**97.00**	**98.41**	**98.12**	**98.48**	93.43

E Additional Results on MNIST and CIFAR-10

In Table 12 we present additional results using the PGD^{1000} threat model. We use step size of 0.003 and constrain the attacks by the same perturbation $\epsilon = 0.031$. Table 13 presents a comparison of our method combined with AWP to other the variants of AWP that were presented in Wu et al. (2020). In addition, in Table 14 we add the F_1-robust scores for different variants of AWP.

Table 12: PGD^{1000} attack on MNIST and CIFAR-10 on white-box and black-box settings.

	MNIST		CIFAR-10	
Defense Model	White-box	Black-box	White-box	Black-box
TRADES	95.22	97.81	56.43	83.80
MART	95.74	97.89	56.55	82.47
AT	95.36	97.78	54.40	83.96
ATDA	96.20	97.34	41.02	75.11
$DIAL_{CE}$ (Ours)	**96.78**	**98.10**	51.57	**88.22**
$DIAL_{KL}$ (Ours)	95.99	97.89	**56.73**	84.00

Table 13: Robustness comparison of DIAL-AWP and other variants of AWP that do not require additional data under the ℓ_∞ threat model.

Defense Model	Natural	PGD^{20}	PGD^{100}	CW_∞	AA
DIAL-AWP (Ours)	**85.91**	**61.10**	**59.86**	**57.67**	**56.78**
TRADES-AWP (Wu et al., 2020)	85.36	59.27	59.12	57.07	56.17
MART-AWP (Wu et al., 2020)	84.43	60.68	59.32	56.37	54.23
AT-AWP (Wu et al., 2020)	85.57	58.14	57.94	55.96	54.04

Table 14: F_1-robust measurement on AWP variants based on white-box attack.

Defense Model	F_1-robust
DIAL-AWP (Ours)	**0.69753**
TRADES-AWP (Wu et al., 2020)	0.68162
MART-AWP (Wu et al., 2020)	0.68857
AT-AWP (Wu et al., 2020)	0.67381

F Extened results on Unforeseen Corruptions

We present full accuracy results against unforeseen corruptions in Tables 15 and 16.

Table 15: Accuracy (%) against unforeseen corruptions.

Defense Model	brightness	defocus blur	fog	glass blur	jpeg compression	motion blur	saturate	snow	speckle noise
TRADES	82.63	80.04	60.19	78.00	82.81	76.49	81.53	80.68	80.14
MART	80.76	78.62	56.78	76.60	81.26	74.58	80.74	78.22	79.42
AT	83.30	80.42	60.22	77.90	82.73	76.64	82.31	80.37	80.74
ATDA	72.67	69.36	45.52	64.88	73.22	63.47	72.07	68.76	72.27
DIAL (Ours)	**87.14**	**84.84**	**66.08**	**81.82**	**87.07**	**81.20**	**86.45**	**84.18**	**84.94**

Table 16: Accuracy (%) against unforeseen corruptions.

Defense Model	contrast	elastic transform	frost	gaussian noise	impulse noise	pixelate	shot noise	spatter	zoom blur
TRADES	43.11	79.11	76.45	79.21	73.72	82.73	80.42	80.72	78.97
MART	41.22	77.77	73.07	78.30	74.97	81.31	79.53	79.28	77.8
AT	43.30	79.58	77.53	79.47	73.76	82.78	80.86	80.49	79.58
ATDA	36.06	67.06	62.56	70.33	64.63	73.46	72.28	70.50	67.31
DIAL (Ours)	**48.84**	**84.13**	**81.76**	**83.76**	**78.26**	**87.24**	**85.13**	**84.84**	**83.93**

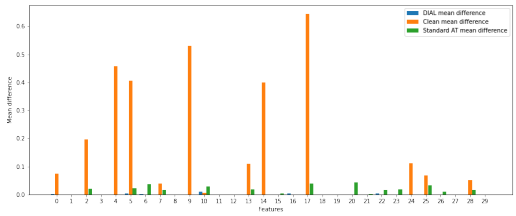
G Transfer Learning Settings

The models used are the same models from previous experiments. We follow the common procedure of "fixed-feature" setting, where only a linear layer on top of the pre-trained network is trained. We train a linear classifier on CIFAR-100 on top of the pre-trained network which was trained on CIFAR-10. We also train a linear classifier on CIFAR-10 on top of the pre-trained nwork which was trained on CIFAR-100. We train the linear classifier for 100 epochs, and an initial learning rate of 0.1 which is decayed by a factor of 10 at epochs 50 and 75. We used SGD optimizer with momentum 0.9.

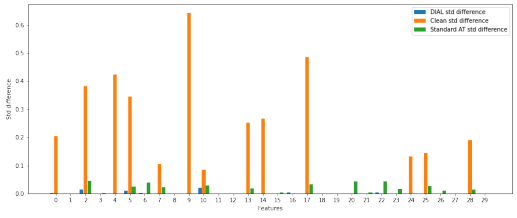
H Extended visualizations

In Figure 7, we provide additional visualizations of the different adversarial training methods presented above. We visualize the models outputs (logits) using t-SNE with two components on the natural test data and the corresponding adversarial test data generated using PGD^{20} white-box attack with step size 0.003 and $\epsilon = 0.031$ on CIFAR-10.

In Figure 6 we visualize statistical differences between natural and adversarial examples in the feature representation layer. Specifically, we show the differences in mean and std on thirty random feature values from the feature representation layer as we pass through a network the natural test examples and their corresponding adversarial examples. We present the results on same network architecture (WRN-34-10), trained using three different training procedures: naturally trained network, network trained using standard adversarial training (AT) Madry et al. (2017), and DIAL on the CIFAR-10 dataset. When the statistical characteristics of each feature differ from each other, it implies that the features layer is less domain invariant. That is, smaller differences in mean/std yields a better invariance to adversarial examples. One can observe that for DIAL, there is almost no differences between the mean/std of natural examples and their corresponding adversarial examples. Moreover, for the vast majority of the features, DIAL present smaller differences compared to the naturally trained model and the model trained with standard adversarial training. Best viewed in colors.

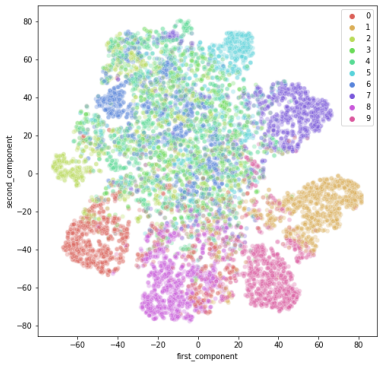


(a) Mean difference comparison

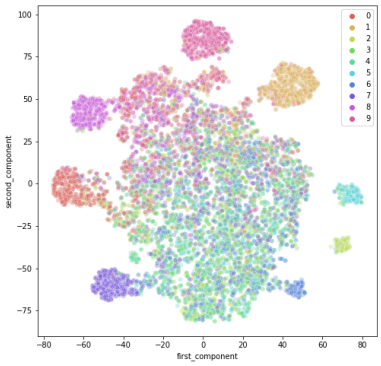


(b) Standard deviation difference comparison

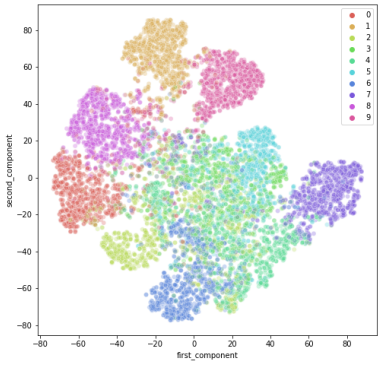
Figure 6: Mean and std differences comparison between DIAL, naturally trained model and model trained using standard adversarial training on thirty random features from the features layer on the CIFAR-10 dataset with WRN-34-10 architecture. Each bar represents the absolute difference between the means/std of the natural examples and the mean/std of their corresponding adversarial examples on this same feature.



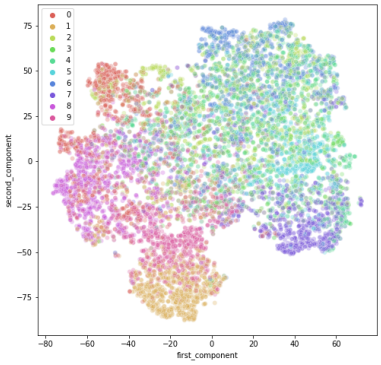
(a) **MART** embedded model output (logits) on natural test data



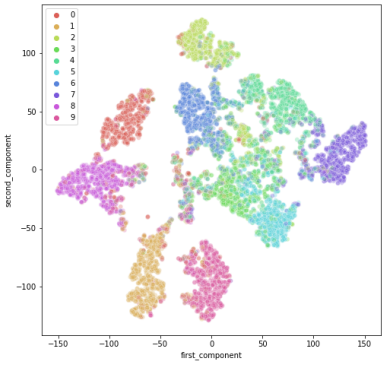
(b) **MART** embedded model output (logits) on adversarial test data



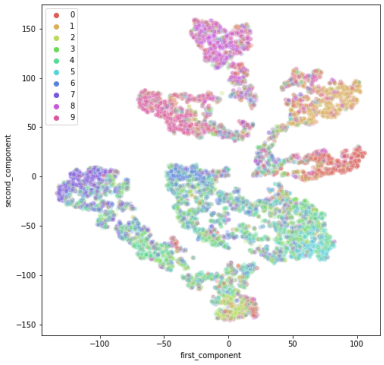
(c) **AT** embedded model output (logits) on natural test data



(d) **AT** embedded model output (logits) on adversarial test data



(e) **ATDA** embedded model output (logits) on natural test data

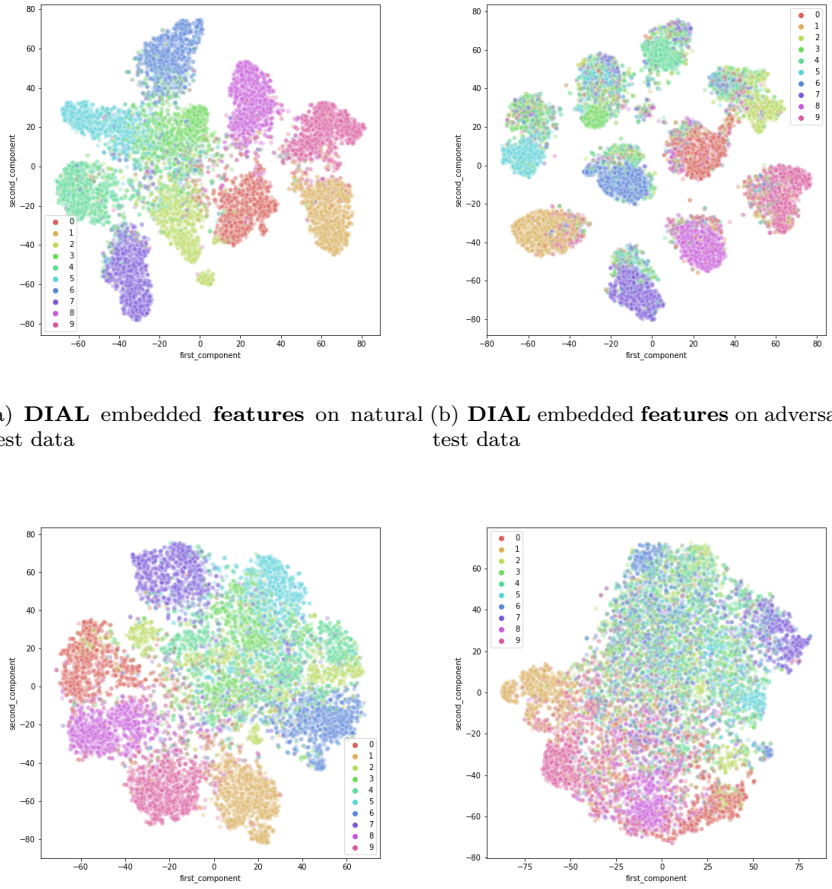


(f) **ATDA** embedded model output (logits) on adversarial test data

Figure 7: t-SNE embedding of model output, **logits**, in two-dimensional space for MART, AT, and ATDA under natural and adversarial test data from CIFAR-10.

I In-depth Analysis

Additionally, we present two additional results and visualizations that can show our performance difference is solid. First, in Figure 8 we present a 2d T-SNE plot on the **features** layer (unlike the logits layer , which is more related to the quantitative results), to further demonstrate that our method indeed learns a domain invariant feature representation better than the other methods, and compared it to TRADES. Second, we wish to demonstrate that our performance gain is not a due to specific aspects of the domain (e.g., improvement only on specific classes). To do so, we visualize in figures 9, 10 the classification improvement obtained by our method on each class in CIFAR-10. As can be seen, on the natural examples, our method improved the accuracy on all classes, and on the adversarial examples, our method improves robustness on 9 out of 10 classes. This further demonstrates the generalization of our approach.



(a) **DIAL** embedded **features** on natural test data

(b) **DIAL** embedded **features** on adversarial test data

(c) **TRADES** embedded **features** on natural test data

(d) **TRADES** embedded **features** on adversarial test data

Figure 8: Visualizing the two-dimensional T-SNE **feature space embedding** of DIAL and TRADES for (1) natural test data, and (2) adversarial test data from CIFAR-10.

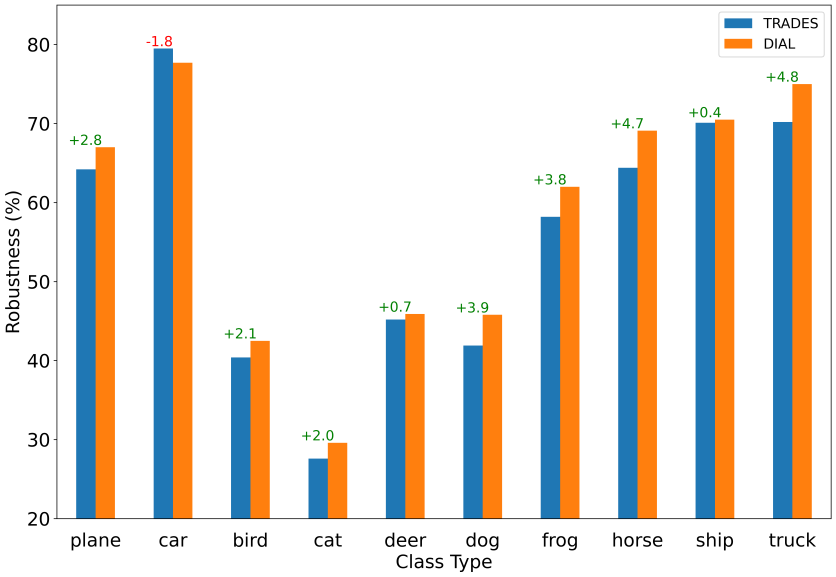


Figure 9: DIAL_{KL} and TRADES Robustness (%) for each class on CIFAR-10. Adversarial examples are generated using PGD-20. Our method manages to improve robustness over TRADES on 9 out of 10 classes. Green annotation presented the difference percentage improvement.

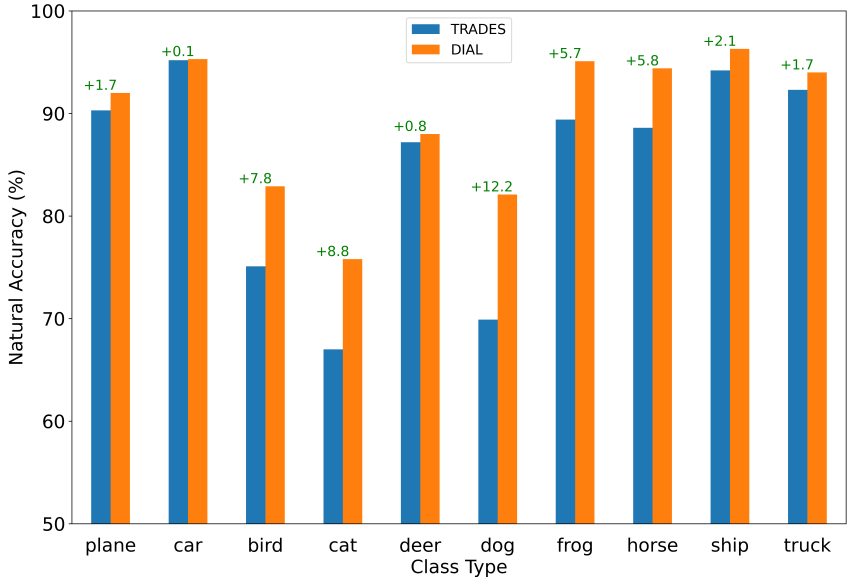


Figure 10: DIAL_{CE} and TRADES natural accuracy (%) for each class on CIFAR-10. Our method manages to improve natural accuracy on all 10 classes. Green annotation presented the difference percentage improvement.