Understanding and Mitigating the Tradeoff Between Robustness and Accuracy

Aditi Raghunathan [*1] **Sang Michael Xie** [*1] **Fanny Yang** [2] **John C. Duchi** [1] **Percy Liang** [1]

Abstract

Adversarial training augments the training set with perturbations to improve the robust error (over worst-case perturbations), but it often leads to an increase in the standard error (on unperturbed test inputs). Previous explanations for this tradeoff rely on the assumption that no predictor in the hypothesis class has low standard and robust error. In this work, we precisely characterize the effect of augmentation on the standard error in linear regression when the optimal linear predictor has zero standard and robust error. In particular, we show that the standard error could increase even when the augmented perturbations have noiseless observations from the optimal linear predictor. We then prove that the recently proposed robust self-training (RST) estimator improves robust error without sacrificing standard error for noiseless linear regression. Empirically, for neural networks, we find that RST with different adversarial training methods improves both standard and robust error for random and adversarial rotations and adversarial ℓ_∞ perturbations in CIFAR-10.

1. Introduction

Adversarial training methods (Goodfellow et al., 2015; Madry et al., 2017) attempt to improve the robustness of neural networks against adversarial examples (Szegedy et al., 2014) by augmenting the training set (on-the-fly) with perturbed examples that preserve the label but that fool the current model. While such methods decrease the *robust error*, the error on worst-case perturbed inputs, they have been observed to cause an undesirable increase in the *standard error*, the error on unperturbed inputs (Madry et al., 2018; Zhang et al., 2019; Tsipras et al., 2019).

[*]Equal contribution [1]Stanford University [2]ETH Zurich. Correspondence to: Aditi Raghunathan <aditir@stanford.edu>, Sang Michael Xie <xie@cs.stanford.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

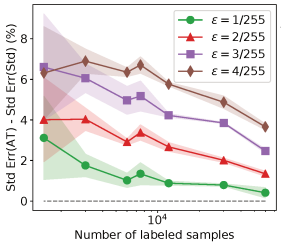


Figure 1. Gap between the standard error of adversarial training (Madry et al., 2018) with ℓ_∞ perturbations, and standard training. The gap decreases with increase in training set size, suggesting that the tradeoff between standard and robust error should disappear with infinite data.

Previous works attempt to explain the tradeoff between standard error and robust error in two settings: when no accurate classifier is *consistent* with the perturbed data (Tsipras et al., 2019; Zhang et al., 2019; Fawzi et al., 2018), and when the hypothesis class is not expressive enough to contain the true classifier (Nakkiran, 2019). In both cases, the tradeoff persists even with infinite data. However, adversarial perturbations in practice are typically defined to be imperceptible to humans (e.g. small ℓ_∞ perturbations in vision). Hence by definition, there exists a classifier (the human) that is both robust and accurate with no tradeoff in the infinite data limit. Furthermore, since deep neural networks are expressive enough to fit not only adversarial but also randomly labeled data perfectly (Zhang et al., 2017), the explanation of a restricted hypothesis class does not perfectly capture empirical observations either. Empirically on CIFAR-10, we find that the gap between the standard error of adversarial training and standard training decreases as we increase the labeled data size, thereby also suggesting the tradeoff could disappear with infinite data (See Figure 1).

In this work, we provide a different explanation for the tradeoff between standard and robust error that takes *generalization* from finite data into account. We first consider a linear model where the true linear function has zero standard and robust error. Adversarial training augments the original training set with *extra* data, consisting of samples (x_{ext}, y) where the perturbations x_{ext} are *consistent*, meaning that the conditional distribution stays constant $P_{\text{y}}(\cdot \mid x_{\text{ext}}) = P_{\text{y}}(\cdot \mid x)$. We show that even in this simple setting, the *augmented estimator*, i.e. the minimum norm interpolant of the augmented data (standard + extra data), could have a larger standard error than that of the *standard estimator*, which is the minimum norm interpolant of the standard data alone. We found this surprising given that adding consistent perturbations enforces the predictor to

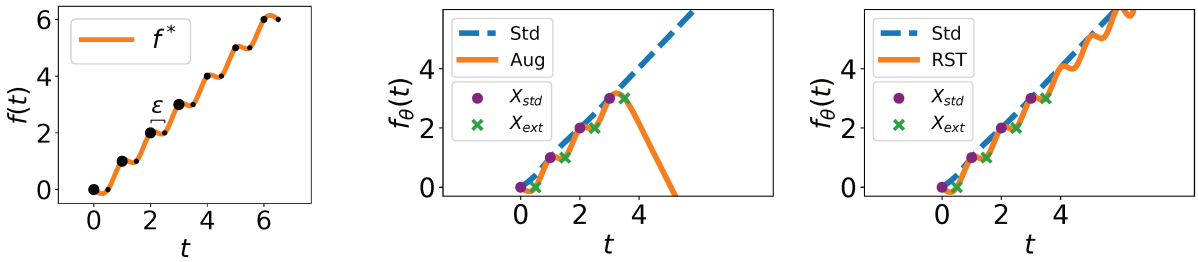


Figure 2. We consider function interpolation via cubic splines. **(Left)** The underlying distribution P_{x} denoted by sizes of the circles. The true function is a staircase. **(Middle)** With a small number of standard training samples (purple circles), an augmented estimator that fits local perturbations (green crosses) has a large error. In constrast, the standard estimator that does not fit perturbations is a simple straight line and has small error. **(Right)** Robust self-training (RST) regularizes the predictions of an augmented estimator towards the predictions of the standard estimator thereby obtaining both small error on test points and their perturbations.

satisfy invariances that the true model exhibits. One might think adding this information would only restrict the hypothesis class and thus enable better generalization, not worse.

We show that this tradeoff stems from overparameterization. If the *restricted* hypothesis class (by enforcing invariances) is still overparameterized, the inductive bias of the estimation procedure (e.g., the norm being minimized) plays a key role in determining the generalization of a model.

Figure 2 shows an illustrative example of this phenomenon with cubic smoothing splines. The predictor obtained via standard training (dashed blue) is a line that captures the global structure and obtains low error. Training on augmented data with locally consistent perturbations of the training data (crosses) restricts the hypothesis class by encouraging the predictor to fit the local structure of the high density points. Within this set, the cubic splines predictor (solid orange) minimizes the second derivative on the augmented data, compromising the global structure and performing badly on the tails (Figure 2(b)). More generally, as we characterize in Section 3, the tradeoff stems from the inductive bias of the minimum norm interpolant, which minimizes a fixed norm independent of the data, while the standard error depends on the geometry of the covariates.

Recent works (Carmon et al., 2019; Najafi et al., 2019; Uesato et al., 2019) introduced robust self-training (RST), a robust variant of self-training that overcomes the sample complexity barrier of learning a model with low robust error by leveraging extra unlabeled data. In this paper, our theoretical understanding of the tradeoff between standard and robust error in linear regression motivates RST as a method to improve robust error without sacrificing standard error. In Section 4.2, we prove that RST *eliminates* the tradeoff for linear regression—RST does not increase standard error compared to the standard estimator while simultaneously achieving the best possible robust error, matching the standard error (see Figure 2(c) for the effect of RST on the spline problem). Intuitively, RST regularizes the predictions of the robust estimator towards that of the standard estimator on the unlabeled data thereby eliminating the tradeoff.

As previous works only focus on the empirical evaluation of the gains in robustness via RST, we systematically evaluate the effect of RST on *both* the standard and robust error on CIFAR-10 when using unlabeled data from Tiny Images as sourced in Carmon et al. (2019). We expand upon empirical results in two ways. First, we study the effect of the labeled training set sizes and and find that the RST improves both robust and standard error over vanilla adversarial training across *all* sample sizes. RST offers maximum gains at smaller sample sizes where vanilla adversarial training increases the standard error the most. Second, we consider an additional family of perturbations over random and adversarial rotation/translations and find that RST offers gains in both robust and standard error.

2. Setup

We consider the problem of learning a mapping from an input $x \in \mathcal{X} \subseteq \mathbb{R}^d$ to a target $y \in \mathcal{Y}$. For our theoretical analysis, we focus on regression where $\mathcal{Y} = \mathbb{R}$ while our empirical studies consider general $\mathcal{Y}$. Let P_{xy} be the underlying distribution, P_{x} the marginal on the inputs and $P_{\mathrm{y}}(\cdot \mid x)$ the conditional distribution of the targets given inputs. Given n training pairs $(x_i, y_i) \sim P_{\mathrm{xy}}$, we use X_{std} to denote the measurement matrix $[x_1, x_2, \dots x_n]^\top \in \mathbb{R}^{n \times d}$ and y_{std} to denote the target vector $[y_1, y_2, \dots y_n]^\top \in \mathbb{R}^n$. Our goal is to learn a predictor $f_\theta : \mathcal{X} \to \mathcal{Y}$ that (i) has low standard error on inputs x and (ii) low robust error with respect to a set of perturbations $T(x)$. Formally, the error metrics for a predictor f_θ and a loss function ℓ are the *standard error*

$$L_{\mathrm{std}}(\theta) = \mathbb{E}_{P_{\mathrm{xy}}}[\ell(f_\theta(x), y)] \tag{1}$$

and the *robust error*

$$L_{\mathrm{rob}}(\theta) = \mathbb{E}_{P_{\mathrm{xy}}}[\max_{x_{\mathrm{ext}} \in T(x)} \ell(f_\theta(x_{\mathrm{ext}}), y)], \tag{2}$$

for *consistent perturbations* $T(x)$ that satisfy

$$P_{\mathsf{y}}(\cdot \mid x_{\text{ext}}) = P_{\mathsf{y}}(\cdot \mid x), \quad \forall x_{\text{ext}} \in T(x). \tag{3}$$

Such transformations may consist of small rotations, horizontal flips, brightness or contrast changes (Krizhevsky et al., 2012; Yaeger et al., 1996), or small ℓ_p perturbations in vision (Szegedy et al., 2014; Goodfellow et al., 2015) or word synonym replacements in NLP (Jia & Liang, 2017; Alzantot et al., 2018).

Noiseless linear regression. In section 3, we analyze noiseless linear regression on inputs x with targets $y = x^\top \theta^\star$ with true parameter $\theta^\star \in \mathbb{R}^k$.[1] For linear regression, ℓ is the squared loss which leads to the standard error (Equation 1) taking the form

$$L_{\text{std}}(\theta) = \mathbb{E}_{P_{\mathsf{x}}}[(x^\top\theta - x^\top\theta^\star)^2] = (\theta-\theta^\star)^\top \Sigma (\theta-\theta^\star), \tag{4}$$

where $\Sigma = \mathbb{E}_{P_{\mathsf{x}}}[xx^\top]$ is the population covariance.

Minimum norm estimators. In this work, we focus on interpolating estimators in highly overparameterized models, motivated by modern machine learning models that achieve near zero training loss (on both standard and extra data). Interpolating estimators for linear regression have been studied in many recent works such as (Ma et al., 2018; Belkin et al., 2018; Hastie et al., 2019; Liang & Rakhlin, 2018; Bartlett et al., 2019). We present our results for interpolating estimators with minimum Euclidean norm, but our analysis directly applies to more general Mahalanobis norms via suitable reparameterization (see Appendix A).

We consider robust training approaches that augment the *standard* training data $X_{\text{std}}, y_{\text{std}} \in \mathbb{R}^{n\times d} \times \mathbb{R}$ with some *extra* training data $X_{\text{ext}}, y_{\text{ext}} \in \mathbb{R}^{m\times d} \times \mathbb{R}$ where the rows of X_{ext} consist of vectors in the set $\{x_{\text{ext}} : x_{\text{ext}} \in T(x), x \in X_{\text{std}}\}$.[2] We call the standard data together with the extra data as *augmented* data. We compare the following min-norm estimators: (i) the *standard estimator* $\hat{\theta}_{\text{std}}$ interpolating $[X_{\text{std}}, y_{\text{std}}]$ and (ii) the *augmented estimator* $\hat{\theta}_{\text{aug}}$ interpolating $X = [X_{\text{std}}; X_{\text{ext}}], Y = [y_{\text{std}}; y_{\text{ext}}]$:

$$\hat{\theta}_{\text{std}} = \arg\min_\theta \left\{ \|\theta\|_2 : X_{\text{std}}\theta = y_{\text{std}} \right\}$$
$$\hat{\theta}_{\text{aug}} = \arg\min_\theta \left\{ \|\theta\|_2 : X_{\text{std}}\theta = y_{\text{std}}, X_{\text{ext}}\theta = y_{\text{ext}} \right\}. \tag{5}$$

Notation. For any vector $z \in \mathbb{R}^n$, we use z_i to denote the i^{th} coordinate of z.

[1] Our analysis extends naturally to arbitrary feature maps $\phi(x)$.

[2] In practice, X_{ext} is typically generated via iterative optimization such as in adversarial training (Madry et al., 2018), or by random sampling as in data augmentation (Krizhevsky et al., 2012; Yaeger et al., 1996).

3. Analysis in the Linear Regression Setting

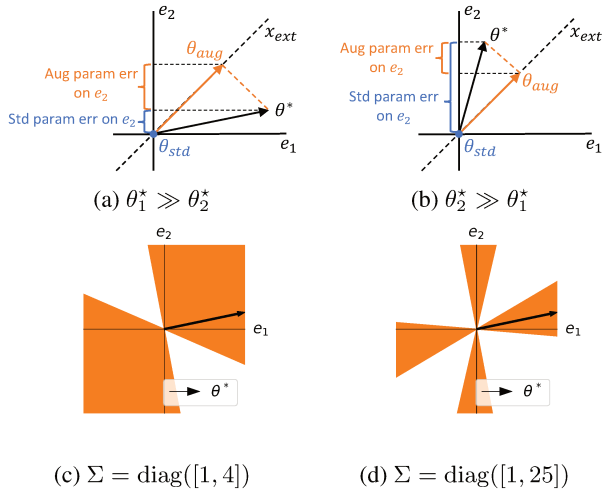
In this section, we compare the standard errors of the standard estimator and the augmented estimator in noiseless linear regression. We begin with a simple toy example that describes the intuition behind our results (Section 3.1) and provide a more complete characterization in Section 3.2. This section focuses only on the standard error of both estimators; we revisit the robust error together with the standard error in Section 4.

3.1. Simple Illustrative Problem

We consider a simple example in 3D where $\theta^\star \in \mathbb{R}^3$ is the true parameter. Let $e_1 = [1,0,0]; e_2 = [0,1,0]; e_3 = [0,0,1]$ denote the standard basis vectors in $\mathbb{R}^3$. Suppose we have one point in the standard training data $X_{\text{std}} = [0,0,1]$. By definition (5), $\hat{\theta}_{\text{std}}$ satisfies $X_{\text{std}}\hat{\theta}_{\text{std}} = y_{\text{std}}$ and hence $(\hat{\theta}_{\text{std}})_3 = \theta^\star_3$. However, $\hat{\theta}_{\text{std}}$ is unconstrained on the subspace spanned by e_1, e_2 (the nullspace Null(X_{std})). The min-norm objective chooses the solution with $(\hat{\theta}_{\text{std}})_1 = (\hat{\theta}_{\text{std}})_2 = 0$. Figure 3 visualizes the projection of various quantities on Null(X_{std}). For simplicity of presentation, we omit the projection operator in the figure. The projection of $\hat{\theta}_{\text{std}}$ onto Null(X_{std}) is the blue dot at the origin, and the parameter error $\theta^\star - \hat{\theta}_{\text{std}}$ is the projection of $\theta^\star$ onto Null(X_{std}).

Effect of augmentation on parameter error. Suppose we augment with an extra data point $X_{\text{ext}} = [1,1,0] = e_1 + e_2$ which lies in Null(X_{std}) (black dashed line in Figure 3). The augmented estimator $\hat{\theta}_{\text{aug}}$ still fits the standard data X_{std} and thus $(\hat{\theta}_{\text{aug}})_3 = \theta^\star_3 = (\hat{\theta}_{\text{std}})_3$. Due to fitting the extra data X_{ext}, $\hat{\theta}_{\text{aug}}$ (orange vector in Figure 3) must also satisfy an additional constraint $X_{\text{ext}}\hat{\theta}_{\text{aug}} = X_{\text{ext}}\theta^\star$. The crucial observation is that additional constraints along one direction ($e_1 + e_2$ in this case) could actually increase parameter error along other directions. For example, let's consider the direction e_2 in Figure 3. Note that fitting X_{ext} makes $\hat{\theta}_{\text{aug}}$ have a large component along e_2. Now if $\theta^\star_2$ is small (precisely, $\theta^\star_2 < \theta^\star_1/3$), $\hat{\theta}_{\text{aug}}$ has a larger parameter error along e_2 than $\hat{\theta}_{\text{std}}$, which was simply zero (Figure 3 (a)). Conversely, if the true component $\theta^\star_2$ is large enough (precisely, $\theta^\star_2 > \theta^\star_1/3$), the parameter error of $\hat{\theta}_{\text{aug}}$ along e_2 is smaller than that of $\hat{\theta}_{\text{std}}$.

Effect of parameter error on standard error. The contribution of different components of the parameter error to the standard error is scaled by the population covariance Σ (see Equation 4). For simplicity, let $\Sigma = \text{diag}([\lambda_1, \lambda_2, \lambda_3])$. In our example, the parameter error along e_3 is zero since both estimators interpolate the standard training point $X_{\text{std}} = e_1 = 3$. Then, the ratio between λ_1 and λ_2 determines which component of the parameter error contributes



(a) $\theta_1^\star \gg \theta_2^\star$ (b) $\theta_2^\star \gg \theta_1^\star$

(c) $\Sigma = \mathrm{diag}([1, 4])$ (d) $\Sigma = \mathrm{diag}([1, 25])$

Figure 3. Illustration of the 3-D example described in Sec. 3.1. **(a)-(b) Effect of augmentation on parameter error for different $\theta^\star$.** We show the projections of the standard estimator $\hat{\theta}_{\text{std}}$ (blue circle), augmented estimator $\hat{\theta}_{\text{aug}}$ (orange arrow), and true parameters $\theta^\star$ (black arrow) on Null(X_{std}), spanned by e_1 and e_2. For simplicity of presentation, we omit the projection operator in the figure labels. Depending on $\theta^\star$, the parameter error of $\hat{\theta}_{\text{aug}}$ along e_2 could be larger or smaller than the parameter error of $\hat{\theta}_{\text{std}}$ along e_2. **(c)–(d) Dependence of space of safe augmentations on Σ.** Visualization of the space of extra data points x_{ext} (orange), that do not cause an increase in the standard error for the illustrated $\theta^\star$ (black vector), as result of Theorem 1.

more to the standard error.

When is $L_{\text{std}}(\hat{\theta}_{\text{aug}}) > L_{\text{std}}(\hat{\theta}_{\text{std}})$? Putting the two effects together, we see that when $\theta_2^\star$ is small as in Fig 3(a), $\hat{\theta}_{\text{aug}}$ has larger parameter error than $\hat{\theta}_{\text{std}}$ in the direction e_2. If $\lambda_2 \gg \lambda_1$, error in e_2 is weighted much more heavily in the standard error and consequently $\hat{\theta}_{\text{aug}}$ would have a larger standard error. Precisely, we have

$$L_{\text{std}}(\hat{\theta}_{\text{aug}}) > L_{\text{std}}(\hat{\theta}_{\text{std}}) \iff \lambda_2(\theta_1^\star - 3\theta_2^\star) > \lambda_1(3\theta_1^\star - \theta_2^\star).$$

We present a formal characterization of this tradeoff in general in the next section.

3.2. General Characterizations

In this section, we precisely characterize when the augmented estimator $\hat{\theta}_{\text{aug}}$ that fits extra training data points X_{ext} in addition to the standard points X_{std} has higher standard error than the standard estimator $\hat{\theta}_{\text{std}}$ that only fits X_{std}. In particular, this enables us to understand when there is a "tradeoff" where the augmented estimator $\hat{\theta}_{\text{aug}}$ has lower robust error than $\hat{\theta}_{\text{std}}$ by virtue of fitting perturbations, but has higher standard error. In Section 3.1, we illustrated how the parameter error of $\hat{\theta}_{\text{aug}}$ could be larger than $\hat{\theta}_{\text{std}}$ in some directions, and if these directions are weighted heavily in the population covariance Σ, the standard error of $\hat{\theta}_{\text{aug}}$ would be larger.

Formally, let us define the parameter errors $\Delta_{\text{std}} \overset{\text{def}}{=} \hat{\theta}_{\text{std}} - \theta^\star$ and $\Delta_{\text{aug}} \overset{\text{def}}{=} \hat{\theta}_{\text{aug}} - \theta^\star$. Recall that the standard errors are

$$L_{\text{std}}(\hat{\theta}_{\text{std}}) = \Delta_{\text{std}}^\top \Sigma \Delta_{\text{std}}, \quad L_{\text{std}}(\hat{\theta}_{\text{aug}}) = \Delta_{\text{aug}}^\top \Sigma \Delta_{\text{aug}}, \tag{6}$$

where Σ is the population covariance of the underlying inputs drawn from P_{x}.

To characterize the effect of the inductive bias of minimum norm interpolation on the standard errors, we define the following projection operators: $\Pi_{\text{std}}^\perp$, the projection matrix onto Null(X_{std}) and $\Pi_{\text{aug}}^\perp$, the projection matrix onto Null$([X_{\text{ext}}; X_{\text{std}}])$ (see formal definition in Appendix B). Since $\hat{\theta}_{\text{aug}}$ and $\hat{\theta}_{\text{std}}$ are minimum norm interpolants, $\Pi_{\text{std}}^\perp \hat{\theta}_{\text{std}} = 0$ and $\Pi_{\text{aug}}^\perp \hat{\theta}_{\text{aug}} = 0$. Further, in noiseless linear regression, $\hat{\theta}_{\text{std}}$ and $\hat{\theta}_{\text{aug}}$ have no error in the span of X_{std} and $[X_{\text{std}}; X_{\text{ext}}]$ respectively. Hence,

$$\Delta_{\text{std}} = \Pi_{\text{std}}^\perp \theta^\star, \quad \Delta_{\text{aug}} = \Pi_{\text{aug}}^\perp \theta^\star. \tag{7}$$

Our main result relies on the key observation that for any vector u, $\Pi_{\text{std}}^\perp u$ can be decomposed into a sum of two orthogonal components v and w such that $\Pi_{\text{std}}^\perp u = v + w$ with $w = \Pi_{\text{aug}}^\perp u$ and $v = \Pi_{\text{std}}^\perp \Pi_{\text{aug}} u$. This is because Null$([X_{\text{std}}; X_{\text{ext}}]) \subseteq$ Null(X_{std}) and thus $\Pi_{\text{std}}^\perp \Pi_{\text{aug}}^\perp = \Pi_{\text{aug}}^\perp$. Now setting $u = \theta^\star$ and using the error expressions in Equation 6 and Equation 7 gives a precise characterization of the difference in the standard errors of $\hat{\theta}_{\text{std}}$ and $\hat{\theta}_{\text{aug}}$.

Theorem 1. *The difference in the standard errors of the standard estimator $\hat{\theta}_{std}$ and augmented estimator $\hat{\theta}_{aug}$ can be written as follows.*

$$L_{std}(\hat{\theta}_{std}) - L_{std}(\hat{\theta}_{aug}) = v^\top \Sigma v + 2w^\top \Sigma v, \tag{8}$$

where $v = \Pi_{std}^\perp \Pi_{aug} \theta^\star$ and $w = \Pi_{aug}^\perp \theta^\star$.

The proof of Theorem 1 is in Appendix B.3. The increase in standard error of the augmented estimator can be understood in terms of the vectors w and v defined in Theorem 1. The first term $v^\top \Sigma v$ is always positive, and corresponds to the decrease in the standard error of the augmented estimator $\hat{\theta}_{\text{aug}}$ by virtue of fitting extra training points in some directions. However, the second term $2w^\top \Sigma v$ can be negative and intuitively measures the cost of a possible increase in the parameter error along other directions (similar to the increase along e_2 in the simple setting of Figure 3(a)). When the cost outweighs the benefit, the standard error of $\hat{\theta}_{\text{aug}}$ is larger. Note that both the cost and benefit is determined by Σ which governs how the parameter error affects the standard error.

We can use the above expression (Theorem 1) for the difference in standard errors of $\hat{\theta}_{\text{aug}}$ and $\hat{\theta}_{\text{std}}$ to characterize

different "safe" conditions under which augmentation with extra data does not increase the standard error. See Appendix B.7 for a proof.

Corollary 1. *The following conditions are sufficient for $L_{std}(\hat{\theta}_{aug}) \leq L_{std}(\hat{\theta}_{std})$, i.e. the standard error does not increase when fitting augmented data.*

1. *The population covariance Σ is identity.*
2. *The augmented data $[X_{std}; X_{ext}]$ spans the entire space, or equivalently $\Pi^{\perp}_{aug} = 0$.*
3. *The extra data $x_{ext} \in \mathbb{R}^d$ is a single point such that x_{ext} is an eigenvector of Σ.*

Matching inductive bias. We would like to draw special attention to the first condition. When $\Sigma = I$, notice that the norm that governs the standard error (Equation 6) matches the norm that is minimized by the interpolants (Equation 5). Intuitively, the estimators have the "right" inductive bias; under this condition, the augmented estimator $\hat{\theta}_{\text{aug}}$ does not have higher standard error. In other words, the observed increase in the standard error of $\hat{\theta}_{\text{aug}}$ can be attributed to the "wrong" inductive bias. In Section 4, we will use this understanding to propose a method of robust training which does not increase standard error over standard training.

Safe extra points. We use Theorem 1 to plot the safe extra points $x_{\text{ext}} \in \mathbb{R}^d$ that do not lead to an increase in standard error for any $\theta^\star$ in the simple 3D setting described in Section 3.1 for two different Σ (Figure 3 (c), (d)). The safe points lie in cones which contain the eigenvectors of Σ (as expected from Corollary 1). The width and alignment of the cones depends on the alignment between $\theta^\star$ and the eigenvectors of Σ. As the eigenvalues of Σ become less skewed, the space of safe points expands, eventually covering the entire space when $\Sigma = I$ (see Corollary 1).

Local versus global structure. We now tie our analysis back to the cubic splines interpolation problem from Figure 2. The inputs can be appropriately rotated and scaled such that the cubic spline interpolant is the minimum Euclidean norm interpolant (as in Equation 5). Under this transformation, the different eigenvectors of the nullspace of the training data Null(X_{std}) represent the "local" high frequency components with small eigenvalues or "global" low frequency components with large eigenvalues (see Figure 4). An augmentation that encourages the fitting local components in Null(X_{std}) could potentially increase the error along other global components (like the increase in error along e_2 in Figure 3(a)). Such an increase, coupled with the fact that global components have larger eigenvalue in Σ, results in the standard error of $\hat{\theta}_{\text{aug}}$ being larger than that of $\hat{\theta}_{\text{std}}$. See Figure 8 and Appendix C.3.1 for more details.

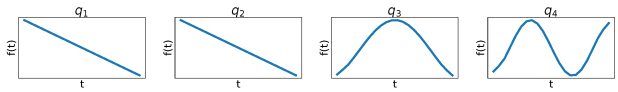


Figure 4. Top 4 eigenvectors of Σ in the splines problem (from Figure 2), representing wave functions in the input space. The "global" eigenfunctions, varying less over the domain, correspond to larger eigenvalues, making errors in global dimensions costly in terms of test error.

This is similar to the recent observation that adversarial training with ℓ_∞ perturbations encourages neural networks to fit the high frequency components of the signal while compromising on the low-frequency components (Yin et al., 2019).

Model complexity. Finally, we relate the magnitude of increase in standard error of the augmented estimator to the complexity of the true model.

Proposition 1. *For a given X_{std}, X_{ext}, Σ,*

$$L_{std}(\hat{\theta}_{aug}) - L_{std}(\hat{\theta}_{std}) > c \implies \|\theta^\star\|_2^2 - \|\hat{\theta}_{std}\|_2^2 > \gamma c$$

for some scalar $\gamma > 0$ that depends on X_{std}, X_{ext}, Σ.

In other words, for a large increase in standard error upon augmentation, the true parameter $\theta^\star$ needs to be sufficiently more complex (in the ℓ_2 norm) than the standard estimator $\hat{\theta}_{\text{std}}$. For example, the construction of the cubic splines interpolation problem relies on the underlying function (staircase) being more complex with additional local structure than the standard estimator—a linear function that fits most points and can be learned with few samples. Proposition 1 states that this requirement holds more generally. The proof of Proposition 1 appears in Appendix B.5. A similar intuition can be used to construct an example where augmentation can increase standard error for minimum ℓ_1-norm interpolants when $\theta^\star$ is dense (Appendix G).

4. Robust Self-Training

We now use insights from Section 3 to construct estimators with low robust error without increasing the standard error. While Section 3 characterized the effect of adding extra data X_{ext} in general, in this section we consider robust training which augments the dataset with extra data X_{ext} that are consistent perturbations of the standard training data X_{std}.

Since the standard estimator has small standard error, a natural strategy to mitigate the tradeoff is to regularize the augmented estimator to be closer to the standard estimator. The choice of distance between the estimators we regularize is very important. Recall from Section 3.1 that the population covariance Σ determines how the parameter error affects the standard error. This suggests using a regularizer that incorporates information about Σ.

We first revisit the recently proposed robust self-training (RST) (Carmon et al., 2019; Najafi et al., 2019; Uesato et al., 2019) that incorporates additional unlabeled data via pseudo-labels from a standard estimator. Previous work only focused on the effectiveness of RST in improving the robust error. In Section 4.2, we prove that in linear regression, RST eliminates the tradeoff between standard and robust error (Theorem 2). The proof hinges on the connection between RST and the idea of regularizing towards the standard estimator discussed above. In particular, we show that the RST objective can be rewritten as minimizing a suitable Σ-induced distance to the standard estimator.

In Section 4.3, we expand upon previous empirical RST results for CIFAR-10 across various training set sizes and perturbations (rotations/translations in addition to ℓ_∞). We observe that across all settings, RST substantially improves the standard error while also improving the robust error over the vanilla supervised robust training counterparts.

4.1. General Formulation of RST

We first describe the general two-step robust self-training (RST) procedure (Carmon et al., 2019; Uesato et al., 2019) for a parameteric model f_θ:

1. Perform standard training on labeled data $\{(x_i, y_i)\}_{i=1}^n$ to obtain $\hat{\theta}_{\text{std}} = \arg\min_\theta \sum_{i=1}^n \ell\big(f_\theta(x_i), y_i\big)$.

2. Perform robust training on both the labeled data and unlabeled inputs $\{\tilde{x}_i\}_{i=1}^m$ with *pseudo-labels* $\tilde{y}_i = f_{\hat{\theta}_{\text{std}}}(\tilde{x}_i)$ generated from the standard estimator $\hat{\theta}_{\text{std}}$.

The second stage typically involves a combination of the standard loss ℓ and a robust loss ℓ_{rob}. The robust loss encourages invariance of the model over perturbations $T(x)$, and is generally defined as

$$\ell_{\text{rob}}(f_\theta(x_i), y_i) = \max_{x_{\text{adv}} \in T(x_i)} \ell(f_\theta(x_{\text{adv}}), y_i). \tag{9}$$

It is convenient to summarize the robust self-training estimator $\hat{\theta}_{\text{rst}}$ as the minimizer of a weighted combination of four separate losses as follows. We define the losses on the labeled dataset $\{(x_i, y_i)\}_{i=1}^n$ as

$$\hat{L}_{\text{std-lab}}(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(f_\theta(x_i), y_i),$$
$$\hat{L}_{\text{rob-lab}}(\theta) = \frac{1}{n}\sum_{i=1}^n \ell_{\text{rob}}(f_\theta(x_i), y_i).$$

	Standard	Robust
Labeled	$(y - x^\top\theta)^2$ Noiseless targets	$\max_{x_{\text{adv}} \in T(x)} (x^\top\theta - x_{\text{adv}}^\top\theta)^2$ Consistent perturbations
Unlabeled	$(\tilde{y} - \tilde{x}^\top\theta)^2$ Imperfect pseudo-labels	$\max_{\tilde{x}_{\text{adv}} \in T(\tilde{x})} (\tilde{x}^\top\theta - \tilde{x}_{\text{adv}}^\top\theta)^2$ Consistent perturbations

Figure 5. Illustration shows the four components of the RST loss (Equation (10)) in the special case of linear regression (Eq. (11)). Green cells contain hard constraints where the optimal $\theta^\star$ obtains zero loss. The orange cell contains the soft constraint that is minimized while satisfying hard constraints to obtain the final linear RST estimator.

The losses on the unlabeled samples $\{\tilde{x}_i\}_{i=1}^m$ which are psuedo-labeled by the standard estimator are

$$\hat{L}_{\text{std-unlab}}(\theta; \hat{\theta}_{\text{std}}) = \frac{1}{m}\sum_{i=1}^m \ell(f_\theta(\tilde{x}_i), f_{\hat{\theta}_{\text{std}}}(\tilde{x}_i)),$$
$$\hat{L}_{\text{rob-unlab}}(\theta; \hat{\theta}_{\text{std}}) = \frac{1}{m}\sum_{i=1}^m \ell_{\text{rob}}(f_\theta(\tilde{x}_i), f_{\hat{\theta}_{\text{std}}}(\tilde{x}_i)).$$

Putting it all together, we have

$$\hat{\theta}_{\text{rst}} := \arg\min_\theta \Big(\alpha\hat{L}_{\text{std-lab}}(\theta) + \beta\hat{L}_{\text{rob-lab}}(\theta) + \gamma\hat{L}_{\text{std-unlab}}(\theta; \hat{\theta}_{\text{std}}) + \lambda\hat{L}_{\text{rob-unlab}}(\theta; \hat{\theta}_{\text{std}})\Big), \tag{10}$$

for fixed scalars $\alpha, \beta, \gamma, \lambda \geq 0$.

4.2. Robust Self-Training for Linear Regression

We now return to the noiseless linear regression as described in Section 2 and specialize the general RST estimator described in Equation (10) to this setting. We prove that RST eliminates the decrease in standard error in this setting while achieving low robust error by showing that RST appropriately regularizes the augmented estimator towards the standard estimator.

Our theoretical results hold for RST procedures where the pseudo-labels can be generated from any interpolating estimator $\theta_{\text{int-std}}$ satisfying $X_{\text{std}}\theta_{\text{int-std}} = y_{\text{std}}$. This includes but is not restricted to the mininum-norm standard estimator $\hat{\theta}_{\text{std}}$ defined in (5). We use the squared loss as the loss function ℓ. For consistent perturbations $T(\cdot)$, we analyze the following RST estimator for linear regression

$$\hat{\theta}_{\text{rst}} = \arg\min_\theta \{L_{\text{std-unlab}}(\theta; \theta_{\text{int-std}}) : L_{\text{rob-unlab}}(\theta) = 0, \hat{L}_{\text{std-lab}}(\theta) = 0, \hat{L}_{\text{rob-lab}}(\theta) = 0\}. \tag{11}$$

Figure 5 shows the four losses of RST in this special case of linear regression.

Obtaining this specialized estimator from the general RST estimator in Equation (10) involves the following steps. First, for convenience of analysis, we assume access to the population covariance Σ via infinite unlabeled data and thus replace the finite sample losses on the unlabeled data $\hat{L}_{\text{std-unlab}}(\theta), \hat{L}_{\text{rob-unlab}}(\theta)$ by their population losses $L_{\text{std-unlab}}(\theta), L_{\text{rob-unlab}}(\theta)$. Second, the general RST objective minimizes some weighted combination of four losses. When specializing to the case of noiseless linear regression, since $\hat{L}_{\text{std, lab}}(\theta^\star) = 0$, rather than minimizing $\alpha\hat{L}_{\text{std-lab}}(\theta^\star)$, we set the coefficients on the losses such that the estimator satisfies a hard constraint $\hat{L}_{\text{std-lab}}(\theta^\star) = 0$. This constraint which enforces interpolation on the labeled dataset $y_i = x_i^\top\theta \;\forall i = 1, \dots n$ allows us to rewrite the robust loss (Equation 9) on the labeled examples equivalently as a self-consistency loss defined independent of labels.

$$\hat{L}_{\text{rob-lab}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \max_{x_{\text{adv}} \in T(x)} (x_i^\top\theta - x_{\text{adv}}^\top\theta)^2.$$

Since $\theta^\star$ is invariant on perturbations $T(x)$ by definition, we have $\hat{L}_{\text{rob-lab}}(\theta^\star) = 0$ and thus we introduce a constraint $\hat{L}_{\text{rob-lab}}(\theta) = 0$ in the estimator.

For the losses on the unlabeled data, since the pseudo-labels are not perfect, we minimize $L_{\text{std-unlab}}$ in the objective instead of enforcing a hard constraint on $L_{\text{std-unlab}}$. However, similarly to the robust loss on labeled data, we can reformulate the robust loss on unlabeled samples $L_{\text{rob-unlab}}$ as a self-consistency loss that does not use pseudo-labels. By definition, $L_{\text{rob-unlab}}(\theta^\star) = 0$ and thus we enforce $L_{\text{rob-unlab}}(\theta) = 0$ in the specialized estimator.

We now study the standard and robust error of the linear regression RST estimator defined above in Equation (11).

Theorem 2. *Assume the noiseless linear model $y = x^\top\theta^\star$. Let $\theta_{\text{int-std}}$ be an arbitrary interpolant of the standard data, i.e. $X_{std}\theta_{\text{int-std}} = y_{std}$. Then*

$$L_{std}(\hat{\theta}_{rst}) \le L_{std}(\theta_{\text{int-std}}).$$

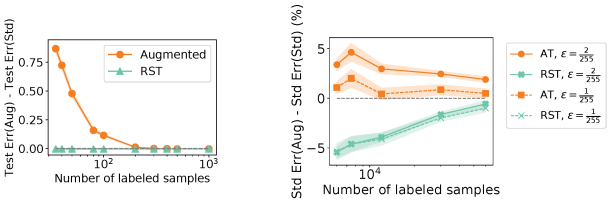
Simultaneously, $L_{rob}(\hat{\theta}_{rst}) = L_{std}(\hat{\theta}_{rst})$.

See Appendix D for a full proof.

The crux of the proof is that the optimization objective of RST is an inductive bias that regularizes the estimator to be close to the standard estimator, weighing directions by their contribution to the standard error via Σ. To see this, we rewrite

$$\begin{aligned} L_{\text{std-unlab}}(\theta; \theta_{\text{int-std}}) &= \mathbb{E}_{P_x}[(\tilde{x}^\top\theta_{\text{int-std}} - \tilde{x}^\top\theta)^2] \\ &= (\theta_{\text{int-std}} - \theta)^\top\Sigma(\theta_{\text{int-std}} - \theta). \end{aligned}$$

By incorporating an appropriate Σ-induced regularizer while satisfying constraints on the robust losses, RST ensures that the standard error of the estimator never exceeds the standard error of $\hat{\theta}_{\text{std}}$. The robust error of any estimator is lower bounded by its standard error, and this gap can be arbitrarily large for the standard estimator. However, the robust error of the RST estimator matches the lower bound of its standard error which in turn is bounded by the standard error of the standard estimator and hence is small. To provide some graphical intuition for the result, see Figure 2 that visualizes the RST estimator on the cubic splines interpolation problem that exemplifies the increase in standard error upon augmentation. RST captures the global structure and obtains low standard error by matching $\hat{\theta}_{\text{std}}$ (straight line) on unlabeled inputs. Simultaneously, RST enforces invariance on local transformations on both labeled and unlabeled inputs, and obtains low robust error by capturing the local structure across the domain.



(a) Spline Staircase (b) CIFAR-10 (AT)

Figure 6. Effect of data augmentation on test error as we vary the number of training samples. **(a)-(b)** We plot the difference in errors of the augmented estimator and standard estimator. In both the spline staircase simulations and data augmentation with adversarial ℓ_∞ perturbations via adversarial training (AT) on CIFAR-10, the increase in test error decreases as the training sample size increases. In **(b)**, robust self-training (RST+AT) not only mitigates the increase in test error from AT but even improves test error beyond that of the standard estimator.

Implementation of linear RST. The constraint on the standard loss on labeled data simply corresponds to interpolation on the standard labeled data. The constraints on the robust self-consistency losses involve a maximization over a set of transformations. In the case of linear regression, such constraints can be equivalently represented by a set of at most d linear constraints, where d is the dimension of the covariates. Further, with this finite set of constraints, we only require access to the covariance Σ in order to constrain the population robust loss. Appendix D gives a practical iterative algorithm that computes the RST estimator for linear regression reminiscent of adversarial training in the semi-supervised setting.

4.3. Empirical Evaluation of RST

Carmon et al. (2019) empirically evaluate RST with a focus on studying gains in the robust error. In this work, we focus on *both* the standard and robust error and expand upon results from previous work. Carmon et al. (2019) used

Method	Robust Test Acc.	Standard Test Acc.	
Standard Training	0.8%	95.2%	Vanilla Supervised
PG-AT (Madry et al., 2018)	45.8%	87.3%	
TRADES (Zhang et al., 2019)	55.4%	84.0%	
Standard Self-Training	0.3%	96.4%	Semisupervised with same unlabeled data
Robust Consistency Training (Carmon et al., 2019)	56.5%	83.2%	
RST + PG-AT (this paper)	**58.5%**	**91.8%**	
RST + TRADES (this paper)	**63.1%**	**89.7%**	
(Carmon et al., 2019)			
Interpolated AT (Lamb et al., 2019)[3]	45.1%	93.6%	Modified supervised
Neural Arch. Search (Cubuk et al., 2017)	50.1%	93.2%	

Method	Robust Test Acc.	Standard Test Acc.	
Standard Training	0.2%	94.6%	Vanilla Supervised
Worst-of-10	73.9%	95.0%	
Random	67.7%	95.1%	
RST + Worst-of-10 (this paper)	**75.1%**	**95.8%**	Semisupervised
RST + Random (this paper)	**70.9%**	**95.8%**	
Worst-of-10 (Engstrom et al., 2019)[4]	69.2%	91.3%	Existing baselines (smaller model)
Random (Yang et al., 2019)[5]	58.3%	91.8%	

Table 1. Performance of robust self-training (RST) applied to different perturbations and adversarial training algorithms. **(Left)** CIFAR-10 standard and robust test accuracy against ℓ_∞ perturbations of size $\epsilon = 8/255$. All methods use $\epsilon = 8/255$ while training and use the WRN-28-10 model. Robust accuracies are against a PG based attack with 20 steps. **(Right)** CIFAR-10 standard and robust test accuracy against a grid attack of rotations up to 30 degrees and translations up to $\sim 10\%$ of the image size, following (Engstrom et al., 2019). All adversarial and random methods use the same parameters during training and use the WRN-40-2 model. For both tables, shaded rows make use of 500K unlabeled images from 80M Tiny Images sourced in (Carmon et al., 2019). RST improves *both* the standard and robust accuracy over the vanilla counterparts for different algorithms (AT and TRADES) and different perturbations (ℓ_∞ and rotation/translations).

TRADES (Zhang et al., 2019) as the robust loss in the general RST formulation (10); we additionally evaluate RST with Projected Gradient Adversarial Training (AT) (Madry et al., 2018) as the robust loss. Carmon et al. (2019) considered ℓ_∞ and ℓ_2 perturbations. We study rotations and translations in addition to ℓ_∞ perturbations, and also study the effect of labeled training set size on standard and robust error. Table 1 presents the main results. More experiment details appear in Appendix D.3[6].

Both RST+AT and RST+TRADES have lower robust and standard error than their supervised counterparts AT and TRADES across all perturbation types. This mirrors the theoretical analysis of RST in linear regression (Theorem 2) where the RST estimator has small robust error while provably not sacrificing standard error, and never obtaining larger standard error than the standard estimator.

Effect of labeled sample size. Recall that our work motivates studying the tradeoff between robust and standard error while taking *generalization* from finite data into account. We showed that the gap in the standard error of a standard estimator and that of a robust estimator is large for small training set sizes and decreases as the labeled dataset is larger (Figure 1). We now study the effect of RST as we vary the training set size in Figure 6. We find that RST+AT has *lower* standard error than standard training across all sample sizes for small ϵ, while simultaneously achieving lower robust error than AT (see Appendix E.2.1). In the small data regime where vanilla adversarial training hurts the standard error the most, we find that RST+AT gives about 3x more absolute improvement than in the large data regime. We note that this set of experiments are complementary to the experiments in (Schmidt et al., 2018) which study the effect of the training set size only on robust error.

Effect on transformations that do not hurt standard error. We also test the effect of RST on perturbations where robust training slightly improves standard error rather than hurting it. Since RST regularizes towards the standard estimator, one might suspect that the improvements from robust training disappear with RST. In particular, we consider spatial transformations $T(x)$ that consist of simultaneous rotations and translations. We use two common forms of robust training for spatial perturbations, where we approximately maximize over $T(x)$ with either adversarial (worst-of-10) or random augmentations (Yang et al., 2019; Engstrom et al., 2019). Table 1 (right) presents the results. In the regime where vanilla robust training does not hurt standard error, RST in fact further improves the standard error by almost 1% and the robust error by 2-3% over the standard and robust estimators for both forms of robust training. Thus in settings where vanilla robust training improves standard error, RST seems to further amplify the gains while in settings where vanilla robust training hurts standard error, RST mitigates the harmful effect.

Comparison to other semi-supervised approaches. The RST estimator minimizes both a robust loss and a stan-

[6]Code for our experiments are available here.

dard loss on the unlabeled data with pseudo-labels (bottom row, Figure 5). Both of these losses are necessary to simultaneously the standard and robust error over vanilla supervised robust training. Standard self-training, which only uses standard loss on unlabeled data, has very high robust error ($\approx 100\%$). Similarly, Robust Consistency Training, an extension of Virtual Adversarial Training (Miyato et al., 2018) that only minimizes a robust self-consistency loss on unlabeled data, marginally improves the robust error but actually *hurts* standard error (Table 1).

5. Related Work

Existence of a tradeoff. Several works have attempted to explain the tradeoff between standard and robust error by studying simple models. These explanations are based on an *inherent tradeoff* that persists even in the infinite data limit. In Tsipras et al. (2019); Zhang et al. (2019); Fawzi et al. (2018), standard and robust error are fundamentally at odds, meaning no classifier is both accurate and robust. In Nakkiran (2019), the tradeoff is due to the hypothesis class not being expressive enough to contain an accurate and robust classifier even if it exists. In contrast, we explain the tradeoff in a more realistic setting with label-preserving consistent perturbations (like imperceptible ℓ_∞ perturbations or small rotations) in a well-specified setting (to mirror expressive neural networks) where there is no tradeoff with infinite data. In particular, our work takes into account generalization from finite data to explain the tradeoff.

In concurrent and independent work, Chen et al. (2020) also study the effect of dataset size on the tradeoff. They prove that in a "strong adversary" regime, there is a tradeoff even with infinite data, as the perturbations are large enough to change the ground truth target. They also identify a "weak adversary" regime (smaller perturbations) where the gap in standard error between robust and standard estimators first increases and then decreases, with no tradeoff in the infinite data limit. Similar to our work, this provides an example of a tradeoff due to generalization from finite data. However, their experimental validation of the tradeoff trends is restricted to simulated settings and they do not study how to mitigate the tradeoff.

Mitigating the tradeoff. To the best of our knowledge, ours is the first work that theoretically studies how to mitigate the tradeoff between standard and robust error. While robust self-training (RST) was proposed in recent works (Carmon et al., 2019; Najafi et al., 2019; Uesato et al., 2019) as a way to improve *robust error*, we prove that RST eliminates the tradeoff between standard and robust error in noiseless linear regression and systematically study the effect on RST on the tradeoff with several different perturbations and adversarial training algorithms on CIFAR-10.

Interpolated Adversarial Training (IAT) (Lamb et al., 2019) and Neural Architecture Search (NAS) (Cubuk et al., 2017) were proposed to mitigate the tradeoff bbetween standard and robust error empirically. IAS considers a different training algorithm based on Mixup, NAS (Cubuk et al., 2017) uses RL to search for more robust architectures. In Table 1, we also report the standard and robust errors of these methods. RST, IAT and NAS are incomparable as they find different tradeoffs between standard and robust error. Recently, Xie et al. (2020) showed that adversarial training with appropriate batch normalization (AdvProp) with small perturbations can actually *improve* standard error. However, since they only aim to improve and evaluate the standard error, it is unclear if the robust error improves. We believe that since RST provides a complementary statistical perspective on the tradeoff, it can be combined with methods like IAT, NAS or AdvProp to see further gains in standard and robust errors. We leave this to future work.

6. Conclusion

We study the commonly observed increase in standard error upon adversarial training due to generalization from finite data in a well-specified setting with consistent perturbations. Surprisingly, we show that methods that augment the training data with consistent perturbations, such as adversarial training, can increase the standard error even in the simple setting of noiseless linear regression where the true linear function has zero standard and robust error. Our analysis reveals that the mismatch between the inductive bias of models and the underlying distribution of the inputs causes the standard error to increase even when the augmented data is perfectly labeled. This insight motivates a method that provably eliminates the tradeoff in linear regression by incorporating an appropriate regularizer that utilizes the distribution of the inputs. While not immediately apparent, we show that this is a special case of the recently proposed robust self-training (RST) procedure that uses additional unlabeled data to estimate the distribution of the inputs. Previous works view RST as a method to improve the robust error by increasing the sample size. Our work provides some theoretical justification for why RST improves *both* the standard and robust error, thereby mitigating the tradeoff between accuracy and robustness in practice. How to best utilize unlabeled data, and whether sufficient unlabeled data would completely eliminate the tradeoff remain open questions.

Acknowledgements

We are grateful to Tengyu Ma, Yair Carmon, Ananya Kumar, Pang Wei Koh, Fereshte Khani, Shiori Sagawa and Karan Goel for valuable discussions and comments. This work was funded by an Open Philanthropy Project Award and NSF Frontier Award as part of the Center for Trustworthy Machine Learning (CTML). AR was supported by Google Fellowship and Open Philanthropy AI Fellowship. SMX was supported by an NDSEG Fellowship. FY was supported by the Institute for Theoretical Studies ETH Zurich and the Dr. Max Rossler and the Walter Haefner Foundation. FY and JCD were supported by the Office of Naval Research Young Investigator Awards.

References

Alzantot, M., Sharma, Y., Elgohary, A., Ho, B., Srivastava, M., and Chang, K. Generating natural language adversarial examples. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *arXiv*, 2019.

Belkin, M., Ma, S., and Mandal, S. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning (ICML)*, 2018.

Carmon, Y., Raghunathan, A., Schmidt, L., Liang, P., and Duchi, J. C. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Chen, L., Min, Y., Zhang, M., and Karbasi, A. More data can expand the generalization gap between adversarially robust and standard models. In *International Conference on Machine Learning (ICML)*, 2020.

Cubuk, E. D., Zoph, B., Schoenholz, S. S., and Le, Q. V. Intriguing properties of adversarial examples. *arXiv preprint arXiv:1711.02846*, 2017.

Diamond, S. and Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research (JMLR)*, 17(83):1–5, 2016.

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning (ICML)*, pp. 1802–1811, 2019.

Fawzi, A., Fawzi, O., and Frossard, P. Analysis of classifiers' robustness to adversarial perturbations. *Machine Learning*, 107(3):481–508, 2018.

Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA: Springer series in statistics New York, NY, USA:, 2001.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

Jia, R. and Liang, P. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1097–1105, 2012.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Lamb, A., Verma, V., Kannala, J., and Bengio, Y. Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy. *arXiv*, 2019.

Liang, T. and Rakhlin, A. Just interpolate: Kernel" ridgeless" regression can generalize. *arXiv preprint arXiv:1808.00387*, 2018.

Ma, S., Bassily, R., and Belkin, M. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In *International Conference on Machine Learning (ICML)*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks (published at ICLR 2018). *arXiv*, 2017.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Miyato, T., Maeda, S., Ishii, S., and Koyama, M. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Najafi, A., Maeda, S., Koyama, M., and Miyato, T. Robustness to adversarial perturbations in learning from incomplete data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Nakkiran, P. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532*, 2019.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1163–1171, 2016.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5014–5026, 2018.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019.

Uesato, J., Alayrac, J., Huang, P., Stanforth, R., Fawzi, A., and Kohli, P. Are labels required for improving adversarial robustness? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A. L., and Le, Q. V. Adversarial examples improve image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 819–828, 2020.

Xie, Q., Dai, Z., Hovy, E., Luong, M., and Le, Q. V. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019.

Yaeger, L., Lyon, R., and Webb, B. Effective training of a neural network character classifier for word recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 807–813, 1996.

Yang, F., Wang, Z., and Heinze-Deml, C. Invariance-inducing regularization using worst-case transformations suffices to boost accuracy and spatial robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Yin, D., Lopes, R. G., Shlens, J., Cubuk, E. D., and Gilmer, J. A fourier perspective on model robustness in computer vision. *arXiv preprint arXiv:1906.08988*, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *British Machine Vision Conference*, 2016.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.