# Generalist: Decoupling Natural and Robust Generalization

Hongjun Wang[1*] Yisen Wang[1,2†]
[1] National Key Lab of General Artificial Intelligence
School of Intelligence Science and Technology, Peking University
[2] Institute for Artificial Intelligence, Peking University

## Abstract

*Deep neural networks obtained by standard training have been constantly plagued by adversarial examples. Although adversarial training demonstrates its capability to defend against adversarial examples, unfortunately, it leads to an inevitable drop in the natural generalization. To address the issue, we decouple the natural generalization and the robust generalization from joint training and formulate different training strategies for each one. Specifically, instead of minimizing a global loss on the expectation over these two generalization errors, we propose a bi-expert framework called* Generalist *where we simultaneously train base learners with task-aware strategies so that they can specialize in their own fields. The parameters of base learners are collected and combined to form a global learner at intervals during the training process. The global learner is then distributed to the base learners as initialized parameters for continued training. Theoretically, we prove that the risks of Generalist will get lower once the base learners are well trained. Extensive experiments verify the applicability of Generalist to achieve high accuracy on natural examples while maintaining considerable robustness to adversarial ones. Code is available at* https://github.com/PKU-ML/Generalist.

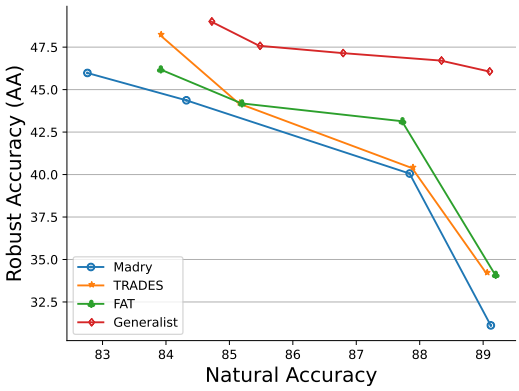



Figure 1. Comparison with other advanced adversarial training methods. Both clean accuracy and robust accuracy (against AutoAttack [9]) are given for a handy reference. It is noted that current adversarial training methods achieve high clean accuracy by greatly sacrificing robustness. That means it is hard to obtain sufficient robustness but maintain high clean accuracy in the joint training framework. Our Generalist attains excellent clean accuracy while staying competitively robust. The improvement of Generalist is notable since we only use the naive cross-entropy loss with negligible computational overhead and even without increasing the model size.

## 1. Introduction

Modern deep learning techniques have achieved remarkable success in many fields, including computer vision [14, 16], natural language processing [10, 31], and speech recognition [28, 36]. Yet, deep neural networks (DNNs) suffer a catastrophic performance degradation by human imperceptible adversarial perturbations where wrong predictions are made with extremely high confidence [13, 29, 34]. The vulnerability of DNNs has led to the proposal of various defense approaches [3, 24, 25, 33, 40] for protecting DNNs from adversarial attacks. One of those representative techniques is adversarial training (AT) [20, 21, 37, 38], which dynamically injects perturbed examples that deceive the current model but preserve the right label into the training set. Adversarial training has been demonstrated to be the most effective method to improve adversarially robust generalization [2, 39].

Despite these successes, such attempts of adversarial training have found a tradeoff between natural and robust accuracy, *i.e.*, there exists an undesirable increase in the error on unperturbed images when the error on the worst-case perturbed images decreases, as illustrated in Figure 1. Prior works [30, 43] even argue that natural and robust accuracy are fundamentally at odds, which indicates that a robust classifier can be achieved only when compromising the natural generalization. However, the following works found that

*Work was done as an internship at Peking University. Now, he is a Ph.D. student at the University of Hong Kong.
†Corresponding Author: Yisen Wang (yisen.wang@pku.edu.cn)

the tradeoff may be settled in a roundabout way, such as incorporating additional labeled/unlabeled data [1, 8, 22, 26] or relaxing the magnitude of perturbations to generate suitable adversarial examples for better optimization [18, 44]. These works all focus on the data used for training while we propose to tackle the tradeoff problem from the perspective of the **training paradigm** in this paper.

Inspired by the spirit of the divide-and-conquer method, we decouple the objective function of adversarial training into two sub-tasks: one is used for natural example classification while the other one is used for adversarial example classification. Specifically, for each sub-task, we train a base learner on natural/adversarial datasets with the task-specific configuration while sharing the same model architecture. The parameters of base learners are collected and combined to form a global learner at intervals during the training process, which is then distributed to base learners as initialized parameters for continued training. We name the framework as *Generalist* whose proof-of-concept pipeline is shown in Figure 2. Different from the traditional joint training framework for natural and robust generalization, our proposed Generalist fully leverages task-specific information to individually train the base learners, which makes each sub-task to be solved better. Theoretically, we show that if the base learners are well trained, the final global learner is guaranteed to have a lower risk. Our proposed Generalist is the first to effectively address the tradeoff between natural and robust generalization by utilizing task-aware training strategies to achieve high clean accuracy in the natural setting, while also maintaining considerable robustness to the adversarial setting (as shown in Figure 1).

In summary, the main contributions are as follows:

- For the tradeoff between natural and robust generalization, previous methods have struggled to find a sweet point to meet both goals in the joint training framework. Here, we propose a novel Generalist paradigm, which constructs multiple task-aware base learners to respectively achieve the generalization goal on natural and adversarial counterparts separately.
- For each task, rather than being constricted in a stiff manner, every detail of the training strategies (*e.g.*, optimization scheme) can be totally customized, thus each base learner can better explore the optimal trajectory in its field while the global learner can fully leverage the merits of all base learners.
- We conduct extensive experiments in common settings against a wide range of adversarial attacks to demonstrate the effectiveness of our approach. Results show that our Generalist paradigm greatly improves both clean and robust accuracy on benchmark datasets compared to relevant techniques.

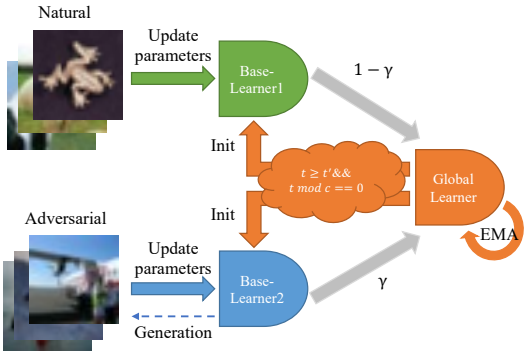



Figure 2. A pipeline for the proposed Generalist. It consists of two base learners separately trained within their respective fields and a global learner aggregates the parameters of base learners through the training process. The global learner assigns its accumulated knowledge to each base learner with a fixed frequency, based on which the base learner continues learning.

## 2. Preliminaries and Related Work

In this section, we briefly introduce some relevant background knowledge and terminology about adversarial training and meta-learning.

**Notations.** Consider an image classification task with input space $\mathcal{X}$ and output space $\mathcal{Y}$. Let $x \in \mathcal{X} \subseteq \mathbb{R}^d$ denote a natural image and $y \in \mathcal{Y} = \{1, 2, \ldots, K\}$ denote the corresponding ground-truth label. The natural and adversarial datasets $\mathcal{X} \times \mathcal{Y} = \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{X}' \times \mathcal{Y} = \{(x_i', y_i)\}_{i=1}^n$ are sampled from a distribution $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. We denote a DNN model as $f_\theta : \mathcal{X} \to \mathbb{R}^K$ whose parameters are $\boldsymbol{\theta} \in \Theta$, which should classify any input image into one of $K$ classes. The objective functions $\ell_1$ and $\ell_2$ for the natural and adversarial setting can be defined as: $\ell_1 \overset{def}{=} \mathcal{D}_1 \times \Theta \to [0, \infty)$ and $\ell_2 \overset{def}{=} \mathcal{D}_2 \times \Theta \to [0, \infty)$, which are usually positive, bounded, and upper-semi continuous [4, 6, 32].

### 2.1. Standard Adversarial Training

The goal of the adversary is to generate a malignant example $x'$ by adding an imperceptible perturbation $\varepsilon \in \mathbb{R}^d$ to $x$. And the generated adversarial example $x'$ should be in the vicinity of $x$ so that it looks visually similar to the original one. This neighbor region $\mathbb{B}_\varepsilon(x)$ anchored at $x$ with apothem $\varepsilon$ can be defined as $\mathbb{B}_\varepsilon(x) = \{(x', y) \in \mathcal{D}_2 \mid \|x - x'\|_\infty \leq \varepsilon\}$. For adversarial training, it first generates adversarial examples and then updates the parameters over these samples. The iteration process of adversarial training can be summed up as:

$$\begin{cases} x'^{(t+1)} = \Pi_{\mathbb{B}(x,\epsilon)}\left(x'^{(t)} + \alpha \operatorname{sign}\left(\nabla_{x'} \ell_2\left(x'^{(t)}, y; \boldsymbol{\theta}^t\right)\right)\right) \\ \boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \tau \nabla_{\boldsymbol{\theta}} \mathbb{E}[\ell_1(x, y; \boldsymbol{\theta}^t) + \beta \mathcal{R}(x', x, y; \boldsymbol{\theta}^t)], \end{cases} \tag{1}$$

where $\Pi_{\mathbb{B}(x,\epsilon)}$ is the projection operator, $\alpha$ is the step size, $\tau$ is the learning rate, and $\mathcal{R}(\cdot)$ is the loss difference of $\ell_2(x', y; \boldsymbol{\theta}^t) - \ell_1(x, y; \boldsymbol{\theta}^t)$. The tradeoff factor $\beta$ balances the importance of natural and robust errors. Various adversarial training methods can be derived from Eq. 1. For instance, when $\beta = 1$, it is equivalent to the vanilla PGD training [20], and when $\beta = 1/2$, it is transformed into the half-half loss in [13]. The formulation degenerates to standard natural training as $\beta = 0$. Besides, we can get the formulation in TRADES [43] when replacing $\mathcal{R}(\cdot)$ with the KL-divergence.

### 2.2. Multi-Task Learning and Meta-Initialization

**Multi-Task Learning.** Multi-Task Learning (MTL) is to improve performance across tasks through joint training of different models [5, 19, 41]. Consider a set of assignments containing data distribution and loss function defined as $\mathcal{A} = \{\mathcal{D}, \ell\}$ with corresponding models $\{\mathcal{M}_a\}_{a=1}^{|\mathcal{A}|}$ parameterized by trainable tensors $\boldsymbol{\theta}_{\mathcal{M}_a}$. In MTL, these sets have non-trivial pairwise intersections, and are trained in a joint model to find optimal parameters $\boldsymbol{\theta}^\star_{\mathcal{M}_a}$ for each task:

$$\bigcup_{a=1}^{|\mathcal{A}|} \boldsymbol{\theta}^\star_{\mathcal{M}_a} = \underset{\cup_{a=1}^{|\mathcal{A}|} \boldsymbol{\theta}_{\mathcal{M}_a}}{\operatorname{argmin}} \mathbb{E}_{\mathcal{A}} \mathbb{E}_{\mathcal{D}}\, \ell_a\left(\mathcal{D}_a; \boldsymbol{\theta}_{\mathcal{M}_a}\right), \tag{2}$$

where $\ell_a(\mathcal{D}_a; \boldsymbol{\theta}_{\mathcal{M}_a})$ measures the performance of a model trained using $\boldsymbol{\theta}_{\mathcal{M}_a}$ on dataset $\mathcal{D}_a$. Our approach Generalist is directly related to MTL at first glance because both of them tend to learn a specific predictive model for different sources. However, Generalist differs significantly from MTL, *i.e.*, multiple tasks are still learned *jointly* under a unified form in MTL while each assignment can be optimized by heterogeneous strategies in Generalist.

**Meta-Learning.** Meta-learning is to train a model that can quickly adapt to a new task. Suppose $\mathcal{A}$ is divided into non-overlapping splits $\mathcal{V}$ and $\mathcal{W}$, the model is first trained on the training sets and then guided by a small validation set on a set of tasks to make the trained model can be well adapted to new tasks:

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathbb{E}_{\mathcal{V}} \mathbb{E}_{\mathcal{D}_\mathcal{V}}\, \ell_\mathcal{V}\left(\mathcal{D}_\mathcal{V}; \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathbb{E}_\mathcal{W} \mathbb{E}_{\mathcal{D}_\mathcal{W}}\, \ell_\mathcal{W}\left(\mathcal{D}_\mathcal{W}; \boldsymbol{\theta}\right)\right), \tag{3}$$

meta-learning [12, 23] is often designed to generalize across unseen tasks, whereas the goal of MTL is to tackle a series of known tasks. Nonetheless, our approach Generalist uses the technique of meta-learning to set good initializations for base learners to transfer knowledge between tasks.

## 3. The Proposed New Framework: Generalist

Similar to a physical-world Generalist who has broad knowledge across many topics and expertise in a few, our proposed Generalist can deal with both natural and adversarial samples during test time. It consists of different base learners gradually specialized in their own disjointed fields. Over time they stretch their expertise to encompass knowledge respectively. From a starting point, Generalist takes his suitcase packed full of wide-ranging experience with him wherever it goes (*i.e.*, the global learner spreads accumulated knowledge and each expert learns from the re-initialization after a certain epoch).

**Algorithm 1** Generalist: Leverage the learning trajectory with respect to task-aware base learners

**Input:** A DNN classifier $f(\cdot)$ with initial learnable parameters $\boldsymbol{\theta}_g$ for the global learner and $\boldsymbol{\theta}_n, \boldsymbol{\theta}_r$ for each base learner with objective function $\ell_1, \ell_2$; number of iterations $T$; number of adversarial attack steps $K$; magnitude of perturbation $\varepsilon$; step size $\kappa$; learning rate $\tau_n, \tau_r$; exponential decay rates for ensembling $\alpha' = 0.999$; mixing ratio $\gamma$; starting point and frequency of communication $t', c$.
Initialize $\boldsymbol{\theta}_g, \boldsymbol{\theta}_n, \boldsymbol{\theta}_r$ in $\Theta$ space.
**for** t $\leftarrow 1, 2, \cdots, T$ **do**
  Sample a minibatch $(x, y)$ from data distribution $\mathcal{D}_1$
  */* Parallel-1: Update parameters of base learner-1 over $\mathcal{D}_1$ */*
  (Optional) Performing model ensembling, data augmentation or label smoothing, etc.
  $\boldsymbol{\theta}_n \leftarrow \mathcal{Z}_n\left[\mathbb{E}_{(x,y)}(\nabla_{\boldsymbol{\theta}} \ell_1(x, y; \boldsymbol{\theta}_n)), \tau_n\right]$
  */* Parallel-2: Update parameters of base learner-2 over $\mathcal{D}_2$ */*
  $x'_0 \leftarrow x + \varepsilon, \varepsilon \sim \text{Uniform}(-\varepsilon, \varepsilon)$.
  **for** k $\leftarrow 1, 2, \cdots, K$ **do**
    $x'_k \leftarrow \Pi_{x'_k \in \mathbb{B}_\varepsilon(x)}\left(\kappa \operatorname{sign}\left(x'_{k-1} + \nabla_{x'_{k-1}} \ell_2(x'_{k-1}, y; \boldsymbol{\theta}_r)\right)\right)$
  **end for**
  (Optional) Performing model ensembling, data augmentation or label smoothing, etc.
  $\boldsymbol{\theta}_r \leftarrow \mathcal{Z}_r\left[\mathbb{E}_{(x',y)}(\nabla_{\boldsymbol{\theta}} \ell_2(x'_K, y; \boldsymbol{\theta}_r)), \tau_r\right]$
  */* For the global learner*/*
  $\boldsymbol{\theta}_g \leftarrow \alpha' \boldsymbol{\theta}_g + (1 - \alpha')(\gamma \boldsymbol{\theta}_r + (1 - \gamma)\boldsymbol{\theta}_n)$
  **if** $t \geq t'$ and $t \mod c == 0$ **then**
    $\boldsymbol{\theta}_r, \boldsymbol{\theta}_n \leftarrow \boldsymbol{\theta}_g$
  **end if**
**end for**
**Return** Parameters of the global learner $\boldsymbol{\theta}_g$

### 3.1. Overview

The overall procedure of our proposed algorithm is shown in Algorithm 1, which mainly comprises two steps: optimizing parameters of the base learner $\boldsymbol{\theta}_a$ in its assigned data distribution $\mathcal{D}_a$ and distributing parameters of the global learner $\boldsymbol{\theta}_g$ to all base learners. Base learners and the global learner share the same architecture, *i.e.*, $\mathcal{M}_1 = \mathcal{M}_2 = \cdots = \mathcal{M}_{|\mathcal{A}|}$. Since we only focus on recognizing natural examples and adversarial ones in our setting, the total number of tasks $\mathcal{W}$ is set to two.

### 3.2. Task-aware Base Learners

Given a global data distribution $\mathcal{D}$ for the tradeoff problem, as denoted in Section 2, $\mathcal{D}_1, \mathcal{D}_2$ are subject to the distribution of training data $\mathcal{D}_\mathcal{W}$. And natural images $(x, y) \sim \mathcal{D}_1$ while adversarial examples $(x', y) \sim \mathcal{D}_2$ generated by Eq. 1. So the training process of base learners is to solve the inner minimization of Eq. 3 over different distributions in a

distributed manner:

$$\{\boldsymbol{\theta}_n^\star, \boldsymbol{\theta}_r^\star\} = \underset{\bigcup_{\mathcal{W}=1}^{2} \boldsymbol{\theta}_{\mathcal{W}}}{\operatorname{argmin}} \mathbb{E}_{\mathcal{D}_{\mathcal{W}}} \ell_{\mathcal{W}} \left(\mathcal{D}_{\mathcal{W}}; \boldsymbol{\theta}_{\mathcal{W}}\right). \tag{4}$$

Specifically, during the process, base learners $f_{\boldsymbol{\theta}_n}$ and $f_{\boldsymbol{\theta}_r}$ are assigned different subproblems that only requires accessing their own data distribution, respectively. Note that two base learners work in a complementary manner, meaning the update of parameters is independent among base learners and the global learner always collects parameters of both base learners. So the subproblem for each base learner is defined as:

$$\boldsymbol{\theta}_{\mathcal{W}}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{Z}_{\mathcal{W}}^T \left[\mathbb{E}_{\mathcal{W}}(\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{W}}(\mathcal{D}_{\mathcal{W}}; \boldsymbol{\theta}_{\mathcal{W}})), \tau_{\mathcal{W}}\right], \tag{5}$$

where the task-aware optimizer $\mathcal{Z}_{\mathcal{W}}^T(\cdot,\cdot)$ search the optimal parameter states $\boldsymbol{\theta}_{\mathcal{W}}^\star$ over the subproblem $\mathcal{W}$ in $T$ rounds. Loss functions can also be task-specific and applied to each base learner separately. It is natural to consider minimizing the 0-1 loss in the natural and robust errors, however, solving the optimization problem is NP-hard thus computationally intractable. In practice, we select cross-entropy as the surrogate loss for both $\ell_1$ and $\ell_2$ since it is simple but good enough.

### 3.3. Initialization from the Global Learner

During the initial training periods, base learners are less instrumental since they are not adequately learned. Directly initializing parameters of base learners may mislead the training procedure and further accumulate bias when mixing them. Therefore, we set aside $t'$ epochs from the beginning for fully training base learners and just aggregates states on the searching trajectory of base learners through optimization by exponential moving average (EMA), computed as: $\boldsymbol{\theta}_g \leftarrow \alpha' \boldsymbol{\theta}_g + (1-\alpha')(\gamma \boldsymbol{\theta}_r + (1-\gamma)\boldsymbol{\theta}_t)$, where $\alpha'$ is the exponential decay rates for EMA and $\gamma$ is the mixing ratio for base learners. They then learn an initialization from parameters of the global learner every $c$ epochs when each base learner is well trained in its field. Thus, the optimization of each base learner for every interlude can be expressed in Eq. 6:

$$\boldsymbol{\theta}_{\mathcal{W}}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{Z}_{\mathcal{W}}^c \left[\mathbb{E}_{\mathcal{W}}(\nabla_{\boldsymbol{\theta}} \ell_{\mathcal{W}}(\mathcal{D}_{\mathcal{W}}; \boldsymbol{\theta}_g)), \tau_{\mathcal{W}}\right]. \tag{6}$$

Note that $\boldsymbol{\theta}_g$ contains both $\boldsymbol{\theta}_n$ and $\boldsymbol{\theta}_r$, meaning there always exists a term updated by gradient information of distribution different from the current subproblem. This mechanism enables fast learning within a given assignment and improves generalization, and the acceleration is applicable to the given assignment for its corresponding base learner only (proof in Appendix B.1).

With all discussed above, the learning progress of Generalist can be constructed by decending the gradient of $\boldsymbol{\theta}_r, \boldsymbol{\theta}_n$ and mixing both of them. The calculating steps in Algorithm 1 can be summarized in Eq. 7.

$$\begin{cases} \boldsymbol{\theta}_n^t = \mathcal{Z}_n \left[\mathbb{E}_{(x,y)\sim\mathcal{D}_1}(\nabla_{\boldsymbol{\theta}_n} \ell_1(x, y; \boldsymbol{\theta}_n^{t-1})), \tau_1\right] \\ \boldsymbol{\theta}_r^t = \mathcal{Z}_r \left[(\mathbb{E}_{(x',y)\sim\mathcal{D}_2} \nabla_{\boldsymbol{\theta}_r} \ell_2(x', y; \boldsymbol{\theta}_r^{t-1})), \tau_2\right] \\ \boldsymbol{\theta}_g^{t+1} = \alpha' \boldsymbol{\theta}_g^{t-1} + (1-\alpha')[\gamma \boldsymbol{\theta}_r^t + (1-\gamma)\boldsymbol{\theta}_n^t] \\ \boldsymbol{\theta}_n^t = \mathcal{B}(t, t', c)\boldsymbol{\theta}_g^{t+1} + (1 - \mathcal{B}(t, t', c))\boldsymbol{\theta}_n^t \\ \boldsymbol{\theta}_r^t = \mathcal{B}(t, t', c)\boldsymbol{\theta}_g^{t+1} + (1 - \mathcal{B}(t, t', c))\boldsymbol{\theta}_r^t, \end{cases} \tag{7}$$

where $\mathcal{B}(t, t', c)$ is a Boolean function that returns one only when both $t \geq t'$ and $t \mod c == 0$, otherwise it returns zero. $\mathcal{Z}_n$ and $\mathcal{Z}_r$ are optimizers for natural training and adversarial training assignments.

### 3.4. Theoretical Analysis

In this part, we theoretically analyze how base learners help global learner in Generalist. For brevity, we omit the expectation notation over samples from each distribution without losing generalization.

**Definition 1.** ***(Tradeoff Regret with Mixed Strategies)*** *For the natural training assignment $a_1$ and adversarial training assignment $a_2$, consider an algorithm generates the trajectory of states $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ for two base learners, the regret of both base learners on its corresponding loss function $\ell_1$, $\ell_2$ is*

$$\mathbf{R}_T = \frac{1}{2} \sum_{a=1}^{2} \left( \sum_{t=1}^{T} \ell_a \left(\boldsymbol{\theta}_a^t\right) - \inf_{\boldsymbol{\theta}_a^t \in \Theta} \sum_{t=1}^{T} \ell_a \left(\boldsymbol{\theta}_a^t\right) \right). \tag{8}$$

The last term obtains the oracle state $\boldsymbol{\theta}_a^\star$, theoretically optimal parameters for each task $a$. $\mathbf{R}_T$ is the sum of the difference between the parameters of each base learner and the theoretically optimal parameters for each task. Based on the definition, we can give the following upper bound on the expected error of classifier trained by Generalist with respect to $\mathbf{R}_T$ as:

**Theorem 1.** *(Proof in Appendix B.2) Consider an algorithm with regret bound $R_T$ that generates the trajectory of states for two base learners, for any parameter state $\boldsymbol{\theta} \in \Theta$, given a sequence of convex surrogate evaluation functions $\ell : \Theta \mapsto [0,1]_{a\in\mathcal{A}}$ drawn i.i.d. from some distribution $\mathcal{L}$, the expected error of the global learner $\boldsymbol{\theta}_g$ on both tasks over the test set can be bounded with probability at least $1-\delta$:*

$$\mathop{\mathbb{E}}_{\ell\sim\mathcal{L}} \ell\left(\boldsymbol{\theta}_g\right) \leq \mathop{\mathbb{E}}_{\ell\sim\mathcal{L}} \ell\left(\boldsymbol{\theta}\right) + \frac{\mathbf{R}_T}{T} + 2\sqrt{\frac{2}{T} \log \frac{1}{\delta}}. \tag{9}$$

So the above inequality indicates that any strategy beneficial to reducing the error of each task that makes $\mathbf{R}_T$ smaller will decrease the error bound of the global learner. Considering Generalist divides the tradeoff problem into two independent tasks, Theorem 1 guarantees the upper bound of the risks given by the global learner trained by Generalist

will get lower once the error for each task becomes lower. In practice, we can apply customized learning rate strategies, optimizers, and weight averaging to guarantee the error reduction of each base learner.

# 4. Experiments

We conduct a series of experiments on ResNet-18 [14] and WRN-32-10 [42] on benchmark datasets MNIST, SVHN, CIFAR-10, and CIFAR-100 under the $L_\infty$ norm.

**Baselines.** We select six approaches to compare with: AT using PGD ($\beta = 1$ in Eq. 1) [20], AT using the half-half loss ($\beta = 1/2$ in Eq. 1) [13], TRADES with different $\lambda$ [43], Friendly Adversarial Training (FAT) [44], Interpolated Adversarial Training (IAT) [17], and Robust Self Training (RST) [26] used labeled data for fair comparison. For Generalist, we set $t' = 75$ and the optimal mixing strategy will be discussed in Section 4.2.2.

**Evaluation.** To evaluate the robustness of the proposed method, we apply several adversarial attacks including PGD [20], MIM [11], CW [7], AutoAttack (AA) [9] and all its components ($\mathrm{APGD}_{ce}$, $\mathrm{APGD}_{dlr}$, $\mathrm{APGD}_t$, $\mathrm{FAB}_t$, and Square attacks).

## 4.1. Tradeoff Performance on Benchmark Datasets

To comprehensively manifest the power of our Generalist method, we present the results of both ResNet-18 and WRN-32-10 on CIFAR-10 in Table 1.

In Table 1(a), Generalist consistently improves standard test error relative to models trained by several robust methods, while maintaining adversarial robustness at the same level. More specifically, Generalist achieves the second highest standard accuracy of 89.09% (only lower than 93.04% obtained by natural training (NT)), while meantime robust accuracy against AA is 46.07%, hanging on to 48.2% from TRADES. If we force TRADES to meet the same level of clean accuracy as Generalist (89%), the robustness of TRADES against APGD will drop to 30% (see TRADES in Appendix A.4), which is significantly worse than Generalist. That means it is hard to obtain acceptable robustness but maintain clean accuracy above 89% in the joint training framework even if it is equipped with an advanced loss function, while the improvement of Generalist is notable since we only use the naive cross-entropy loss. Contrary to FAT managing the tradeoff through adaptively decreasing the step size of PGD, which still hurts robustness a lot, Generalist is the only method with clean accuracy above 89% and robust accuracy against AA above 46%. We should emphasize the final obtained model of Generalist is the *same size* as other trained models are. For the training time, Generalist does perform both NT and naive AT but the cost of NT is negligible, so the overhead of Generalist is smaller than TRADES, and whatever serial and parallel versions of Generalist are even *faster* than TRADES (see Appendix A.4).

Things become more obvious when it comes to WRN-32-10. In Table 1(b), the gap between test natural accuracy of Generalist and NT is reduced to 2.27%, a relative decrease of 3.65% in standard test error as compared to the second highest natural accuracy (except NT) achieved by FAT. It is also remarkable that the boost of accuracy does not hurt the robustness of Generalist, instead, Generalist even outperforms TRADES across multiple types of adversarial attacks. In particular, we find that Generalist has a standard test error of 6.7% while TRADES with $\lambda = 6$ has a standard test error of 14.89% only. And the improved robustness of Generalist among PGD20/100, MIM, CW, $\mathrm{FAT}_t$ and Square is conspicuous. Besides, the best performance on AA, which is an ensemble of different attacks and the most powerful adaptive adversarial attack so far, demonstrates the reliability of Generalist. Likewise, only Generalist attains robust accuracy of AA higher than 52% along with clean accuracy higher than 90%. It should be emphasized that these features confirm the practicability of Generalist. In short, Generalist has consistently improved robustness without loss of natural accuracy. More results on benchmark datasets of MNIST, SVHN, and CIFAR-100 are in Appendix A.2 - A.3.

## 4.2. Comprehensive Understanding of Generalist

We run a number of ablations to analyze the Generalist framework in this part. As illustrated in Algorithm 1, two factors control the tradeoff between accuracy and robustness of the global learner: *frequency of communication* $c$ and *mixing ratio* $\gamma$. Here, we investigate how these parameters affect performance. If not specified otherwise, the experiments are conducted on CIFAR-10 using ResNet-18.

### 4.2.1 Mixing Strategies of $\gamma$

In Generalist, $\gamma$ controls the tradeoff via balancing the contribution of individuals to the global learner when base learners are gradually well trained. Note that $\gamma$ is a scalar but we do not explicitly assign a fixed value to it. Instead, we set several breakpoints and dynamically adjust the value along the training process using a piecewise linear function to decrease.

Results are shown in Figure 3. The numbers in brackets are the values at the 0/40/80/120-th epoch. If $\gamma$ gets smaller, the base learner in charge of natural classification has a pronounced influence on the global learner. Among all configurations, the best one is to apply $\gamma = (1, 1, 1, 0)$ and $c = 5$ to the global learner after the 75th epoch. When compared to strategies that $\gamma$ decays during late periods, $\gamma = (1, 1, 0.8, 0.2)$ shows lower standard and robust accuracy, confirming that more sophisticated initialization could be useful for both accuracy and robustness. With the increase of the last breakpoint of dynamical strategies, the robust accuracy gradually increases; while the standard ac-

Table 1. Comparison of our algorithm with different training methods using ResNet-18 and WRN-32-10 on CIFAR-10. The maximum perturbation is $\varepsilon = 8/255$. The best checkpoint is selected based on the tradeoff between clean accuracy and robust accuracy against PGD20 on the test set. We highlight the top two results on each task. We omit standard deviations of Generalist as they are very small ($< 0.5\%$). Average accuracy rates (in %) have shown that the proposed Generalist method greatly mitigates the tradeoff of the model.

(a) Evaluation results based on ResNet-18.

| Method | NAT | PGD20 | PGD100 | MIM | CW | $\text{APGD}_{ce}$ | $\text{APGD}_{dlr}$ | $\text{APGD}_t$ | $\text{FAT}_t$ | Square | AA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NT | **93.04** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| AT ($\beta = 1$) | 84.32 | 48.29 | 48.12 | 47.95 | 49.57 | **47.47** | 48.57 | 45.14 | 46.17 | 54.21 | 44.37 |
| AT ($\beta = 1/2$) | 87.84 | 44.51 | 44.53 | 47.30 | 44.93 | 40.58 | 42.55 | 40.20 | 44.56 | 50.76 | 40.06 |
| TRADES ($\lambda = 6$) | 83.91 | **54.25** | **52.21** | **55.65** | **52.22** | **53.47** | **50.89** | **48.23** | **48.53** | **55.75** | **48.20** |
| TRADES ($\lambda = 1$) | 87.88 | 45.58 | 45.60 | 47.91 | 45.05 | 42.95 | 42.49 | 40.38 | 43.89 | 53.49 | 40.32 |
| FAT | 87.72 | 46.69 | 46.81 | 47.03 | 49.66 | 46.20 | 47.51 | 44.88 | 45.76 | 52.98 | 43.14 |
| IAT | 84.60 | 40.83 | 40.87 | 43.07 | 39.57 | 37.56 | 37.95 | 35.13 | 36.06 | 49.30 | 35.13 |
| RST | 84.71 | 44.23 | 44.31 | 45.33 | 42.82 | 41.25 | 42.01 | 40.41 | 46.54 | 50.49 | 37.68 |
| Generalist | **89.09** | **50.01** | **50.00** | **52.19** | **50.04** | 46.53 | **48.70** | **46.37** | **47.32** | **56.68** | **46.07** |

(b) Evaluation results based on WRN-32-10.

| Method | NAT | PGD20 | PGD100 | MIM | CW | $\text{APGD}_{ce}$ | $\text{APGD}_{dlr}$ | $\text{APGD}_t$ | $\text{FAT}_t$ | Square | AA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NT | **93.30** | 0.01 | 0.02 | 0.05 | 0.00 | 0.00 | 0.00 | 0.00 | 0.87 | 0.28 | 0.00 |
| AT ($\beta = 1$) | 87.32 | 49.01 | 48.83 | 48.25 | 52.80 | 48.83 | 49.00 | 46.34 | 48.17 | 54.26 | 46.11 |
| AT ($\beta = 1/2$) | 89.27 | 48.95 | 48.86 | 51.35 | 49.56 | 45.98 | 47.66 | 44.89 | 46.42 | 56.83 | 44.81 |
| TRADES ($\lambda = 6$) | 85.11 | **54.58** | **54.82** | **55.67** | **54.91** | **54.89** | **55.50** | **52.71** | **52.61** | **57.62** | **52.19** |
| TRADES ($\lambda = 1$) | 87.20 | 51.33 | 51.65 | 52.47 | 53.19 | 51.60 | 51.88 | 49.97 | 50.01 | 54.83 | 49.81 |
| FAT | 89.65 | 48.74 | 48.69 | 48.24 | 52.11 | 48.50 | 48.81 | 46.70 | 46.17 | 51.51 | 44.73 |
| IAT | 87.93 | 50.55 | 50.72 | 52.37 | 48.71 | 47.71 | 46.55 | 43.84 | 45.78 | 56.52 | 43.80 |
| RST | 87.27 | 46.55 | 46.76 | 47.02 | 45.99 | 45.73 | 46.58 | 45.78 | 43.18 | 52.44 | 41.52 |
| Generalist | **91.03** | **56.88** | **56.92** | **58.87** | **57.23** | **53.94** | **55.80** | **53.00** | **53.65** | **63.10** | **52.91** |

curacy decreases by a small margin. We also investigate the static/dynamic strategy for $\gamma$. By observing $\gamma = 0.5$ and $\gamma = (1, 1, 1, 0.5)$, the scheduled mixing strategy makes Generalist more robust to various attacks.

#### 4.2.2 Communication Frequency $c$

In Generalist, $c$ controls the communication frequency between the global learner and base learners. Therefore, for $c$, with the fixed mixing ratio strategy, we sweep over the frequency of communication from 1 to 15.

Results are shown in Figure 3, and we have the following observations. Intuitively, a larger $c$ means base learners communicate with the global learner less frequently to get the initialization, so they barely have the opportunity to move alternately towards two optimal solution manifolds. But specifically, the natural accuracy falls back down after reaching the peak while the robust accuracy in different adversarial settings roughly shows a trough. Such observation manifests that too much/little communication has a negative influence on standard accuracy but results in relatively higher robustness. It captures a tradeoff between natural and robust errors with respect to $c$.

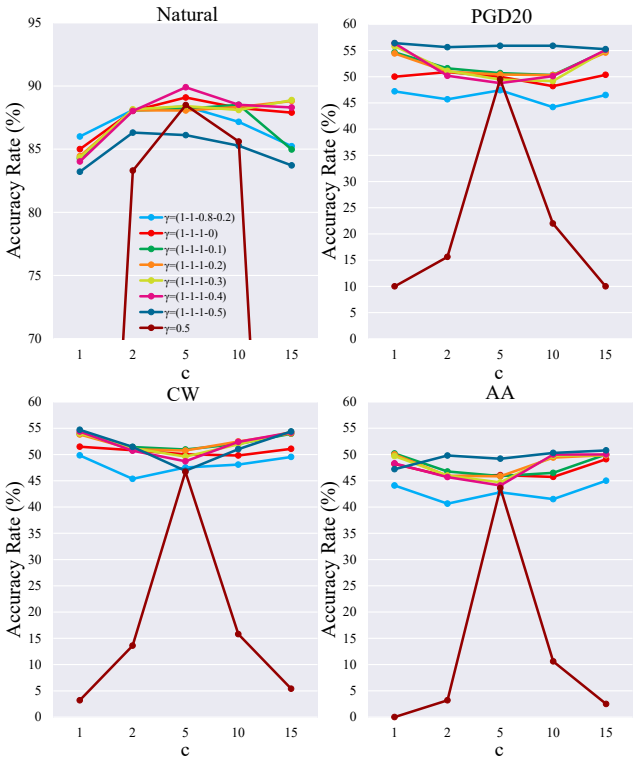
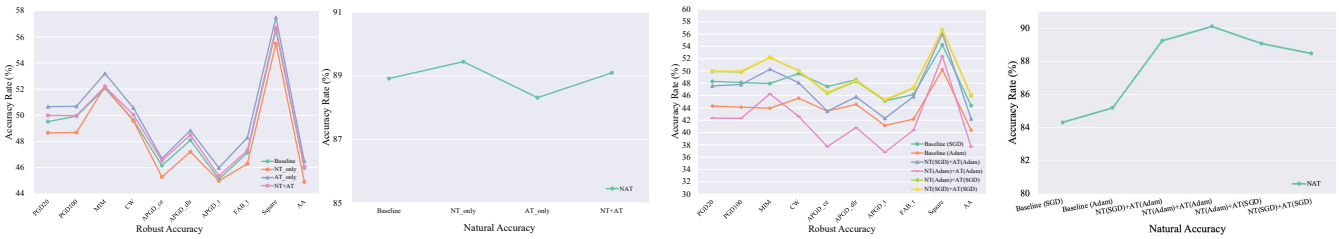



Figure 3. Generalist with different mixing ratio strategies and various values of frequency on CIFAR-10. We evaluate both natural accuracy and robustness against PGD20, C&W and AA attacks using ResNet-18.

#### 4.2.3 Parameter Selection

In practice, it is natural to select the mixing parameter $\gamma$ and the frequency of communication $c$ under a scenario without knowing the target model or dataset. We can find the best parameters on specific architecture and dataset, which is




(a) Weight Averaging (b) Different Optimizers

Figure 4. (a) We apply weight averaging to one of the base learners or both of them. Results demonstrate that using weight averaging through training can bring performance boost in its corresponding sub-task, and thus has an effect on predictions of the global learner. (b) Base learners of Generalist optimized by different optimizers. The optimal selection is using Adam for the natural classification task but maintaining SGD for the adversarial one.

then transferred to others, *i.e.*, choosing the best $\gamma$, $c$, and their strategies on one model/dataset and then used for other models/datasets, which still works well. Specifically, for the experimental results on MNIST/SVHN/CIFAR100 shown in Appendix A.2 and A.3, we just find the optimal parameters and updating strategies on CIFAR10 and apply similar $\gamma$ and $c$ on the target datasets and architectures without much fine-tuning. The performance is still very good.

### 4.3. Customized Policies for Individuals

As stressed above, one of the major advantages of Generalist in comparison with the standard joint training framework is that each base learner enables to customize the corresponding strategy for their own tasks freely rather than using the same strategy for all tasks. In this part, we investigate whether Generalist performs better when cooperating with diverse techniques.

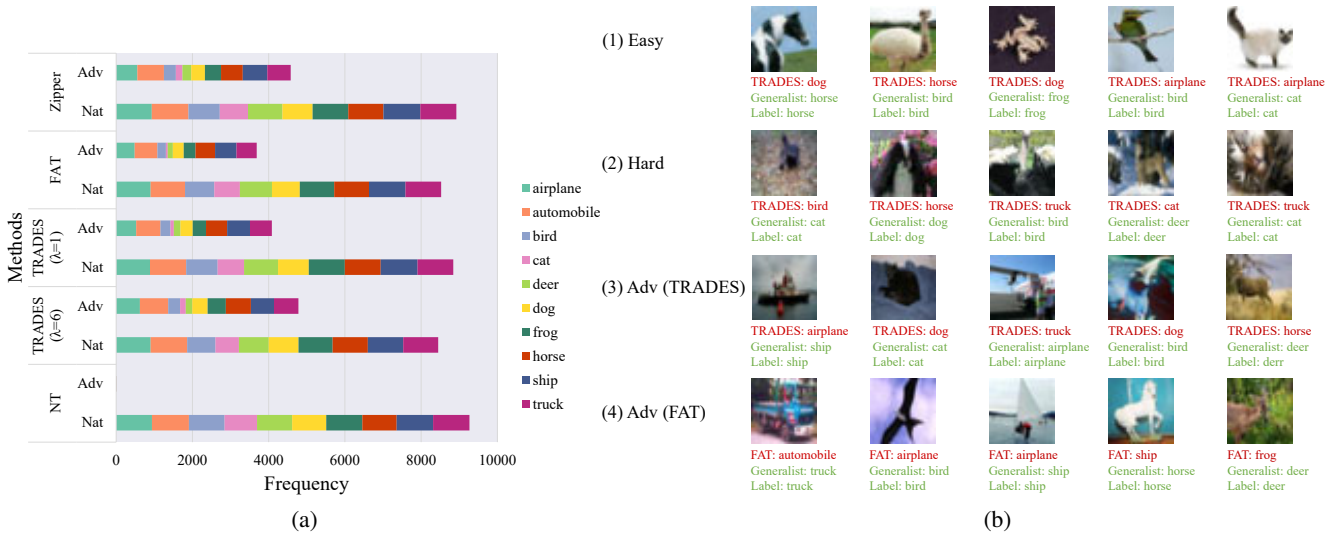
**Weight Averaging.** Recent works [15,27,35] have shown that weight averaging (WA) greatly improves both natural and robust generalization. The average parameters of all history model snapshot through the training process to build an ensemble model on the fly. However, such technique cannot benefit both accuracy and robustness in the joint training framework. Therefore, we introduce WA into base learners separately. Results are shown in Figure 4 (a). We employ WA in either NT (NT_only) or AT (AT_only) or both of them (NT+AT). Overall, the results confirm that the performance of the global learner can be further improved after both base learners exploit WA. But unfortunately, an obvious tradeoff happens if only one of the base learner is equipped with WA. For instance, the standard test accuracy of NT_only continues to increase at the expense of the drop in the ability to defend attacks. A likely reason is that WA implicitly controls the learning speed of base learners. Indeed, the base learner with WA becomes an expert much faster than the one without WA in its sub-task, meaning the fast one is not in accordance with the slow one. This result is important because it not only illustrates the potential of Generalist comes from its base learners but also identifies a key challenge of tradeoff for future improvement.

**Different Optimizers.** We also investigate the effect of optimizers designed for different tasks. We choose AT ($\beta = 1$) using SGD with momentum and Adam for piecewise learning rate schedule optimized by joint training as the baseline. The initial learning rate for Adam is 0.0001. We alternately apply these two optimizers in each subproblem. The comparison of the results is shown in Figure 4 (b). We can see that the gap of robust accuracy between models adversarially trained by Adam and the ones trained by SGD is significant. All three schemes equipped with Adam, namely NT (Adam)+AT (Adam), NT (SGD)+AT (Adam), and Baseline (Adam), perform worse than the ones using SGD when evaluated by adversarial attacks. But on the other hand, by comparing the results of Baseline (Adam) and NT (Adam)+AT (SGD), it confirms a proper optimization scheme with respect to data distribution can effectively benefit the corresponding performance without overlooking the other. That not only demonstrates the necessity of Generalist to decouple task-aware assignments from joint training but also indicates using Adam may not be the principal reason for robustness drop. It is just ill-suited for the outer and inner optimization in AT. Besides, though the best results still come from using SGD, the learning rate for different tasks can be customized which is not feasible in the joint framework, as shown in Appendix A.5.

### 4.4. Visualization

Considering the proposed method achieves impressive clean accuracy without a harsh drop in robustness, it is naturally to ask what improvements Generalist has secured in comparison with robust methods in detail. Thus, we further investigate the predictions that robust classifiers are prone to make. As shown in Figure 5, we provide two perspectives to analyze the differences that classifiers trained by different AT methods.

To broadly study the case, we perform experiments on NT, TRADES with different $\lambda$, FAT and Generalist, then plot the distribution of the correct predictions of all methods




(a) (b)

Figure 5. Analyses of predilections that different robust classifiers have on CIFAR-10 using ResNet-18. (a) Distribution of the correct predictions of different training methods for each class. We separate out results on natural examples from adversarial ones (AA). Note that results of 'NT Adv' does not appear in the figure just because they are literally zero. (b) Visualization of samples that other methods misclassify while Generalist makes right predictions.

for each class in Figure 5(a). As evident at first glance, we note that animals are more frequently misclassified, especially cats/dogs in the natural scenario and cats/deers in the adversarial scenario. In addition, the classifier trained by standard natural training does not always outperform the ones adversarially trained. Actually, they are equally skilled at most categories and the outcome is decided by specific categories (e.g. birds, cats and dogs). Generalist keeps pace with NT in the natural task, and meanwhile promotes the higher improvements in difficult items (e.g. cats and deers) against AA attack.

In Figure 5(b), we display specific samples in the testing dataset that are misclassified by robust classifiers (TRADES and FAT) but recognized by our proposed method, including both natural examples (the first two rows) and adversarial examples (the last two rows). Here, images shown in the first row are *easy* ones where the foreground objects stand out from the clear backgrounds, while *hard* samples are referred to those having confused objects with messy backgrounds. It is worth noting that TRADES delivers poor performances not only on hard examples with complex backgrounds or obscured objects but also on simple ones. For example, each image in the first row is typically plain and regular, however, TRADES fails in categorizing them into the right class. A plausible explanation for the issue is that TRADES lacks in a set of support measures specially devised for the natural classification task unlike Generalist does, highlighting design differentiation for sub-tasks is necessary.

Another interesting finding is that though both TRADES and FAT can build a robust classifier, they still rely on spurious background information and thus are easily deceived when encountering images with similar backgrounds but different objects. This phenomenon can be verified from the misclassification of the fourth and fifth images in the first row (taking white/blue backgrounds as evidence), and the fifth image in the fourth row (confused by the green background). But Generalist has the ability to sift the invariant feature of the foreground object while ignoring the background information spuriously correlated with the categories in both natural and adversarial settings. On the whole, Generalist demonstrates its strength to differentiate difficult samples close to the decision boundary and its potential to learn a background-invariant classifier.

## 5. Conclusion

In this paper, we propose a bi-expert framework named Generalist for improving the tradeoff issue between natural and robust generalization, which trains two base learners responsible for complementary fields and collects their parameters to construct a global learner. By decoupling from the joint training paradigm, each base learner can wield customized strategies based on data distribution. We provide theoretical analysis to justify the effectiveness of task-aware strategies and extensive experiments show that Generalist better mitigates the tradeoff of accuracy and robustness.

## Acknowledgement

Yisen Wang is partially supported by the National Key R&D Program of China (2022ZD0160304), the National Natural Science Foundation of China (62006153), Open Research Projects of Zhejiang Lab (No. 2022RC0AB05), and Huawei Technologies Inc.

## References


[1] Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? In *NeurIPS*, 2019. 2

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018. 1

[3] Yang Bai, Yan Feng, Yisen Wang, Tao Dai, Shu-Tao Xia, and Yong Jiang. Hilbert-based generative defense for adversarial examples. In *ICCV*, 2019. 1

[4] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. In *COLT*, 2001. 2

[5] Hakan Bilen and Andrea Vedaldi. Integrated perception with recurrent multi-task neural networks. In *NeurIPS*, 2016. 3

[6] Jose H. Blanchet and Karthyek R. A. Murthy. Quantifying distributional model risk via optimal transport. *Math. Oper. Res.*, 44(2):565–600, 2019. 2

[7] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *SP*, 2017. 5

[8] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C. Duchi, and Percy Liang. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019. 2

[9] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 1, 5

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 1

[11] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 5

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 3

[13] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 3, 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5

[15] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018. 7

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[17] Alex Lamb, Vikas Verma, Juho Kannala, and Yoshua Bengio. Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy. In *ACM AISec Workshop*, 2019. 5

[18] Saehyung Lee, Hyungyu Lee, and Sungroh Yoon. Adversarial vertex mixup: Toward better adversarially robust generalization. In *CVPR*, 2020. 2

[19] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogério Schmidt Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *CVPR*, 2017. 3

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 1, 3, 5

[21] Yichuan Mo, Dongxian Wu, Yifei Wang, Yiwen Guo, and Yisen Wang. When adversarial training meets vision transformers: Recipes from training to architecture. In *NeurIPS*, 2022. 1

[22] Amir Najafi, Shin-ichi Maeda, Masanori Koyama, and Takeru Miyato. Robustness to adversarial perturbations in learning from incomplete data. In *NeurIPS*, 2019. 2

[23] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. 3

[24] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *SP*, 2016. 1

[25] Yao Qin, Nicholas Frosst, Sara Sabour, Colin Raffel, Garrison W. Cottrell, and Geoffrey E. Hinton. Detecting and diagnosing adversarial images with class-conditional capsule reconstructions. In *ICLR*, 2020. 1

[26] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *ICML*, 2020. 2, 5

[27] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, and Timothy A. Mann. Fixing data augmentation to improve adversarial robustness. *CoRR*, abs/2103.01946, 2021. 7

[28] Hasim Sak, Andrew W. Senior, Kanishka Rao, and Françoise Beaufays. Fast and accurate recurrent neural network acoustic models for speech recognition. In *INTERSPEECH*, 2015. 1

[29] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1

[30] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019. 1

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[32] C. Villani. Topics in optimal transportation. 2003. 2

[33] Hongjun Wang, Guanbin Li, Xiaobai Liu, and Liang Lin. A hamiltonian monte carlo method for probabilistic adversarial attack and learning. *T-PAMI*, 2020. 1

[34] Hongjun Wang, Guangrun Wang, Ya Li, Dongyu Zhang, and Liang Lin. Transferable, controllable, and inconspicuous adversarial attacks on person re-identification with deep misranking. In *CVPR*, 2020. 1

[35] Hongjun Wang and Yisen Wang. Self-ensemble adversarial training for improved robustness. In *ICLR*, 2022. 7

[36] Yisen Wang, Xuejiao Deng, Songbai Pu, and Zhiheng Huang. Residual convolutional ctc networks for automatic speech recognition. *arXiv preprint arXiv:1702.07793*, 2017. 1

[37] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019. 1

[38] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020. 1

[39] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020. 1

[40] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *NDSS*, 2018. 1

[41] Yongxin Yang and Timothy M. Hospedales. Deep multi-task representation learning: A tensor factorisation approach. In *ICLR*, 2017. 3

[42] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. 5

[43] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 1, 3, 5

[44] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan S. Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020. 2, 5