Adversarial Attack Generation Empowered by Min-Max Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

The worst-case training principle that minimizes the maximal adversarial loss, also known as adversarial training (AT), has shown to be a state-of-the-art approach for enhancing adversarial robustness. Nevertheless, min-max optimization beyond the purpose of AT has not been rigorously explored in the adversarial context. In this paper, we show how a general notion of min-max optimization over multiple domains can be leveraged to advance the design of different types of adversarial attacks. In particular, given a set of risk sources, minimizing the worst-case attack loss can be reformulated as a min-max problem by introducing domain weights that are maximized over the probability simplex of the domain set. We showcase this unified framework in three attack generation problems – attacking model ensembles, devising universal perturbation under multiple inputs, and crafting attacks resilient to data transformations. Extensive experiments demonstrate that our approach leads to substantial attack improvement over the existing heuristic strategies as well as robustness improvement over state-of-the-art defense methods trained to be robust against multiple perturbation types. Furthermore, we find that the self-adjusted domain weights learned from our min-max framework can provide a holistic tool to explain the difficulty level of attack across domains.

1 Introduction

Training a machine learning model that is capable of assuring its worst-case performance against possible adversaries given a specified threat model is a fundamental and challenging problem, especially for deep neural networks (DNNs) [54, 19, 11]. A common practice to train an adversarially robust model is based on a specific form of min-max training, known as *adversarial training* (AT) [19, 35], where the minimization step learns model weights under the adversarial loss constructed at the maximization step in an alternative training fashion. In practice, AT has achieved the state-of-the-art defense performance against ℓ_p-norm-ball input perturbations [3].

Although the min-max principle is widely used in AT and its variants [35, 50, 64, 55], few work has studied its power in attack generation. Thus, we ask: *Beyond AT, can other types of min-max formulation and optimization techniques advance the research in adversarial attack generation?* In this paper, we give an affirmative answer corroborated by the substantial performance gain and the ability of self-learned risk interpretation using our proposed min-max framework on several tasks for adversarial attack.

We demonstrate the utility of a general formulation for minimizing the maximal loss induced from a set of risk sources (domains). Our considered min-max formulation is fundamentally different from AT, as our maximization step is taken over the probability simplex of the set of domains. Moreover, we show that many problem setups in adversarial attacks can in fact be reformulated under this general min-max framework, including attacking model ensembles [56, 30], devising universal perturbation

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

to input samples [39] and data transformations [6, 8]. However, current methods for solving these tasks often rely on simple heuristics (e.g., uniform averaging), resulting in significant performance drops when comparing to our proposed min-max optimization framework.

Contributions ① With the aid of min-max optimization, we propose a unified alternating one-step projected gradient descent-ascent (APGDA) attack method, which can readily be specified to generate model ensemble attack, universal attack over multiple images, and robust attack over data transformations. ② In theory, we show that APGDA has an $O(1/T)$ convergence rate, where T is the number of iterations. In practice, we show that APGDA obtains 17.48%, 35.21% and 9.39% improvement on average compared with conventional min-only PGD attack methods on CIFAR-10. ③ More importantly, we demonstrate that by tracking the learnable weighting factors associated with multiple domains, our method can provide tools for self-adjusted importance assessment on the mixed learning tasks. ④ Finally, we adapt the idea of the domain weights into a defense setting [55], where multiple ℓ_p-norm perturbations are generated, and achieve superior performance as well as intepretability.

Related work Recent studies have identified that DNNs are highly vulnerable to adversarial manipulations in various applications [54, 10, 24, 29, 23, 12, 65, 17, 13, 28], thus leading to an arms race between adversarial attacks [11, 3, 20, 43, 40, 60, 1, 16] and defenses [35, 50, 64, 55, 37, 59, 62, 58, 47, 14]. One intriguing property of adversarial examples is the transferability across multiple domains [31, 57, 42, 52], which indicates a more challenging yet promising research direction – devising universal adversarial perturbations over model ensembles [56, 30], input samples [39, 38, 48] and data transformations [3, 6, 8].

Besides, many recent works started to produce physical realizable perturbations that expose real world threats. The most popular approach [4, 61, 18], as known as Expectation Over Transformation (EOT), is to train the attack under different data transformation (e.g., different view angles and distances). However, current approaches suffer from a significant performance loss for resting on the uniform averaging strategy or heuristic weighting schemes [30, 48]. We will compare these works with our min-max method in Sec. 4. As a natural extension following min-max attack, we study the generalized AT under multiple perturbations [55, 2, 25, 15]. Finally, our min-max framework is adapted and inspired by previous literature on robust learning over multiple domains [45, 46, 33, 32].

2 Min-Max Across Domains

Consider K loss functions $\{F_i(\mathbf{v})\}$ (each of which is defined on a learning domain), the problem of robust learning over K domains can be formulated as [45, 46, 33]

$$\underset{\mathbf{v}\in\mathcal{V}}{\text{minimize}}\ \underset{\mathbf{w}\in\mathcal{P}}{\text{maximize}} \quad \textstyle\sum_{i=1}^{K} w_i F_i(\mathbf{v}), \tag{1}$$

where $\mathbf{v}$ and $\mathbf{w}$ are optimization variables, $\mathcal{V}$ is a constraint set, and $\mathcal{P}$ denotes the probability simplex $\mathcal{P} = \{\mathbf{w} \,|\, \mathbf{1}^T\mathbf{w} = 1, w_i \in [0,1], \forall i\}$. Since the inner maximization problem in (1) is a linear function of $\mathbf{w}$ over the probabilistic simplex, problem (1) is thus equivalent to

$$\underset{\mathbf{v}\in\mathcal{V}}{\text{minimize}}\ \underset{i\in[K]}{\text{maximize}} \quad F_i(\mathbf{v}), \tag{2}$$

where $[K]$ denotes the integer set $\{1, 2, \ldots, K\}$.

Benefit and Challenge from (1). Compared to multi-task learning in a finite-sum formulation which minimizes K losses on *average*, problem (1) provides consistently robust *worst-case* performance across all domains. This can be explained from the epigraph form of (2),

$$\underset{\mathbf{v}\in\mathcal{V},t}{\text{minimize}}\ t, \quad \text{subject to}\ F_i(\mathbf{v}) \le t, i \in [K], \tag{3}$$

where t is an epigraph variable [7] that provides the t-level robustness at each domain.

In computation, the inner maximization problem of (1) always returns the one-hot value of $\mathbf{w}$, namely, $\mathbf{w} = \mathbf{e}_i$, where $\mathbf{e}_i$ is the ith standard basis vector, and $i = \arg\max_i\{F_i(\mathbf{v})\}$. However, this one-hot coding reduces the generalizability to other domains and induces instability of the learning procedure in practice. Such an issue is often mitigated by introducing a *strongly concave regularizer* in the inner maximization step to strike a balance between the average and the worst-case performance [33, 45].

Regularized Formulation. Following [45], we penalize the distance between the *worst-case* loss and the *average* loss over K domains. This yields

$$\underset{\mathbf{v}\in\mathcal{V}}{\text{minimize}}\ \underset{\mathbf{w}\in\mathcal{P}}{\text{maximize}} \quad \sum_{i=1}^{K} w_i F_i(\mathbf{v}) - \frac{\gamma}{2}\|\mathbf{w} - \mathbf{1}/K\|_2^2, \tag{4}$$

where $\gamma > 0$ is a regularization parameter. As $\gamma \to 0$, problem (4) is equivalent to (1). By contrast, it becomes the finite-sum problem when $\gamma \to \infty$ since $\mathbf{w} \to \mathbf{1}/K$. *In this sense, the trainable* $\mathbf{w}$ *provides an essential indicator on the importance level of each domain.* The larger the weight is, the more important the domain is. We call $\mathbf{w}$ *domain weights* in this paper.

3 Min-Max Power in Attack Design

To the best of our knowledge, few work has studied the power of min-max in attack generation. In this section, we demonstrate how the unified min-max framework (4) fits into various attack settings. With the help of domain weights, our solution yields better empirical performance and explainability. Finally, we present the min-max algorithm with convergence analysis to craft robust perturbations against multiple domains.

3.1 A Unified Framework for Robust Adversarial Attacks

The general goal of adversarial attack is to craft an adversarial example $\mathbf{x}' = \mathbf{x}_0 + \boldsymbol{\delta} \in \mathbb{R}^d$ to mislead the prediction of machine learning (ML) or deep learning (DL) systems, where $\mathbf{x}_0$ denotes the natural example with the true label t_0, and $\boldsymbol{\delta}$ is known as *adversarial perturbation*, commonly subject to ℓ_p-norm ($p \in \{0, 1, 2, \infty\}$) constraint $\mathcal{X} := \{\boldsymbol{\delta} \,|\, \|\boldsymbol{\delta}\|_p \leq \epsilon,\ \mathbf{x}_0 + \boldsymbol{\delta} \in [0,1]^d\}$ for a given small number ϵ. Here the ℓ_p norm enforces the similarity between $\mathbf{x}'$ and $\mathbf{x}_0$, and the input space of ML/DL systems is normalized to $[0,1]^d$.

Ensemble Attack over Multiple Models. Consider K ML/DL models $\{\mathcal{M}_i\}_{i=1}^K$, the goal is to find robust adversarial examples that can fool all K models *simultaneously*. In this case, the notion of 'domain' in (4) is specified as 'model', and the objective function F_i in (4) signifies the attack loss $f(\boldsymbol{\delta}; \mathbf{x}_0, y_0, \mathcal{M}_i)$ given the natural input $(\mathbf{x}_0, y_0)$ and the model $\mathcal{M}_i$. Thus, problem (4) becomes

$$\underset{\boldsymbol{\delta}\in\mathcal{X}}{\text{minimize}}\ \underset{\mathbf{w}\in\mathcal{P}}{\text{maximize}} \quad \sum_{i=1}^{K} w_i f(\boldsymbol{\delta}; \mathbf{x}_0, y_0, \mathcal{M}_i) - \frac{\gamma}{2}\|\mathbf{w} - \mathbf{1}/K\|_2^2, \tag{5}$$

where $\mathbf{w}$ encodes the difficulty level of attacking each model.

Universal Perturbation over Multiple Examples. Consider K natural examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^K$ and a single model $\mathcal{M}$, our goal is to find the universal perturbation $\boldsymbol{\delta}$ so that all the corrupted K examples can fool $\mathcal{M}$. In this case, the notion of 'domain' in (4) is specified as 'example', and problem (4) becomes

$$\underset{\boldsymbol{\delta}\in\mathcal{X}}{\text{minimize}}\ \underset{\mathbf{w}\in\mathcal{P}}{\text{maximize}} \quad \sum_{i=1}^{K} w_i f(\boldsymbol{\delta}; \mathbf{x}_i, y_i, \mathcal{M}) - \frac{\gamma}{2}\|\mathbf{w} - \mathbf{1}/K\|_2^2, \tag{6}$$

where different from (5), $\mathbf{w}$ encodes the difficulty level of attacking each example.

Adversarial Attack over Data Transformations. Consider K categories of data transformation $\{p_i\}$, e.g., rotation, lightening, and translation, our goal is to find the adversarial attack that is robust to data transformations. Such an attack setting is commonly applied to generate physical adversarial examples [5, 17]. Here the notion of 'domain' in (4) is specified as 'data transformer', and problem (4) becomes

$$\underset{\boldsymbol{\delta}\in\mathcal{X}}{\text{minimize}}\ \underset{\mathbf{w}\in\mathcal{P}}{\text{maximize}} \quad \sum_{i=1}^{K} w_i \mathbb{E}_{t\sim p_i}[f(t(\mathbf{x}_0 + \boldsymbol{\delta}); y_0, \mathcal{M})] - \frac{\gamma}{2}\|\mathbf{w} - \mathbf{1}/K\|_2^2, \tag{7}$$

where $\mathbb{E}_{t\sim p_i}[f(t(\mathbf{x}_0+\boldsymbol{\delta}); y_0, \mathcal{M})]$ denotes the attack loss under the distribution of data transformation p_i, and $\mathbf{w}$ encodes the difficulty level of attacking each type of transformed example $\mathbf{x}_0$. We remark that if $\mathbf{w} = \mathbf{1}/K$, then problem (7) reduces to the existing expectation of transformation (EOT) setup used for physical attack generation [5].

Benefits of Min-Max Attack Generation with Learnable Domain Weights $\mathbf{w}$: We can interpret (5)-(7) as finding the *robust* adversarial attack against the *worst-case environment* that an adversary encounters, e.g., multiple victim models, data samples, and input transformations. The proposed min-max design of adversarial attacks leads to two main benefits. First, compared to the heuristic weighting strategy (e.g., clipping thresholds on the importance of individual attack losses [48]), our proposal is free of supervised manual adjustment on domain weights. Even by carefully tuning the heuristic weighting strategy, we find that our approach with self-adjusted $\mathbf{w}$ consistently outperforms the clipping strategy in [48] (see Table 2). Second, the learned domain weights can be used to assess the model robustness when facing different types of adversary. We refer readers to Figure 1c and Figure 4 for more details.

3.2 Min-Max Algorithm for Adversarial Attack Generation

We propose the **a**lternating **p**rojected **g**radient **d**escent-**a**scent (APGDA) method (Algorithm 1) to solve problem (4). For ease of presentation, we write problems (5), (6), (7) into the general form

$$\underset{\boldsymbol{\delta}\in\mathcal{X}}{\text{minimize}} \quad \underset{\mathbf{w}\in\mathcal{P}}{\text{maximize}} \quad \sum_{i=1}^{K} w_i F_i(\boldsymbol{\delta}) - \frac{\gamma}{2}\|\mathbf{w} - \mathbf{1}/K\|_2^2,$$

where F_i denotes the ith individual attack loss. We show that at each iteration, APGDA takes only one-step PGD for outer minimization and one-step projected gradient ascent for inner maximization.

Algorithm 1 APGDA to solve problem (4)

1: Input: given $\mathbf{w}^{(0)}$ and $\boldsymbol{\delta}^{(0)}$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: *outer min.*: fixing $\mathbf{w} = \mathbf{w}^{(t-1)}$, call PGD (8) to update $\boldsymbol{\delta}^{(t)}$
4: *inner max.*: fixing $\boldsymbol{\delta} = \boldsymbol{\delta}^{(t)}$, update $\mathbf{w}^{(t)}$ with projected gradient ascent (9)
5: **end for**

Outer Minimization Considering $\mathbf{w} = \mathbf{w}^{(t-1)}$ and $F(\boldsymbol{\delta}) := \sum_{i=1}^{K} w_i^{(t-1)} F_i(\boldsymbol{\delta})$ in (4), we perform one-step PGD to update $\boldsymbol{\delta}$ at iteration t,

$$\boldsymbol{\delta}^{(t)} = \text{proj}_{\mathcal{X}}\left(\boldsymbol{\delta}^{(t-1)} - \alpha \nabla_{\boldsymbol{\delta}} F(\boldsymbol{\delta}^{(t-1)})\right), \tag{8}$$

where $\text{proj}(\cdot)$ denotes the Euclidean projection operator, i.e., $\text{proj}_{\mathcal{X}}(\mathbf{a}) = \arg\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - \mathbf{a}\|_2^2$ at the point $\mathbf{a}$, $\alpha > 0$ is a given learning rate, and $\nabla_{\boldsymbol{\delta}}$ denotes the first-order gradient w.r.t. $\boldsymbol{\delta}$. If $p = \infty$, then the projection function becomes the clip function. In Proposition 1, we derive the solution of $\text{proj}_{\mathcal{X}}(\mathbf{a})$ under different ℓ_p norms for $p \in \{0, 1, 2\}$.

Proposition 1. *Given a point $\mathbf{a} \in \mathbb{R}^d$ and a constraint set $\mathcal{X} = \{\boldsymbol{\delta} | \|\boldsymbol{\delta}\|_p \le \epsilon, \check{\mathbf{c}} \le \boldsymbol{\delta} \le \hat{\mathbf{c}}\}$, the Euclidean projection $\boldsymbol{\delta}^* = \text{proj}_{\mathcal{X}}(\mathbf{a})$ has a closed-form solution when $p \in \{0, 1, 2\}$, where the specific form is given by Appendix A.*

Inner Maximization By fixing $\boldsymbol{\delta} = \boldsymbol{\delta}^{(t)}$ and letting $\psi(\mathbf{w}) := \sum_{i=1}^{K} w_i F_i(\boldsymbol{\delta}^{(t)}) - \frac{\gamma}{2}\|\mathbf{w} - \mathbf{1}/K\|_2^2$ in problem (4), we then perform one-step PGD (w.r.t. $-\psi$) to update $\mathbf{w}$,

$$\mathbf{w}^{(t)} = \text{proj}_{\mathcal{P}}\Big(\underbrace{\mathbf{w}^{(t-1)} + \beta \nabla_{\mathbf{w}} \psi(\mathbf{w}^{(t-1)})}_{\mathbf{b}}\Big) = (\mathbf{b} - \mu \mathbf{1})_+, \tag{9}$$

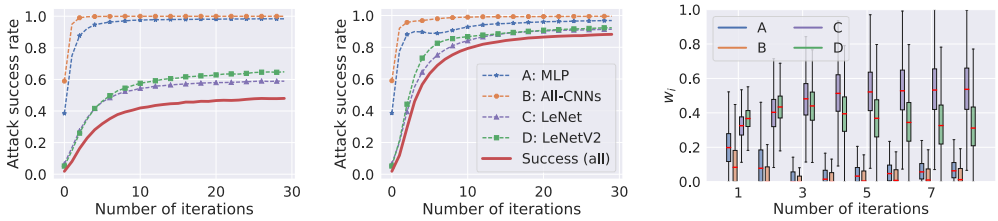
where $\beta > 0$ is a given learning rate, $\nabla_{\mathbf{w}}\psi(\mathbf{w}) = \boldsymbol{\phi}^{(t)} - \gamma(\mathbf{w} - \mathbf{1}/K)$, and $\boldsymbol{\phi}^{(t)} := [F_1(\boldsymbol{\delta}^{(t)}), \ldots, F_K(\boldsymbol{\delta}^{(t)})]^T$. In (9), the second equality holds due to the closed-form of projection operation onto the probabilistic simplex $\mathcal{P}$ [44], where $(x)_+ = \max\{0, x\}$, and μ is the root of the equation $\mathbf{1}^T(\mathbf{b} - \mu\mathbf{1})_+ = 1$. Since $\mathbf{1}^T(\mathbf{b} - \min_i\{b_i\}\mathbf{1} + \mathbf{1}/K)_+ \ge \mathbf{1}^T\mathbf{1}/K = 1$, and $\mathbf{1}^T(\mathbf{b} - \max_i\{b_i\}\mathbf{1} + \mathbf{1}/K)_+ \le \mathbf{1}^T\mathbf{1}/K = 1$, the root μ exists within the interval $[\min_i\{b_i\} - 1/K, \max_i\{b_i\} - 1/K]$ and can be found via the bisection method [7].

Convergence Analysis We remark that APGDA follows the gradient primal-dual optimization framework [32], and thus enjoys the same optimization guarantees.

Theorem 1. *Suppose that in problem (4) $F_i(\boldsymbol{\delta})$ has L-Lipschitz continuous gradients, and $\mathcal{X}$ is a convex compact set. Given learning rates $\alpha \le \frac{1}{L}$ and $\beta < \frac{1}{\gamma}$, then the sequence $\{\boldsymbol{\delta}^{(t)}, \mathbf{w}^{(t)}\}_{t=1}^{T}$ generated by Algorithm 1 converges to a first-order stationary point[1] in rate $\mathcal{O}\left(\frac{1}{T}\right)$.*

Proof: Note that the objective function of problem (4) is strongly concave w.r.t. $\mathbf{w}$ with parameter γ, and has γ-Lipschitz continuous gradients. Moreover, we have $\|\mathbf{w}\|_2 \le 1$ due to $\mathbf{w} \in \mathcal{P}$. Using these facts and Theorem 1 in [32] or [34] completes the proof. □

[1] The stationarity is measured by the ℓ_2 norm of gradient of the objective in (4) w.r.t. $(\boldsymbol{\delta}, \mathbf{w})$.



(a) average case (b) min max (c) weight $\{w_i\}$

Figure 1: Ensemble attack against four DNN models on MNIST. (a) & (b): Attack success rate of adversarial examples generated by average (ensemble PGD) or min-max (APGDA) attack method. (c): Boxplot of weight w in min-max adversarial loss. Here we adopt the same ℓ_∞-attack as Table 1.

Table 1: Comparison of average and min-max (APGDA) ensemble attack on MNIST.

Box constraint	Opt.	Acc_A	Acc_B	Acc_C	Acc_D	ASR_{all}	Lift (↑)
ℓ_0 ($\epsilon = 30$)	*avg.*	7.03	1.51	11.27	2.48	84.03	-
	min max	3.65	2.36	4.99	3.11	**91.97**	**9.45%**
ℓ_1 ($\epsilon = 20$)	*avg.*	20.79	0.15	21.48	6.70	69.31	-
	min max	6.12	2.53	8.43	5.11	**89.16**	**28.64%**
ℓ_2 ($\epsilon = 3.0$)	*avg.*	6.88	0.03	26.28	14.50	69.12	-
	min max	1.51	0.89	3.50	2.06	**95.31**	**37.89%**
ℓ_∞ ($\epsilon = 0.2$)	*avg.*	1.05	0.07	41.10	35.03	48.17	-
	min max	2.47	0.37	7.39	5.81	**90.16**	**87.17%**

Table 2: Comparison of min-max and heuristic weighting schemes (ℓ_∞-attack, $\epsilon = 0.2$).

Opt.	Acc_A	Acc_B	Acc_C	Acc_D	ASR_{avg}	ASR_{all}	Lift (↑)
avg.	1.05	0.07	41.10	35.03	80.69	48.17	-
w_{c+d}	60.37	19.55	15.10	1.87	75.78	29.32	-39.13%
w_{a+c+d}	0.46	21.57	25.36	13.84	84.69	53.39	10.84%
w_{clip} [48]	0.66	0.03	23.43	13.23	90.66	71.54	48.52%
w_{prior}	1.57	0.24	17.67	13.74	91.70	74.34	54.33%
w_{static}	10.58	0.39	9.28	10.05	92.43	77.84	61.59%
min max	2.47	0.37	7.39	5.81	**95.99**	**90.16**	**87.17%**

4 Experiments on Adversarial Exploration

In this section, we first evaluate the proposed min-max optimization strategy on three attack tasks. We show that our approach leads to substantial improvement compared with state-of-the-art attack methods such as ensemble PGD [30] and EOT [3, 8, 5]. We also demonstrate the effectiveness of learnable domain weights in guiding the adversary's exploration over multiple domains.

4.1 Experimental setup

We thoroughly evaluate our algorithm on MNIST and CIFAR-10. A set of diverse image classifiers (denoted from Model A to Model H) are trained, including multi-layer perceptron (MLP), All-CNNs [51], LeNet [27], LeNetV2, VGG16 [49], ResNet50 [21], Wide-ResNet [35, 63] and GoogLeNet [53]. The details about model architectures and training process are provided in Appendix D.1. Note that problem formulations (5)-(7) are applicable to both *untargeted* and *targeted* attack. Here we focus on the former setting and use C&W loss function [11, 35]. The details of crafting adversarial examples are available in Appendix D.2.

4.2 Ensemble Attack over Multiple Models

We craft adversarial examples against an ensemble of known classifiers. Recent work [30] proposed an ensemble PGD attack, which assumed equal importance among different models, namely, $w_i = 1/K$ in problem (5). Throughout this task, we measure the attack performance via $\mathbf{ASR}_{all}$ - the attack success rate (ASR) of fooling model ensembles simultaneously. Compared to the ensemble PGD attack, our approach results in 40.79% and 17.48% ASR_{all} improvement averaged over different ℓ_p-norm constraints on MNIST and CIFAR-10, respectively. In what follows, we provide more detailed results and analysis.

In Table 1, we show that AMGDA significantly outperforms ensemble PGD in ASR_{all}. Taking ℓ_∞-attack on MNIST as an example, our min-max attack leads to a 90.16% ASR_{all}, which largely outperforms 48.17%. The reason is that Model C, D are more difficult to attack, which can be observed from their higher test accuracy on adversarial examples. As a result, although the adversarial examples crafted by assigning equal weights over multiple models are able to attack {A, B} well, they achieve a much lower ASR in {C, D}. By contrast, APGDA automatically handles the worst case {C, D} by slightly sacrificing the performance on {A, B}: 31.47% averaged ASR improvement on {C, D} versus 0.86% degradation on {A, B}. More results on CIFAR-10 are provided in Appendix E.

Table 3: Comparison of average and minmax optimization on universal perturbation over multiple input examples. The adversarial examples are generated by 20-step ℓ_∞-APGDA.

Setting			$K=2$			$K=4$			$K=5$			$K=10$		
Dataset	Model	Opt.	ASR_{avg}	ASR_{all}	Lift (↑)	ASR_{avg}	ASR_{all}	Lift (↑)	ASR_{avg}	ASR_{all}	Lift (↑)	ASR_{avg}	ASR_{all}	Lift (↑)
CIFAR-10	All-CNNs	*avg.*	91.09	83.08	-	85.66	54.72	-	82.76	40.20	-	71.22	4.50	-
		min max	92.22	**85.98**	**3.49%**	87.63	**65.80**	**20.25%**	85.02	**55.74**	**38.66%**	65.64	**11.80**	**162.2%**
	LeNetV2	*avg.*	93.26	86.90	-	90.04	66.12	-	88.28	55.00	-	72.02	8.90	-
		min max	93.34	**87.08**	**0.21%**	91.91	**71.64**	**8.35%**	91.21	**63.55**	**15.55%**	82.85	**25.10**	**182.0%**
	VGG16	*avg.*	90.76	82.56	-	89.36	63.92	-	88.74	55.20	-	85.86	22.40	-
		min max	92.40	**85.92**	**4.07%**	90.04	**70.40**	**10.14%**	88.97	**63.30**	**14.67%**	79.07	**30.80**	**37.50%**
	GoogLeNet	*avg.*	85.02	72.48	-	75.20	32.68	-	71.82	19.60	-	59.01	0.40	-
		min max	87.08	**77.82**	**7.37%**	77.05	**46.20**	**41.37%**	71.20	**33.70**	**71.94%**	45.46	**2.40**	**600.0%**

Table 4: Interpretability of domain weight w for universal perturbation to multiple inputs on MNIST (*Digit 0, 2, 4*). Domain weight w for different images under ℓ_p-norm ($p = 0, 1, 2, \infty$).

Image																
Weight	ℓ_0	0.	0.	0.	0.	1.000	0.	0.	0.909	0.	0.091	0.	0.	0.753	0.	0.247
	ℓ_1	0.	0.	0.	0.	1.000	0.	0.	0.843	0.	0.157	0.018	0.	0.567	0.	0.416
	ℓ_2	0.	0.	0.	0.	1.000	0.	0.	0.788	0.	0.112	0.	0.	0.595	0.	0.405
	ℓ_∞	0.	0.	0.	0.	1.000	0.	0.	0.850	0.	0.150	0.	0.	0.651	0.	0.349
Metric	dist.(C&W ℓ_2)	1.839	1.954	1.347	1.698	**3.041**	1.928	1.439	2.312	1.521	**2.356**	1.558	1.229	**1.939**	0.297	1.303
	ϵ_{min} (ℓ_∞)	0.113	0.167	0.073	0.121	**0.199**	0.082	0.106	**0.176**	0.072	0.171	0.084	0.088	**0.122**	0.060	0.094

Effectiveness of learnable domain weights: Figure 1 depicts the ASR of four models under average/min-max attacks as well as the distribution of domain weights during attack generation. For ensemble PGD (Figure 1a), Model C and D are attacked insufficiently, leading to relatively low ASR and thus weak ensemble performance. By contrast, APGDA (Figure 1b) will encode the difficulty level to attack different models based on the current attack loss. It dynamically adjusts the weight w_i as shown in Figure 1c. For instance, the weight for Model D is first raised to 0.45 because D is difficult to attack initially. Then it decreases to 0.3 once Model D encounters the sufficient attack power and the corresponding attack performance is no longer improved. It is worth noticing that APGDA is highly efficient because w_i converges after a small number of iterations. Figure 1c also shows $w_c > w_d > w_a > w_b$ – indicating a decrease in model robustness for C, D, A and B, which is exactly verified by $\text{Acc}_C > \text{Acc}_D > \text{Acc}_A > \text{Acc}_B$ in the last row of Table 1 (ℓ_∞-norm). As the perturbation radius ϵ varies, we also observe that the ASR of min-max strategy is consistently better or on part with the average strategy (see Figure 2).

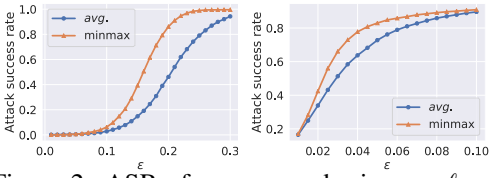


Figure 2: ASR of average and min-max ℓ_∞ ensemble attack versus maximum perturbation magnitude ϵ. Left (MNIST), Right (CIFAR-10).

Comparison with stronger heuristic baselines Apart from *average* strategy, we compare min-max framework with stronger heuristic weighting scheme - loss clipping [48] in Table 2. Specifically, with the prior knowledge of robustness of given models ($C > D > A > B$), we devised several heuristic baselines including: (a) w_{c+d}: ensemble PGD on models C and D only; (b) w_{a+c+d}: ensemble PGD on models A, C and D only; (c) w_{clip}: clipped version of C&W loss (threshold $\beta = 40$) to balance model weights in optimization as suggested in [48]; (d) w_{prior}: larger weights on the more robust models, $w_{prior} = [w_A, w_B, w_C, w_D] = [0.2, 0.1, 0.4, 0.3]$; (e) w_{static}: the converged mean weights of min-max (APGDA) ensemble attack. For ℓ_2 ($\epsilon = 3.0$) and ℓ_∞ ($\epsilon = 0.2$) attacks, $w_{static} = [w_A, w_B, w_C, w_D]$ are $[0.209, 0.046, 0.495, 0.250]$ and $[0.080, 0.076, 0.541, 0.303]$, respectively. Table 2 shows that our approach achieve substantial improvement over baselines consistently. Moreover, we highlight that the use of learnable $\mathbf{w}$ avoids supervised manual adjustment on the heuristic weights or the choice of clipping threshold. Also, we show that even adopting converged min-max weights statically leads to a huge performance drop on attacking model ensembles, which again verifies the power of dynamically optimizing domain weights during attack generation process.

4.3 Multi-Image Universal Perturbation

We evaluate APGDA in universal perturbation on MNIST and CIFAR-10, where 10,000 test images are randomly divided into equal-size groups (K images per group) for universal perturbation. We measure two types of ASR (%), $\mathbf{ASR}_{avg}$ and $\mathbf{ASR}_{all}$. Here the former represents the ASR averaged over all images in all groups, and the latter signifies the ASR averaged over all groups but a successful attack is counted under a more restricted condition: images within each group must be successfully attacked simultaneously by universal perturbation. In Table 3, we compare the proposed min-max strategy with the averaging strategy on the attack performance of generated universal perturbations. APGDA always achieves higher ASR_{all} for different values of K. When $K = 5$, our approach achieves 42.63% and 35.21% improvement over the averaging strategy under MNIST and CIFAR-10. The universal perturbation generated from APGDA can successfully attack 'hard' images (on which the average-based PGD attack fails) by self-adjusting domain weights, and thus leads to a higher ASR_{all}.

Interpreting "*image robustness*" with domain weights $\mathbf{w}$: The min-max universal perturbation also offers interpretability of "*image robustness*" by associating domain weights with image visualization. Figure 4 shows an example in which the large domain weight corresponds to the MNIST letter with clear appearance (e.g., bold letter). To empirically verify the robustness of image, we report two metrics to measure the difficulty of attacking single image: dist. (C&W ℓ_2) denotes the the minimum distortion of successfully attacking images using C&W (ℓ_2) attack; $\epsilon_{\min}$ (ℓ_∞) denotes the minimum perturbation magnitude for ℓ_∞-PGD attack.

4.4 Robust Attack over Data Transformations

EOT [5] achieves state-of-the-art performance in producing adversarial examples robust to data transformations. From (7), we could derive EOT as a special case when the weights satisfy $w_i = 1/K$ (average case). For each input sample (*ori*), we transform the image under a series of functions, e.g., flipping horizontally (*flh*) or vertically (*flv*), adjusting brightness (*bri*), performing gamma correction (*gam*) and cropping (*crop*), and group each image with its transformed variants. Similar to universal perturbation, ASR_{all} is reported to measure the ASR over groups of transformed images (each group is successfully attacked signifies successfully attacking an example under all transformers). In Table 5, compared to EOT, our approach leads to 9.39% averaged lift in ASR_{all} over given models on CIFAR-10 by optimizing the weights for various transformations. we leave the the results under randomness (e.g., flipping images randomly *w.p.* 0.8; randomly clipping the images at specific range) in Appendix E.

Table 5: Comparison of average and min-max optimization on robust attack over multiple data transformations on CIFAR-10.

Model	Opt.	Acc_{ori}	Acc_{flh}	Acc_{flv}	Acc_{bri}	Acc_{gam}	Acc_{crop}	ASR_{all}	Lift (↑)
A	*avg.*	10.80	21.93	14.75	11.52	10.66	20.03	55.88	-
	min max	12.14	18.05	13.61	13.52	11.99	16.78	**60.03**	**7.43%**
B	*avg.*	5.49	11.56	9.51	5.43	5.75	15.89	72.21	-
	min max	6.22	8.61	9.74	6.35	6.42	11.99	**77.43**	**7.23%**
C	*avg.*	7.66	21.88	15.50	8.15	7.87	15.36	56.51	-
	min max	8.51	14.75	13.88	9.16	8.58	13.35	**63.58**	**12.51%**
D	*avg.*	8.00	20.47	13.46	7.73	8.52	15.90	61.13	-
	min max	9.19	13.18	12.72	8.79	9.18	13.11	**67.49**	**10.40%**

5 Extension: Understanding Defense over Multiple Perturbation Domains

In this section, we show that the min-max principle can also be used to gain more insights in generalized adversarial training (AT) from a defender's perspective. Different from promoting robustness of adversarial examples against the *worst-case attacking environment* (Sec. 3), the generalized AT promotes model's robustness against the *worst-case defending environment*, given by the existence of multiple ℓ_p attacks [55]. Our approach obtains better performance than prior works [55, 36] and interpretability by introducing the trainable domain weights.

5.1 Adversarial Training under Mixed Types of Adversarial Attacks

Conventional AT is restricted to a single type of norm-ball constrained adversarial attack [35]. For example, AT under ℓ_∞ attack yields:

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \ \mathbb{E}_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \underset{\|\boldsymbol{\delta}\|_\infty \le \epsilon}{\text{maximize}} \ f_{\text{tr}}(\boldsymbol{\theta}, \boldsymbol{\delta}; \mathbf{x}, y), \tag{10}$$

	MAX [3]	AVG [3]	MSD [2]	AMPGD
Clean Accuracy	98.6%	99.1%	98.3%	98.3%
ℓ_∞ Attacks [55] ($\epsilon = 0.3$)	51.0%	65.2%	62.7%	76.1%
ℓ_2 Attacks [55] ($\epsilon = 2.0$)	61.9%	60.1%	67.9%	70.2%
ℓ_1 Attacks [55] ($\epsilon = 10$)	52.6%	39.2%	65.0%	67.2%
All Attacks [55]	42.1%	34.9%	58.4%	**64.1%**
AA (all attacks) [16]	36.9%	30.5%	55.9%	59.3%
AA+ (all attacks) [16]	34.3%	28.8%	54.8%	**58.3%**

Table 6: Adversarial robustness on MNIST.

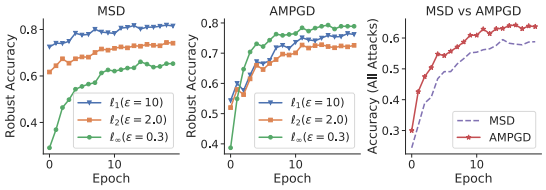


Figure 3: Robust accuracy of MSD and AMPGD.

where $\boldsymbol{\theta} \in \mathbb{R}^n$ denotes model parameters, $\boldsymbol{\delta}$ denotes ϵ-tolerant ℓ_∞ attack, and $f_{\text{tr}}(\boldsymbol{\theta}, \boldsymbol{\delta}; \mathbf{x}, y)$ is the training loss under perturbed examples $\{(\mathbf{x} + \boldsymbol{\delta}, y)\}$. However, there possibly exist blind attacking spots across multiple types of adversarial attacks so that AT under one attack would not be strong enough against another attack [2]. Thus, an interesting question is how to generalize AT under multiple types of adversarial attacks [55]. One possible way is to use the finite-sum formulation in the inner maximization problem of (10), namely, $\underset{\{\boldsymbol{\delta}_i \in \mathcal{X}_i\}}{\text{maximize}} \frac{1}{K} \sum_{i=1}^K f_{\text{tr}}(\boldsymbol{\theta}, \boldsymbol{\delta}_i; \mathbf{x}, y)$, where $\boldsymbol{\delta}_i \in \mathcal{X}_i$ is the ith type of adversarial perturbation defined on $\mathcal{X}_i$, e.g., different ℓ_p attacks.

Since we can map 'attack type' to 'domain' considered in (1), AT can be generalized against the *strongest* adversarial attack across K attack types in order to avoid blind attacking spots:

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \ \mathbb{E}_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} \underset{i \in [K]}{\text{maximize}} \underset{\boldsymbol{\delta}_i \in \mathcal{X}_i}{\text{maximize}} \ f_{\text{tr}}(\boldsymbol{\theta}, \boldsymbol{\delta}_i; \mathbf{x}, y). \tag{11}$$

In Lemma 1, we show that problem (11) can be equivalently transformed into the min-max form.

Lemma 1. *Problem (11) is equivalent to:*

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \ \mathbb{E}_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} \underset{\mathbf{w} \in \mathcal{P}, \{\boldsymbol{\delta}_i \in \mathcal{X}_i\}}{\text{maximize}} \sum_{i=1}^K w_i f_{\text{tr}}(\boldsymbol{\theta}, \boldsymbol{\delta}_i; \mathbf{x}, y), \tag{12}$$

where $\mathbf{w} \in \mathbb{R}^K$ represent domain weights, and $\mathcal{P}$ has been defined in (1).

The proof of Lemma 1 is provided in Appendix B. Similar to (4), a strongly concave regularizer $-\gamma/2\|\mathbf{w} - \mathbf{1}/K\|_2^2$ can be added into the inner maximization problem of (12) for boosting the stability of the learning procedure and striking a balance between the max and the average attack performance:

$$\begin{aligned} &\underset{\boldsymbol{\theta}}{\text{minimize}} \ \mathbb{E}_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} \underset{\mathbf{w} \in \mathcal{P}, \{\boldsymbol{\delta}_i \in \mathcal{X}_i\}}{\text{maximize}} \ \psi(\boldsymbol{\theta}, \mathbf{w}, \{\boldsymbol{\delta}_i\}) \\ &\psi(\boldsymbol{\theta}, \mathbf{w}, \{\boldsymbol{\delta}_i\}) := \textstyle\sum_{i=1}^K w_i f_{\text{tr}}(\boldsymbol{\theta}, \boldsymbol{\delta}_i; \mathbf{x}, y) - \frac{\gamma}{2}\|\mathbf{w} - \mathbf{1}/K\|_2^2 \end{aligned} \tag{13}$$

We propose the **a**lternating **m**ulti-step **p**rojected **g**radient **d**escent (AMPGD) method (Algorithm 2) to solve problem (13). Since AMPGD also follows the min-max principles, we defer more details of this algorithm in Appendix C. We finally remark that our formulation of generalized AT under multiple perturbations covers prior work [55] as special cases ($\gamma = 0$ for max case and $\gamma = 0$ for average case).

Algorithm 2 AMPGD to solve problem (13)

1: Input: given $\boldsymbol{\theta}^{(0)}$, $\mathbf{w}^{(0)}$, $\boldsymbol{\delta}^{(0)}$ and $K > 0$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: given $\mathbf{w}^{(t-1)}$ and $\boldsymbol{\delta}^{(t-1)}$, perform SGD to update $\boldsymbol{\theta}^{(t)}$
4: given $\boldsymbol{\theta}^{(t)}$, perform R-step PGD to update $\mathbf{w}^{(t)}$ and $\boldsymbol{\delta}^{(t)}$
5: **end for**

5.2 Generalized AT vs. Multiple ℓ_p Attacks

Compared to vanilla AT, we show the generalized AT scheme produces model robust to multiple types of perturbation, thus leads to stronger "overall robustness". We present experimental results of generalized AT following [36] to achieve simultaneous robustness to ℓ_∞, ℓ_2, and ℓ_1 perturbations on the MNIST and CIFAR-10 datasets. To the best of our knowledge, MSD proposed in [36] is the state-of-the-art defense against multiple types of ℓ_p attacks. Specifically, we adopted the same architectures as [36] four layer convolutional networks on MNIST and the pre-activation version of the ResNet18 [21]. The perturbation radius ϵ for $(\ell_\infty, \ell_2, \ell_1)$ balls is set as $(0.3, 2.0, 10)$ and $(0.03, 0.5, 12)$ on MNIST and CIFAR-10 following [36]. Apart from the evaluation ℓ_p PGD attacks,

	L_∞-AT	L_2-AT	L_1-AT	MAX [55]	AVG [56]	MSD [36]	AMPGD
Clean Accuracy	83.3%	90.2%	73.3%	81.0%	84.6%	81.1%	81.5%
ℓ_∞ Attacks ($\epsilon = 0.03$) [36]	50.7%	28.3%	0.2%	44.9%	42.5%	48.0%	49.2%
ℓ_2 Attacks ($\epsilon = 0.5$) [36]	57.3%	61.6%	0.0%	61.7%	65.0%	64.3%	68.0%
ℓ_1 Attacks ($\epsilon = 12$) [36]	16.0%	46.6%	7.9%	39.4%	54.0%	53.0%	50.0%
All Attacks [36]	15.6%	27.5%	0.0%	34.9%	40.6%	47.0%	**48.7%**
AA ($\ell_\infty, \epsilon = 0.03$) [16]	47.8%	22.7%	0.0%	39.2%	40.7%	44.4%	46.9%
AA ($\ell_2, \epsilon = 0.5$) [16]	57.5%	63.1%	0.1%	62.0%	65.5%	64.9%	64.4%
AA ($\ell_1, \epsilon = 12$) [16]	13.7%	23.6%	1.4%	36.0%	58.8%	52.4%	52.3%
AA (all attacks) [16]	12.8%	18.4%	0.0%	30.8%	40.4%	44.1%	**46.2%**

Table 7: Summary of adversarial accuracy results for CIFAR-10.

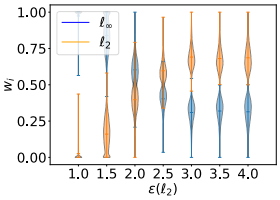


Figure 4: Domain weights.

we also incorporate the state-of-the-art AutoAttack [16] for a more comprehensive evaluation under mixed ℓ_p perturbations.

The adversarial accuracy results on and are reported (higher the better). As shown in Table 6 and 7, our approach outperforms the state-of-the-art defense MSD consistently (4∼6% and 2% improvements on MNIST and CIFAR-10). Compared to MSD that deploys an approximate arg max operation to select the steepest-descent (worst-case) universal perturbation, we leverage the domain weights to self-adjust the strengthens of diverse ℓ_p attacks. Thus, we believe that this helps gain supplementary robustness from individual attacks.

Effectiveness of Domain Weights: Figure 3 shows the robust accuracy curves of MSD and AMPGD on MNIST. As we can see, the proposed AMPGD can quickly adjust the defense strengths to focus on more difficult adversaries - the gap of robust accuracy between three attacks is much smaller. Therefore, it achieves better results by avoiding the trade-off that biases one particular perturbation model at the cost of the others. In Figure 4, we offer deeper insights on how the domain weights work as the strengths of adversary vary. Specifically, we consider two perturbation models on MNIST: ℓ_2 and ℓ_∞. During the training, we fix the ϵ for ℓ_∞ attack during training as 0.2, and change the ϵ for ℓ_2 from 1.0 to 4.0. As shown in Figure 4, the domain weight w increases when the ℓ_2-attack becomes stronger i.e., $\epsilon(\ell_2)$ increases, which is consistent with min-max spirit – defending the strongest attack.

5.3 Additional Discussions of APGDA and AMPGD

More parameters to tune for min-max? Our min-max approaches (APGDA and AMPGD) introduce two more hyperparameters - β and γ. However, our proposal performs reasonably well by choosing the learning rate α same as standard PGD and using a large range of regularization coefficient $\gamma \in [0, 10]$; see Fig. A5 in Appendix. For the learning rate β to update domain weights, we found $1/T$ is usually a very good practice, where T is the total number of attack iterations.

Time complexity of inner maximization? Our proposal achieves significant improvements at a low cost of extra computation. Specifically, (1) our APGDA attack is $1.31\times$ slower than the average PGD; (2) our AMPGD defense is $1.15\times$ slower than average or max AT [55].

6 Conclusion and Broad Impact

In this paper, we revisit the strength of min-max optimization in the context of adversarial attack generation. Beyond adversarial training (AT), we show that many attack generation problems can be re-formulated in our unified min-max framework, where the maximization is taken over the probability simplex of the set of domains. Experiments show our min-max attack leads to significant improvements on three tasks. Importantly, we demonstrate the self-adjusted domain weights not only stabilize the training procedure but also provides a holistic tool to interpret the risk of different domain sources. Our min-max principle also helps understand the generalized AT against multiple adversarial attacks. Our approach results in superior performance as well as intepretability.

Our work is a beneficial supplement to building trustworthy AI systems, in particular for safety-critical AI applications, such as autonomous vehicles and camera surveillance. We do not see negative impacts of our work on its ethical aspects and future societal consequences.

References

[1] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *ECCV (23)*, volume 12368 of *Lecture Notes in Computer Science*, pages 484–501. Springer, 2020.

[2] A. Araujo, R. Pinot, B. Negrevergne, L. Meunier, Y. Chevaleyre, F. Yger, and J. Atif. Robust neural networks using randomized adversarial training. *arXiv preprint arXiv:1903.10219*, 2019.

[3] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[4] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 284–293. PMLR, 2018.

[5] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 284–293, 10–15 Jul 2018.

[6] A. Athalye and I. Sutskever. Synthesizing robust adversarial examples. *ICML*, 2018.

[7] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[8] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017.

[9] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. J. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. *CoRR*, abs/1902.06705, 2019.

[10] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. Hidden voice commands. In *USENIX Security Symposium*, pages 513–530, 2016.

[11] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017.

[12] N. Carlini and D. A. Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *IEEE Symposium on Security and Privacy Workshops*, pages 1–7. IEEE Computer Society, 2018.

[13] H. Chen, H. Zhang, P.-Y. Chen, J. Yi, and C.-J. Hsieh. Attacking visual language grounding with adversarial examples: A case study on neural image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 2587–2597, 2018.

[14] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 2019.

[15] F. Croce and M. Hein. Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. *arXiv preprint arXiv:1905.11213*, 2019.

[16] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 2206–2216. PMLR, 2020.

[17] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

[18] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *CVPR*, pages 1625–1634. IEEE Computer Society, 2018.

[19] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *2015 ICLR*, arXiv preprint arXiv:1412.6572, 2015.

[20] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] M. Hein and M. Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*, pages 2266–2276, 2017.

[23] S. Huang, N. Papernot, I. J. Goodfellow, Y. Duan, and P. Abbeel. Adversarial attacks on neural network policies. In *ICLR (Workshop)*. OpenReview.net, 2017.

[24] R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*, pages 2021–2031. Association for Computational Linguistics, 2017.

[25] D. Kang, Y. Sun, D. Hendrycks, T. Brown, and J. Steinhardt. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019.

[26] H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

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

[28] Q. Lei, L. Wu, P.-Y. Chen, A. G. Dimakis, I. S. Dhillon, and M. Witbrock. Discrete adversarial attacks and submodular optimization with applications to text classification. *SysML*, 2019.

[29] Y. Lin, Z. Hong, Y. Liao, M. Shih, M. Liu, and M. Sun. Tactics of adversarial attack on deep reinforcement learning agents. In *IJCAI*, pages 3756–3762. ijcai.org, 2017.

[30] J. Liu, W. Zhang, and N. Yu. CAAD 2018: Iterative ensemble adversarial attack. *CoRR*, abs/1811.03456, 2018.

[31] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*. OpenReview.net, 2017.

[32] S. Lu, R. Singh, X. Chen, Y. Chen, and M. Hong. Understand the dynamics of GANs via primal-dual optimization, 2019.

[33] S. Lu, I. Tsaknakis, and M. Hong. Block alternating optimization for non-convex min-max problems: Algorithms and applications in signal processing and communications. 2018.

[34] S. Lu, I. Tsaknakis, and M. Hong. Block alternating optimization for non-convex min-max problems: algorithms and applications in signal processing and communications. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[35] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[36] P. Maini, E. Wong, and J. Z. Kolter. Adversarial robustness against the union of multiple perturbation models. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 6640–6650. PMLR, 2020.

[37] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.

[38] J. H. Metzen, M. C. Kumar, T. Brox, and V. Fischer. Universal adversarial perturbations against semantic image segmentation. In *ICCV*, pages 2774–2783. IEEE Computer Society, 2017.

[39] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 86–94, 2017.

[40] S. M. Moosavi Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number EPFL-CONF-218057, 2016.

[41] M. Nouiehed, M. Sanjabi, J. D. Lee, and M. Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *arXiv preprint arXiv:1902.08297*, 2019.

[42] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[43] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016.

[44] N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[45] Q. Qian, S. Zhu, J. Tang, R. Jin, B. Sun, and H. Li. Robust optimization over multiple domains. *CoRR*, abs/1805.07588, 2018.

[46] H. Rafique, M. Liu, Q. Lin, and T. Yang. Non-convex min-max optimization: Provable algorithms and applications in machine learning. *arXiv preprint arXiv:1810.02060*, 2018.

[47] L. Rice, E. Wong, and J. Z. Kolter. Overfitting in adversarially robust deep learning. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 8093–8104. PMLR, 2020.

[48] A. Shafahi, M. Najibi, Z. Xu, J. P. Dickerson, L. S. Davis, and T. Goldstein. Universal adversarial training. *CoRR*, abs/1811.11304, 2018.

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[50] A. Sinha, H. Namkoong, and J. Duchi. Certifying some distributional robustness with principled adversarial training. 2018.

[51] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (Workshop)*, 2015.

[52] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018.

[53] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9. IEEE Computer Society, 2015.

[54] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[55] F. Tramèr and D. Boneh. Adversarial training and robustness for multiple perturbations. *arXiv preprint arXiv:1904.13000*, 2019.

[56] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *2018 ICLR*, arXiv preprint arXiv:1705.07204, 2018.

[57] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

[58] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*. OpenReview.net, 2020.

[59] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[60] K. Xu, S. Liu, P. Zhao, P.-Y. Chen, H. Zhang, Q. Fan, D. Erdogmus, Y. Wang, and X. Lin. Structured adversarial attack: Towards general implementation and better interpretability. In *International Conference on Learning Representations*, 2019.

[61] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P. Chen, Y. Wang, and X. Lin. Adversarial t-shirt! evading person detectors in a physical world. In *ECCV (5)*, volume 12350 of *Lecture Notes in Computer Science*, pages 665–681. Springer, 2020.

[62] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *NDSS*. The Internet Society, 2018.

[63] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*. BMVA Press, 2016.

[64] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.

[65] Z. Zhao, D. Dua, and S. Singh. Generating natural adversarial examples. In *ICLR*. OpenReview.net, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] see Section 5.3.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] All experiments are run 3 times with different random seeds, and the mean metrics are reported.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We provide code and pre-trained models for reproducibility.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]