Domain Invariant Adversarial Learning

Matan Levi
Department of Computer Science
Ben-Gurion University
Beer Sheva, Israel
matanle@post.bgu.ac.il

Idan Attias
Department of Computer Science
Ben-Gurion University
Beer Sheva, Israel
idanatti@post.bgu.ac.il

Aryeh Kontorovich
Department of Computer Science
Ben-Gurion University
Beer Sheva, Israel
karyeh@bgu.ac.il

Abstract

The discovery of adversarial examples revealed one of the most basic vulnerabilities of deep neural networks. Among the variety of techniques introduced to tackle this inherent weakness, adversarial training was shown to be the most common and efficient strategy to achieve robustness. It is usually done by balancing the robust and natural losses. In this work, we aim to achieve better trade-off between robust and natural performances by enforcing a domain-invariant feature representation. We present a new adversarial training method, called *Domain Invariant Adversarial Learning* (DIAL) that learns a feature representation which 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 a 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 better representation. We demonstrate our advantage by improving both robustness and natural accuracy compared to current state-of-the-art adversarial training methods.

1 Introduction

Deep learning models have achieved impressive success on a large range of challenging tasks. However, their performance was shown to be brittle to *adversarial examples*: a small perturbations in the input that flips the decision [8, 9, 22, 28, 32, 41, 43, 17, 42]. Designing reliable robust models has gained significant attention in the arms race against adversarial examples. Adversarial training [41, 22, 30, 57] has been suggested as one of the most effective approaches to defend against such examples, and can be described as solving the following min-max optimization problem:

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

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 [7, 18, 52, 1, 21].

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) [27, 30]. A PGD adversary starts with randomly initialized perturbation and iteratively adjust the perturbation while projecting it back into the ϵ-ball:

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

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 [3], PGD-based adversarial training was one of the few defenses that were not broken under strong attacks.

That said, the gap between robust accuracy on adversarial examples and natural accuracy on natural examples is still high for many tasks (e.g., CIFAR-10 [26], ImageNet [14], etc.). Generally speaking, Tsipras et al. [45] 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. [30] in various ways such as improved adversarial loss functions and regularization techniques [24, 48, 57], semi-supervised approaches [10, 46, 56], adversarial perturbations on model weights [51]. See related work for more details.

Our contribution. In this work, we make another step towards closing the gap between robustness and natural accuracy. In contrast to the aforementioned works, our method enhance adversarial training by enforcing a feature representation which is domain invariant between the natural and adversarial domains. We incorporate the idea of Domain-Adversarial Neural Networks (DANN) [19, 20] directly into the adversarial training process. DANN is a representation learning approach for domain adaptation. This approach aims to ensure that predictions are made based on invariant feature representation that cannot discriminate between source and target domains. Intuitively, the task of adversarial training and the task of domain invariant representation have a similar goal: given a source (natural) domain X and a the target (adversarial) domain X', we hope to achieve $g(X) \approx g(X')$, where g a feature representation function (i.e., neural network). Achieving such dual representation intuitively yields a more general feature representation.

Throughout extensive experiments on benchmark datasets, we show that by enforcing domain invariant representation learning using DANN simultaneously with the adversarial training process, we gain a significant and consistent improvement in both robustness and natural accuracy compared to other state-of-the-art adversarial training methods on benchmark datasets under various attack settings.

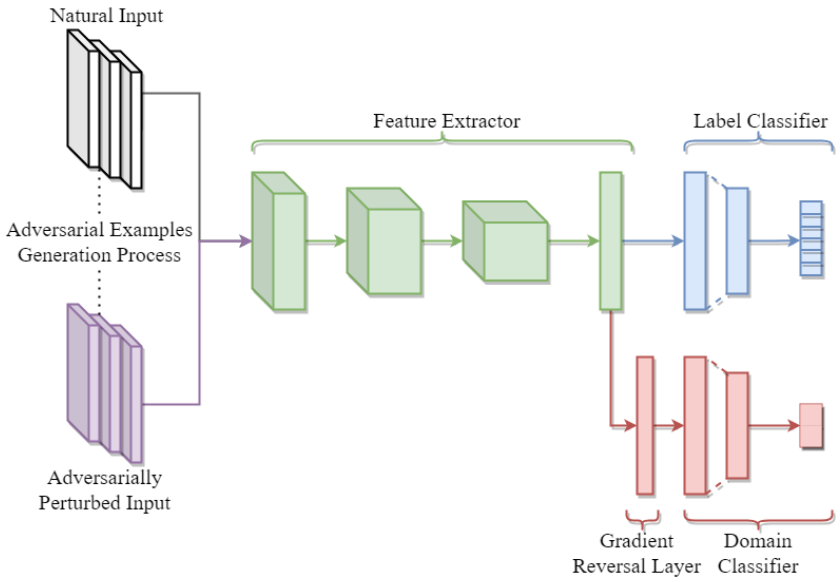


Figure 1: Illustration of the proposed architecture to enforce domain invariant representation. The feature extractor and label classifier form the a regular DNN architecture that can be used for the main natural task. The domain classifier is incorporated besides the label classifier. The reversal gradient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

2 Related Work

2.1 Defense Methods

A variety of empirical [15, 24, 27, 30, 43, 48, 57, 58] and theoretically principled [11, 34, 38, 35, 50, 49, 23] defense approaches for training robust classifiers were proposed since the discovery of adversarial examples. We elaborate on common and best performing adversarial training approaches, and highlight the differences compared to our method.

Madry et al. [30] proposed to minimize the cross entropy loss on adversarial examples that are found using the inner maximization process, i.e., Projected Gradient Descent (PGD). TRADES [57] suggested to decompose the prediction error for adversarial examples as the sum of the natural error and boundary error, and provided a differentiable upper bounds on both terms. From this decomposition, they suggested to use Kullback-Leibler (KL) divergence as a regularization term that will push the decision boundary away from the data. MART [48] suggested that miss-classified examples have a significant impact on final robustness, and differentiate between correctly classified and miss-classified examples during training.

Several works showed that unlabeled data can partially bridge this gap between robustness and natural accuracy and alleviate the need of immense amount of labeled data. Carmon et al. [10] proved that unlabeled data can bridge the sample complexity gap between standard and robust classification. In a similar work, [46] showed that in a simple statistical setting, the sample complexity for learning an adversarially robust model from unlabeled data matches the supervised case up to constant factors. Zhai et al. [56] used a problem presented in [37] to prove that it is possible to achieve better robust generalization if a sufficiently large amount of unlabeled data is provided. In this work, we focus on the settings where no additional data is used.

Another area of research tries to reveal the connection between the loss weight landscape, which is the loss change with respect to the weigh, and adversarial training [33, 54, 51]. Specifically, Wu et al. [51] identified correlation between the flatness of weight loss landscape and robust generalization gap. They proposed Adversarial Weight Perturbation (AWP) mechanism that is integrated into existing adversarial training methods. More recently, this approach was formalized from a theoretical standpoint [44]. However, this method forms a double-perturbation mechanism that perturbs both inputs and weights. This approach can cause a significant increase in calculation overhead. Nevertheless, we will show that our method still improves state-of-the-art results when incorporated with AWP.

A related approach to ours is [39] by Song et al. that suggested to add several constrains to the loss function in order to enforce domain adaptation: correlation alignment and maximum mean discrepancy [6, 40]. 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 [39] 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 [19, 20]. Moreover, [39] focused mainly of FGSM. We empirically demonstrate the superiority of our method in Section 4.

2.2 Robust generalization

Several works investigated the sample complexity requires the ensure adversarial generalization compared to the non-adversarial counterpart. Schmidt et al. [37] has shown that there exists a distribution (consists of 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 and VC-dimension [53, 4, 25, 5, 13, 31, 44] and additional settings [16, 10].

3 Domain Invariant Adversarial Learning

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 the internal representation cannot discriminate

between the natural and adversarial domains. By doing so we enforce additional regularization on the feature representation which results in a better robust representation.

Figure 1 illustrates the high level model architecture. $G_y(G_f(\cdot;\theta_f);\theta_y)$ is essentially the standard model (e.g., wide residual network [55]), while in addition, we have a domain classification layer to enforce a domain invariant on the feature representation.

3.1 Domain Invariant Adversarial Learning Loss

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 $G_d(\cdot;\theta_d)$ the domain classifier with parameters θ_d. Note the $G_y(G_f(\cdot;\theta_f);\theta_y)$ is exactly the standard architecture used for classification (e.g., wide residual network). First, we define the natural and adversarial losses:

$$\mathcal{L}^y_{\text{nat}} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_y(G_y(G_f(x_i;\theta_f);\theta_y), y_i),$$
$$\mathcal{L}^y_{\text{adv}} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_y(G_y(G_f(x_i^{'};\theta_f);\theta_y), y_i).$$

Next, we define the natural and adversarial domain losses:

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

We can now define domain invariant robust loss:

$$\mathcal{L}_{\text{DIAL}} = \mathcal{L}^y_{\text{nat}} + \lambda\mathcal{L}^y_{\text{adv}} + r(\mathcal{L}^d_{\text{nat}} + \mathcal{L}^d_{\text{adv}}).$$

Where $\{x_i\}_{i=1}^n$ are examples from the natural domain, and $\{x_i^{'}\}_{i=1}^n$ are the corresponding generated adversarial examples.

We choose two main variants to represent the adversarial loss. One is the standard cross-entropy loss which we refer to as DIAL_{CE}. The second is the Kullback–Leibler divergence which we refer to as DIAL_{KL}.

We note that the goal is to minimize the loss on the natural and adversarial classification while maximizing the loss for the domains. This way we can achieve feature representation which is domain invariant. The *reversal_ratio* hyper-parameters marked as r is inserted into the network layers as a gradient reversal layer [19, 20] that leaves the input unchanged during forward propagation and reverses the gradient by multiplying it by a negative scalar during the back-propagation. The parameter is adjusted during the training period to enable the main task converge at the beginning of the training, and enforce a domain invariant representation as the training progress.

3.2 DIAL Algorithm

Algorithm 1 describes a pseudo-code of our proposed DIAL algorithm. 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.

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

Output: Robust network $G = (G_f, G_y, G_d)$ parameterized by $\hat{\theta} = (\theta_f, \theta_y, \theta_d)$ respectively

while *Stopping criterion is not met* **do**
  Fetch mini-batch $B_s = \{(x_j, y_j)\}_{j=1}^{m}$
  # Generate adversarial target domain batch
  for *j=1,...,m (in parallel)* **do**
    $x'_j \leftarrow PGD(x_j, y_j, \epsilon, \tau)$
    $B_t \leftarrow B_t + x'_j$
  end
  $\ell_s^y,\ \ell_t^y \leftarrow \mathcal{L}_y(G_y(G_f(B_s))),\ \mathcal{L}_y(G_y(G_f(B_t)))$
  $\ell_s^d,\ \ell_t^d \leftarrow \mathcal{L}_d(G_d(G_f(B_s))),\ \mathcal{L}_d(G_d(G_f(B_t)))$
  $\ell \leftarrow \ell_s^y + \lambda \ell_t^y + r(\ell_s^d + \ell_t^d)$
  $\hat{\theta} \leftarrow \hat{\theta} - \alpha \nabla_{\hat{\theta}}(\ell)$
end

4 Experiments

In this section we evaluate our Domain Invariant Adversarial Learning (DIAL) and show that it achieves better robustness over state-of-the-art adversarial training methods while also improving the natural accuracy on both MNIST [29] and CIFAR-10 [26] benchmark datasets. We test our method in both white-box and black-box settings. We also suggest a new measurement to test the balance between robust and natural accuracy, which we named F_1-robust score.

4.1 Defense Model and Settings

To be consistent with other method, we follow the same experiment setups as in [30, 48, 57]. For each method, we use the best suggested hyper-parameters noted in the paper.

MNIST setup. We use the same CNN architecture as used in [57] 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, the weight the adversarial loss by $\lambda = 6$ and the domains loss by 0.1.

CIFAR-10 setup. We use the wide residual network (WRN-34-10) [55] architecture used in the experiments of [30, 48, 57]. 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 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 after 75 iterations, and by another factor of 10 after 90 iterations. 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 the models in the experiment are trained using SGD with momentum 0.9. For our $\mathrm{DIAL}_{\mathrm{KL}}$ method, the weight the adversarial loss by $\lambda = 8$ and the domains loss by 6. We also introduce a version of our method that incorporates the AWP

double-perturbation mechanism, named DIAL-AWP, trained using the same learning rate schedule used in [51], where the initial 0.1 learning rate decays by a factor of 10 after 100 and 150 iterations.

4.2 Attack models

To show that our results are not caused because of what is referred to as *obfuscated gradients* [3], we evaluate our method with same setup as in our defense model, under strong test attacks (e.g., PGD^{1000}) with white-box and black-box settings. To make sure that the reported improvements are not caused by *adversarial overfitting* [36], we report best robust results for all methods.

4.2.1 White-box Attacks

We summarize the natural and robust accuracy under white-box attacks on CIFAR-10 and MNIST in Tables 1 and 2 respectively. To make the comparison as fair as possible, we followed the same settings as the other state-of-the-art methods and report the best results.

Attacking MNIST. We evaluate all defense models using PGD^{40}, PGD^{100}, PGD^{1000} and CW_∞ (ℓ_∞ version of Carlini and Wagner [9] ℓ_∞ attack optimized by PGD-100) with step size 0.01. We constrain all attacks by the same perturbation $\epsilon = 0.3$. As reported in Table 2, Our method achieves better robustness over the other state-of-the-art methods under the different attack types, while preserving the same level of natural accuracy. We should note that in general, the improvement delta 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.

Attacking CIFAR-10. We evaluate all defense models using PGD^{20}, PGD^{100}, PGD^{1000} and CW_∞ with step size 0.003. We constrain all attacks by the same perturbation $\epsilon = 0.031$. As reported in Table 1, Our method achieves better robustness over the other state-of-the-art methods with respect to the different attacks. Specifically, we see that our method improves robustness over [30] by more than 2%, and around 2% over TRADES [57] using the common PGD^{20} attack while keeping higher natural accuracy. We also observe better natural accuracy of 1.65% over MART [48] while also achieving better robustness over all attacks. Moreover, our results show significant improvement over Song et al. [39] in both natural and robust accuracy. When incorporating the double-perturbation mechanism of AWP, our method improves the TRADES-AWP variant by a margin of almost 2% and reaches state-of-the-art results for robust models with no additional data. Additional results are available in Appendix A.

4.2.2 Black-box Attacks

To generate black-box attacks, we need to train a surrogate model. The surrogate model will be used to calculate the gradients for the adversarial perturbations. We use the same network architectures that were used to train the models in the previous section. In the experiments, two types of surrogate models were used (1) surrogate model that was trained independently using the original training datasets (without adversarial training), and (2) surrogate model that was trained using one of the adversarial training methods.

Attacking MNIST. For consistency, we use same perturbation and step sizes. For MNIST, we use $\epsilon = 0.3$ and step size of 0.01. The natural accuracy of our surrogate (source) model is 99.51%. Attacks results are reported in Table 2. It is worth noting that the improvement delta is not conclusive on MNIST as it is on CIFAR-10, which is a more complex task.

Attacking CIFAR-10. For consistency, we use same perturbation and step sizes. For CIFAR-10, we use $\epsilon = 0.031$ with step size 0.003. The natural accuracy of our surrogate (source) model is 95.61%. Attacks results are reported in Table 1. To keep the comparison fair across all methods, the target models are the same robust model from previous white-box attack section. We can observe that our method improves robustness by a significant and consistent margin over the other methods and attacks. In addition to the naturally trained source model results, we present additional results in Table 3 where the source model is now trained using one of the adversarial training methods. Then, we compare our model to each one of them both as the source model and target model.

Table 1: Robustness against white-box and black-box attacks on CIFAR-10. 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}	CW^{∞}	PGD^{20}	PGD^{100}	CW^{∞}
TRADES [57]	84.92	56.6	55.56	54.2	84.08	83.89	83.91
MART [48]	83.62	58.12	56.48	53.09	82.82	82.52	82.8
Madry et al. [30]	85.1	56.28	54.46	53.99	84.22	84.14	83.92
Song et al. [39]	76.91	43.27	41.13	41.01	75.59	75.37	75.35
$DIAL_{KL}$ (Ours)	85.25	**58.43**	**56.8**	**55**	84.3	84.18	84.05
$DIAL_{CE}$ (Ours)	**90**	52.12	48.88	50.15	**89.13**	**88.89**	**88.92**
DIAL-AWP (Ours)	**85.91**	**61.1**	**59.86**	**57.67**	**85.13**	**84.93**	**85.03**
TRADES-AWP [51]	85.36	59.27	59.12	57.07	84.58	84.58	84.59

Table 2: Robustness against white-box and black-box attacks 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^{∞}
TRADES [57]	99.48	96.07	95.52	95.69	98.12	97.86	98.21
MART [48]	99.38	96.99	96.11	95.98	98.16	97.96	98.28
Madry et al. [30]	99.41	96.01	95.49	95.78	98.05	97.73	98.2
Song et al. [39]	98.72	96.82	96.26	96.31	97.74	97.28	97.76
$DIAL_{KL}$ (Ours)	99.46	97.05	96.06	96.17	98.14	97.83	98.14
$DIAL_{CE}$ (Ours)	**99.49**	**97.38**	**96.45**	**96.41**	**98.37**	**98.12**	**98.35**

Table 3: Black-box PGD^{20} attack using the adversarially trained surrogate models on CIFAR-10

Surrogate (source) model	Target model	robustness %
TRADES [57]	$DIAL_{CE}$	**68.72**
$DIAL_{CE}$	TRADES	67.83
MART [48]	$DIAL_{CE}$	**71.33**
$DIAL_{CE}$	MART	67.27
Madry et al. [30]	$DIAL_{CE}$	**67.68**
$DIAL_{CE}$	Madry et al.	66.68
Song et al. [39]	$DIAL_{CE}$	**68.7**
$DIAL_{CE}$	Song et al.	56.09

4.2.3 Ensemble Attack

In addition to the white-box and black-box settings, we evaluate our method on the Auto-Attack [12] using ℓ_{∞} threat model with perturbation $\epsilon = 0.031$. Auto-Attack is an ensemble of parameter-free attacks. It consists of three white-box attacks: APGD-CE which is a step size-free version of PGD on the cross-entropy [12]. APGD-DLR which is a step size-free version of PGD on the DLR loss [12] and FAB which minimizes the norm of the adversarial perturbations, and one black-box attack: square attack which is a query-efficient black-box attack [2]. Results are presented in Table 4. Based on the auto-attack leader-board [1], our method achieves the 1st place among models without additional data using the WRN-34-10 architecture.

[1] https://github.com/fra31/auto-attack

Table 4: Auto-Attack (AA) on CIFAR-10 with perturbation size $\epsilon = 0.031$ with ℓ_∞ threat model

Defense Model	AA
TRADES [57]	53.08
MART [48]	51.1
Madry et al. [30]	51.52
Song et al. [39]	40.18
DIAL_{CE} (Ours)	47.33
DIAL_{KL} (Ours)	**53.75**
DIAL-AWP (Ours)	**56.78**
TRADES-AWP [51]	56.17

4.3 Balanced Measurement for Robust and 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 he harmonic mean between the precision and recall. However, instead of using precision and recall, we measure the F_1-score between robustness and natural accuracy. We named it **$\mathbf{F_1}$-robust** score.

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

where true_robust are the adversarial examples that were correctly classified, false_robust are the adversarial examples that where miss-classified, and false_natural are the natural examples that were miss-classified. We tested the proposed F_1-robust score using PGD^{20} on CIFAR-10 dataset in white-box and black-box settings. Results are presented in Table 5 and show that our method achieves the best F_1-robust score in both settings, which supports our findings from previous sections.

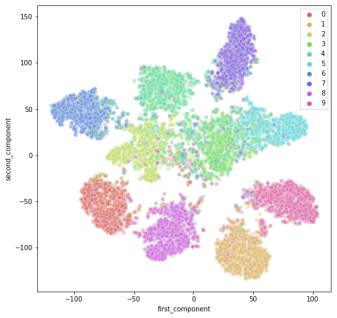
Table 5: F_1-robust measurement using PGD^{20} attack in white-box and black-box settings on CIFAR-10

Defense Model	White-box	Black-box
TRADES [57]	0.65937	0.84435
MART [48]	0.66613	0.83153
Madry et al. [30]	0.65755	0.84574
Song et al. [39]	0.51823	0.76092
DIAL_{CE} (Ours)	0.64265	**0.89519**
DIAL_{KL} (Ours)	**0.67479**	0.84702
DIAL-AWP (Ours)	**0.69753**	**0.85406**
TRADES-AWP [51]	0.68162	0.84917

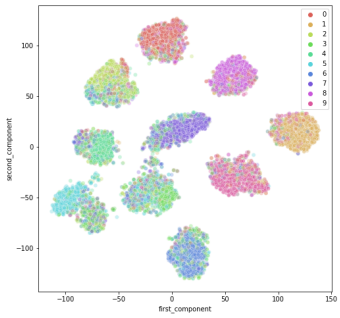
4.4 Visualizing DIAL

To further illustrate our method, we visualize the model outputs using the different methods under natural test data and adversarial test data generated using PGD^{20} white-box attack with step size 0.003 and $\epsilon = 0.031$ on CIFAR-10. Figure 2 shows the embedding received after applying t-SNE [47] with two components on the model output for our method and for TRADES. We can observe that our method 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 B.

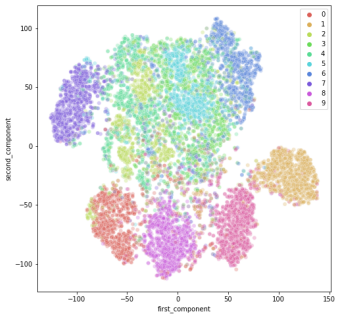
Figure 2: t-SNE embedding of model output in two-dimensional space for DIAL and TRADES under natural and adversarial test data from CIFAR-10



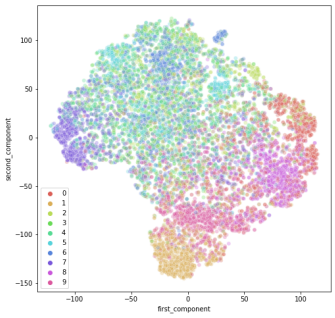
(a) **DIAL** embedded model output on natural test data



(b) **DIAL** embedded model output on adversarial test data



(c) **TRADES** embedded model output on natural test data



(d) **TRADES** embedded model output on adversarial test data

5 Conclusion

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 (DANN) into the adversarial training process, where the natural examples act as our source domain, and the corresponding adversarial examples act as our target domain. The proposed method is generic and can be combined with any network architecture in a wide range of tasks. By extensive empirical analysis, we demonstrate the significant and consistent improvement obtained by DIAL in both robustness and natural accuracy compared to state-of-the-art methods on benchmark datasets under various attack settings.

References

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

[2] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020.

[3] 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*, pages 274–283. PMLR, 2018.

[4] Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for robust learning. In Aurélien Garivier and Satyen Kale, editors, *Algorithmic Learning Theory, ALT 2019, 22-24 March 2019, Chicago, Illinois, USA*, volume 98 of *Proceedings of Machine Learning Research*, pages 162–183. PMLR, 2019.

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

[6] 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.

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

[8] 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*, pages 3–14, 2017.

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

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

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

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

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

[14] 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*, pages 248–255. Ieee, 2009.

[15] 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.

[16] 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*, pages 10359–10368, 2018.

[17] 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*, pages 9185–9193, 2018.

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

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

[20] 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.

[21] 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.

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

[23] 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.

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

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

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

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

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

[29] 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.

[30] 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.

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

[32] 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*, pages 2574–2582, 2016.

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

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

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

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

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

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

[39] 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.

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

[41] 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.

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

[43] 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.

[44] 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.

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

[46] 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.

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

[48] 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*, 2019.

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

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

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

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

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

[54] 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.

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

[56] 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.

[57] 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*, pages 7472–7482. PMLR, 2019.

[58] 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*, pages 11278–11287. PMLR, 2020.

A Additional results

In Table 6 we present additional results using the PGD^{1000} threat model. We use step size 0.003 and constrain the attacks by the same perturbation $\epsilon = 0.031$. Table 7 presents a comparison of our method combined with AWP to other the variants of AWP that were presented in [51]. In addition, in Table 8 we add the F_1-robust scores for different variants of AWP.

Table 6: 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 [57]	95.22	97.81	56.43	83.8
MART [48]	95.74	97.89	56.55	82.47
Madry et al. [30]	95.36	97.78	54.4	83.96
Song et al. [39]	96.2	97.34	41.02	75.11
DIAL_{CE} (Ours)	**96.33**	**98.05**	48.78	**88.78**
DIAL_{KL} (Ours)	95.99	97.89	**56.73**	84

Table 7: 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**	**61.1**	**59.86**	**57.67**	**56.78**
TRADES-AWP [51]	85.36	59.27	59.12	57.07	56.17
MART-AWP [51]	84.43	60.68	59.32	56.37	54.23
Madry-AWP [51]	85.57	58.14	57.94	55.96	54.04

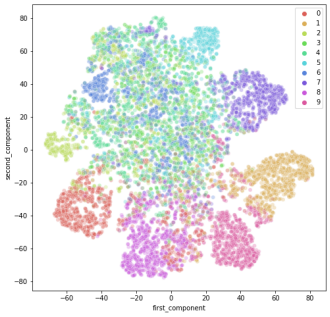
Table 8: F_1-robust measurement on AWP variants based on white-box attack

Defense Model	F_1-robust
DIAL-AWP (Ours)	**0.69753**
TRADES-AWP [51]	0.68162
MART-AWP [51]	0.68857
Madry-AWP [51]	0.67381

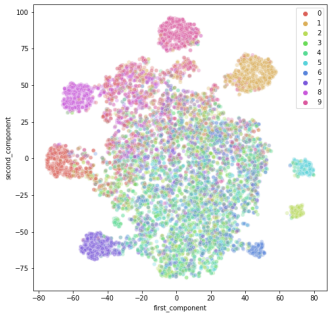
B Additional visualizations

In Figure 3, we provide additional visualizations of the different adversarial training methods presented above. We visualize the models outputs using t-SNE with two components on the natural test data and adversarial test data generated by the PGD^{20} white-box attack with step size 0.003 and $\epsilon = 0.031$ on CIFAR-10.

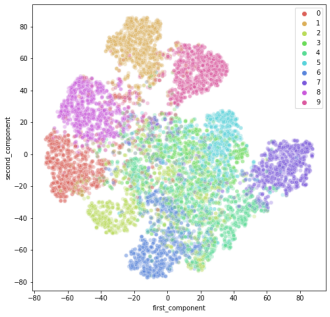
Figure 3: t-SNE embedding of model output in two-dimensional space for MART, Madry et al. and Song et al. under natural and adversarial test data from CIFAR-10



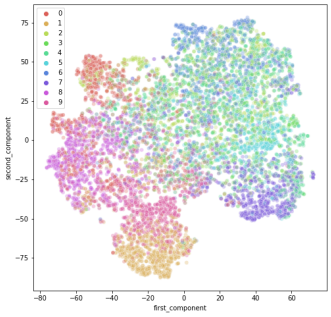
(a) **MART** embedded model output on natural test data



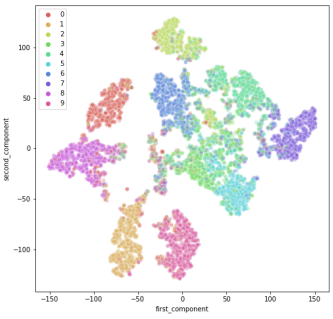
(b) **MART** embedded model output on adversarial test data



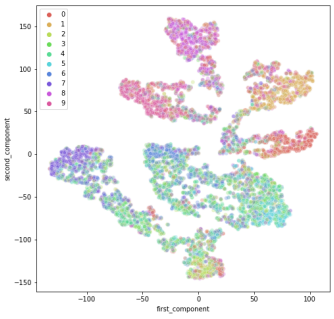
(c) **Madry et al.** embedded model output on natural test data



(d) **Madry et al.** embedded model output on adversarial test data



(e) **Song et al.** embedded model output on natural test data



(f) **Song et al.** embedded model output on adversarial test data