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).

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).

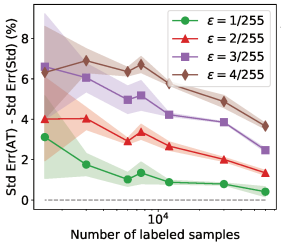


Figure 1. Gap between the standard error of adversarial training (Madry et al., 2018) with ℓ_∞ perturbations, and standard training. The gap decreases with increase in training set size, suggesting that the tradeoff between standard and robust error should disappear with infinite data.

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 satisfy invariances that the true model exhibits. One might think adding this information would only restrict the hypoth-

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

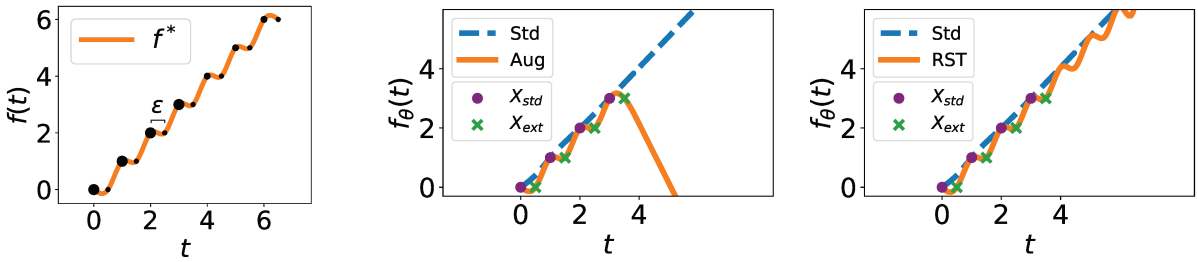


Figure 2. We consider function interpolation via cubic splines. **(Left)** The underlying distribution P_{x} denoted by sizes of the circles. The true function is a staircase. **(Middle)** With a small number of standard training samples (purple circles), an augmented estimator that fits local perturbations (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.

esis 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

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

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

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

where $\Sigma = \mathbb{E}_{P_x}[xx^\top]$ 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 $X = [X_{\text{std}}; X_{\text{ext}}], Y = [y_{\text{std}}; y_{\text{ext}}]$:

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

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

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 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 $\theta^\star_2$ is small as in Fig 3(a), $\hat{\theta}_{\text{aug}}$

[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).

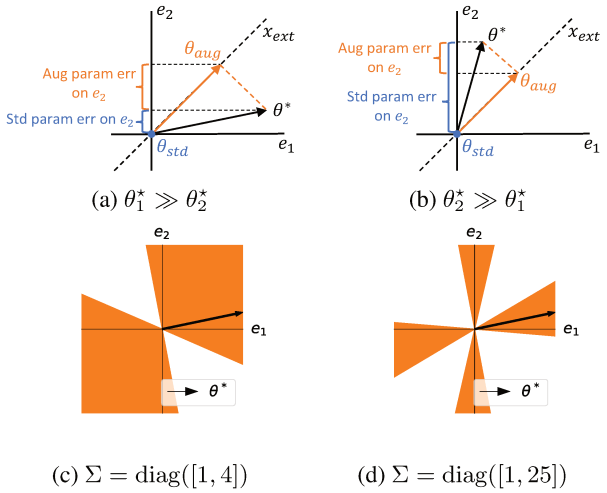


Figure 3. Illustration of the 3-D example described in Sec. 3.1. **(a)-(b) Effect of augmentation on parameter error for different $\theta^\star$.** We show the projections of the standard estimator $\hat{\theta}_{\text{std}}$ (blue circle), augmented estimator $\hat{\theta}_{\text{aug}}$ (orange arrow), 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.

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. 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).

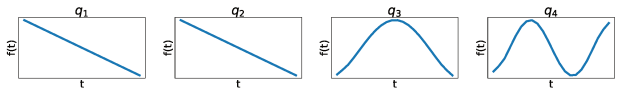


Figure 4. Top 4 eigenvectors of Σ in the splines problem (from Figure 2), representing wave functions in the input space. The "global" eigenfunctions, varying less over the domain, correspond to larger eigenvalues, making errors in global dimensions costly in terms of test error.

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*

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

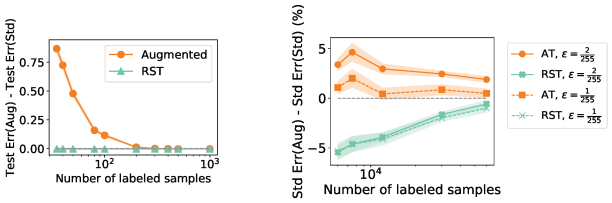
Simultaneously, $L_{rob}(\hat{\theta}_{rst}) = L_{std}(\hat{\theta}_{rst})$.

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

Method	Robust Test Acc.	Standard Test Acc.	
Standard Training	0.2%	94.6%	Vanilla Supervised
Worst-of-10	73.9%	95.0%	Vanilla Supervised
Random	67.7%	95.1%	Vanilla Supervised
RST + Worst-of-10 (this paper)	**75.1%**	**95.8%**	Semisupervised
RST + Random (this paper)	**70.9%**	**95.8%**	Semisupervised
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%	Existing baselines (smaller model)

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.

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 standard 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, Min 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. How to best utilize unlabeled data, and whether sufficient unlabeled data can 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.

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.

Min, Y., Chen, L., and Karbasi, A. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization. *arXiv preprint arXiv:2002.11080*, 2020.

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.

A. Transformations to handle arbitrary matrix norms

Consider a more general minimum norm estimator of the following form. Given inputs X and corresponding targets y as training data, we study the interpolation estimator,

$$\hat{\theta} = \arg\min_{\theta} \Big\{ \theta^\top M \theta : X\theta = y \Big\}, \tag{12}$$

where M is a positive definite (PD) matrix that incorporates prior knowledge about the true model. For simplicity, we present our results in terms of the ℓ_2 norm (ridgeless regression) as defined in Equation 12. However, all our results hold for arbitrary M–norms via appropriate rotations. Given an arbitrary PD matrix M, the rotated covariates $x \leftarrow M^{-1/2}x$ and rotated parameters $\theta \leftarrow M^{1/2}\theta$ maintain $y = X\theta$ and the M-norm of parameters simplifies to $\|\theta\|_2$.

B. Standard error of minimum norm interpolants

B.1. Projection operators

The projection operators $\Pi_{\text{std}}^{\perp}$ and $\Pi_{\text{aug}}^{\perp}$ are formally defined as follows.

$$\Sigma_{\text{std}} = X_{\text{std}}^\top X_{\text{std}}, \;\; \Pi_{\text{std}}^{\perp} = I - \Sigma_{\text{std}}^{+}\Sigma_{\text{std}} \tag{13}$$

$$\Sigma_{\text{aug}} = X_{\text{std}}^\top X_{\text{std}} + X_{\text{ext}}^\top X_{\text{ext}}, \;\; \Pi_{\text{aug}}^{\perp} = I - \Sigma_{\text{aug}}^{+}\Sigma_{\text{aug}}. \tag{14}$$

B.2. Invariant transformations may have arbitrary nullspace components

We show that the transformations which satisfy the invariance condition $(\tilde{x} - x)^\top \theta^\star = 0$ where $\tilde{x} \in T(x)$ is a transformation of x may have arbitrary nullspace components for general transfomation mappings T. Let Π_{std} and $\Pi_{\text{std}}^{\perp}$ be the column space and nullspace projections for the original data X_{std}. The invariance condition is equivalent to

$$(\tilde{x} - x)^\top \theta^\star = (\Pi_{\text{std}}(\tilde{x} - x) + \Pi_{\text{std}}^{\perp}(\tilde{x} - x))^\top \theta^\star = 0 \tag{15}$$

which implies that as long as $\Pi_{\text{std}}^{\perp}\theta^\star \neq 0$, then for any choice of nullspace component $\Pi_{\text{std}}^{\perp}(\tilde{x}) \in \text{Null}(X_{\text{std}}^\top X_{\text{std}})$, there is a choice of $\Pi_{\text{std}}\tilde{x}$ which satisfies the condition. Thus, we consider augmented points X_{ext} with arbitrary components in the nullspace of X_{std}.

B.3. Proof of Theorem 1

Inequality (8) follows from

$$\begin{aligned} L_{\text{std}}(\hat{\theta}_{\text{aug}}) - L_{\text{std}}(\hat{\theta}_{\text{std}}) &= (\theta^\star - \hat{\theta}_{\text{aug}})^\top \Sigma (\theta^\star - \hat{\theta}_{\text{aug}}) - (\theta^\star - \hat{\theta}_{\text{std}})^\top \Sigma (\theta^\star - \hat{\theta}_{\text{std}}) \\ &= (\Pi_{\text{aug}}^{\perp}\theta^\star)^\top \Sigma \Pi_{\text{aug}}^{\perp}\theta^\star - (\Pi_{\text{std}}^{\perp}\theta^\star)^\top \Sigma \Pi_{\text{std}}^{\perp}\theta^\star \\ &= w^\top \Sigma w - (w+v)^\top \Sigma (w+v) \\ &= -2w^\top \Sigma v - v^\top \Sigma v \end{aligned} \tag{16}$$

by decomposition of $\Pi_{\text{std}}^{\perp}\theta^\star = v + w$ where $v = \Pi_{\text{std}}^{\perp}\Pi_{\text{aug}}\theta^\star$ and $w = \Pi_{\text{std}}^{\perp}\Pi_{\text{aug}}^{\perp}\theta^\star$. Note that the error difference does scale with $\|\theta^\star\|^2$, although the sign of the difference does not.

B.4. Proof of Corollary 1

Corollary 1 presents three sufficient conditions under which the standard error of the augmented estimator $L_{\text{std}}(\hat{\theta}_{\text{aug}})$ is never larger than the standard error of the standard estimator $L_{\text{std}}(\hat{\theta}_{\text{std}})$.

1. When the population covariance $\Sigma = I$, from Theorem 1, we see that

$$L_{\text{std}}(\hat{\theta}_{\text{std}}) - L_{\text{std}}(\hat{\theta}_{\text{aug}}) = v^\top v + 2w^\top v = v^\top v \geq 0, \tag{17}$$

since $v = \Pi_{\text{std}}^{\perp}\Pi_{\text{aug}}\theta^\star$ and $w = \Pi_{\text{aug}}^{\perp}\theta^\star$ are orthogonal.

2. When $\Pi_{\text{aug}}^{\perp} = 0$, the vector w in Theorem 1 is 0, and hence we get

$$L_{\text{std}}(\hat{\theta}_{\text{std}}) - L_{\text{std}}(\hat{\theta}_{\text{aug}}) = v^\top v \geq 0. \tag{18}$$

3. We prove the eigenvector condition in Section B.7 which studies the effect of augmenting with a single extra point in general.

B.5. Proof of Proposition 1

The proof of Proposition 1 is based on the following two lemmas that are also useful for characterization purposes in Corollary 2.

Lemma 1. *If a PSD matrix Σ has non-equal eigenvalues, one can find two unit vectors w, v for which the following holds*

$$w^\top v = 0 \qquad \textit{and} \qquad w^\top \Sigma v \neq 0 \tag{19}$$

Hence, there exists a combination of original and augmentation dataset X_{std}, X_{ext} such that condition (19) holds for two directions $v \in Col(\Pi_{std}^{\perp}\Pi_{aug})$ and $w \in Col(\Pi_{std}^{\perp}\Pi_{aug}^{\perp}) = Col(\Pi_{aug}^{\perp})$.

Note that neither w nor v can be eigenvectors of Σ in order for both conditions in equation (19) to hold. Given a population covariance, fixed original and augmentation data for which condition (19) holds, we can now explicitly construct $\theta^\star$ for which augmentation increases standard error.

Lemma 2. *Assume Σ, X_{std}, X_{ext} are fixed. Then condition (19) holds for two directions $v \in Col(\Pi_{std}^{\perp}\Pi_{aug})$ and $w \in Col(\Pi_{std}^{\perp}\Pi_{aug}^{\perp})$ iff there exists a $\theta^\star$ such that $L_{std}(\hat{\theta}_{aug}) - L_{std}(\hat{\theta}_{std}) \geq c$ for some $c > 0$. Furthermore, the ℓ_2 norm of $\theta^\star$ needs to satisfy the following lower bounds with $c_1 := \|\hat{\theta}_{aug}\|^2 - \|\hat{\theta}_{std}\|^2$*

$$\begin{aligned} \|\theta^\star\|^2 - \|\hat{\theta}_{aug}\|^2 &\geq \beta_1 c_1 + \beta_2 \frac{c^2}{c_1} \\ \|\theta^\star\|^2 - \|\hat{\theta}_{std}\|^2 &\geq (\beta_1 + 1) c_1 + \beta_2 \frac{c^2}{c_1} \end{aligned} \tag{20}$$

where β_i are constants that depend on X_{std}, X_{ext}, Σ.

Proposition 1 follows directly from the second statement of Lemma 2 by minimizing the bound (20) with respect to c_1 which is a free parameter to be chosen during construction of $\theta^\star$ (see proof of Lemma (2). The minimum is attained for $c_1 = 2\sqrt{(\beta_1 + 1)(\beta_2 c^2)}$. We hence conclude that $\theta^\star$ needs to be sufficiently more complex than a good standard solution, i.e. $\|\theta^\star\|_2^2 - \|\hat{\theta}_{\text{std}}\|_2^2 > \gamma c$ where $\gamma > 0$ is a constant that depends on the $X_{\text{std}}, X_{\text{ext}}$.

B.6. Proof of technical lemmas

In this section we prove the technical lemmas that are used to prove Theorem 1.

B.6.1. Proof of Lemma 2

Any vector $\Pi_{\text{std}}^{\perp}\theta \in \text{Null}(\Sigma_{\text{std}})$ can be decomposed into orthogonal components $\Pi_{\text{std}}^{\perp}\theta = \Pi_{\text{std}}^{\perp}\Pi_{\text{aug}}^{\perp}\theta + \Pi_{\text{std}}^{\perp}\Pi_{\text{aug}}\theta$. Using the minimum-norm property, we can then always decompose the (rotated) augmented estimator $\hat{\theta}_{\text{aug}} \in \text{Col}(\Pi_{\text{aug}}^{\perp}) = \text{Col}(\Pi_{\text{std}}^{\perp}\Pi_{\text{aug}}^{\perp})$ and true parameter $\theta^\star$ by

$$\begin{aligned} \hat{\theta}_{\text{aug}} &= \hat{\theta}_{\text{std}} + \sum_{v_i \in \text{ext}} \zeta_i v_i \\ \theta^\star &= \hat{\theta}_{\text{aug}} + \sum_{w_j \in \text{rest}} \xi_j w_j, \end{aligned}$$

where we define "ext" as the set of basis vectors which span $\text{Col}(\Pi_{\text{std}}^{\perp}\Pi_{\text{aug}})$ and respectively "rest" for $\text{Null}(\Sigma_{\text{aug}})$. Requiring the standard error increase to be some constant $c > 0$ can be rewritten using identity (16) as follows

$$\begin{aligned} & L_{\text{std}}(\hat{\theta}_{\text{aug}}) - L_{\text{std}}(\hat{\theta}_{\text{std}}) = c \\ \iff & (\sum_{v_i \in \text{ext}} \zeta_i v_i)^\top \Sigma (\sum_{v_i \in \text{ext}} \zeta_i v_i) + c = -2(\sum_{w_j \in \text{rest}} \xi_j w_j)\Sigma(\sum_{v_i \in \text{ext}} \zeta_i v_i) \\ \iff & (\sum_{v_i \in \text{ext}} \zeta_i v_i)^\top \Sigma (\sum_{v_i \in \text{ext}} \zeta_i v_i) + c = -2 \sum_{w_j \in \text{rest}, v_i \in \text{ext}} \xi_j \zeta_i w_j^\top \Sigma v_i \end{aligned} \tag{21}$$

The left hand side of equation (21) is always positive, hence it is necessary for this equality to hold with any $c > 0$, that there exists at least one pair i, j such that $w_j^\top \Sigma v_i \neq 0$ and one direction of the iff statement is proved.

For the other direction, we show that if there exist $v \in \text{Col}(\Pi_{\text{std}}^{\perp}\Pi_{\text{aug}})$ and $w \in \text{Col}(\Pi_{\text{std}}^{\perp}\Pi_{\text{aug}}^{\perp})$ for which condition (19) holds (wlog we assume that the $w^\top \Sigma v < 0$) we can construct a $\theta^\star$ for which the inequality (8) in Theorem 1 holds as follows:

It is then necessary by our assumption that $\xi_j \zeta_i w_j^\top \Sigma v_i > 0$ for at least some i, j. We can then set $\zeta_i > 0$ such that $\|\hat{\theta}_{\text{aug}} - \hat{\theta}_{\text{std}}\|^2 = \|\zeta\|^2 = c_1 > 0$, i.e. that the augmented estimator is not equal to the standard estimator (else obviously there can be no difference in error and equality (21) cannot be satisfied for any desired error increase $c > 0$).

The choice of ξ minimizing $\|\theta^\star - \hat{\theta}_{\text{aug}}\|^2 = \sum_j \xi_j^2$ that also satisfies equation (21) is an appropriately scaled vector in the direction of $x = W^\top \Sigma V \zeta$ where we define $W := [w_1, \ldots, w_{|\text{rest}|}]$ and $V := [v_1, \ldots, v_{|\text{ext}|}]$. Defining $c_0 = \zeta^\top V^\top \Sigma V \zeta$ for convenience and then setting

$$\xi = -\frac{c_0 + c}{2\|x\|_2^2} x \tag{22}$$

which is well-defined since $x \neq 0$, yields a $\theta^\star$ such that augmentation increases standard error. It is thus necessary for $L_{\text{std}}(\hat{\theta}_{\text{aug}}) - L_{\text{std}}(\hat{\theta}_{\text{std}}) = c$ that

$$\begin{aligned} \sum_j \xi_j^2 &= \frac{(c_0 + c)^2}{4\|W^\top \Sigma V \zeta\|^2} = \frac{(\zeta^\top V^\top \Sigma V \zeta + c)^2}{4\zeta^\top V^\top \Sigma W W^\top \Sigma V \zeta} \\ &\geq \frac{(\zeta^\top V^\top \Sigma V \zeta)^2}{4\zeta^\top V^\top \Sigma W W^\top \Sigma V \zeta} + \frac{c^2}{4\zeta^\top V^\top \Sigma W W^\top \Sigma V \zeta} \\ &\geq \frac{c_1}{4} \frac{\lambda_{\min}^2(V^\top \Sigma V)}{\lambda_{\max}^2(W^\top \Sigma V)} + \frac{c^2}{4c_1 \lambda_{\max}^2(W^\top \Sigma V)}. \end{aligned}$$

By assuming existence of i, j such that $\xi_j \zeta_i w_j^\top \Sigma v_i \neq 0$, we are guaranteed that $\lambda_{\max}^2(W^\top \Sigma V) > 0$.

Note due to construction we have $\|\theta^\star\|_2^2 = \|\hat{\theta}_{\text{std}}\|_2^2 + \sum_i \zeta_i^2 + \sum_j \xi_j^2$ and plugging in the choice of ξ_j in equation (22) we have

$$\|\theta^\star\|_2^2 - \|\hat{\theta}_{\text{std}}\|_2^2 \geq c_1 \left[1 + \frac{\lambda_{\min}^2(V^\top \Sigma V)}{4\lambda_{\max}^2(W^\top \Sigma V)}\right] + \frac{c^2}{4\lambda_{\max}^2(W^\top \Sigma V)} \frac{1}{c_1}.$$

Setting $\beta_1 = \left[1 + \frac{\lambda_{\min}^2(V^\top \Sigma V)}{4\lambda_{\max}^2(W^\top \Sigma V)}\right]$, $\beta_2 = \frac{1}{4\lambda_{\max}^2(W^\top \Sigma V)}$ yields the result.

B.6.2. Proof of Lemma 1

Let $\lambda_1, \ldots, \lambda_m$ be the m non-zero eigenvalues of Σ and u_i be the corresponding eigenvectors. Then choose v to be any combination of the eigenvectors $v = U\beta$ where $U = [u_1, \ldots, u_m]$ where at least $\beta_i, \beta_j \neq 0$ for $\lambda_i \neq \lambda_j$. We next construct $w = U\alpha$ by choosing α as follows such that the inequality in (19) holds:

$$\begin{aligned} \alpha_i &= \frac{\beta_j}{\beta_i^2 + \beta_j^2} \\ \alpha_j &= \frac{-\beta_i}{\beta_i^2 + \beta_j^2} \end{aligned}$$

and $\alpha_k = 0$ for $k \neq i, j$. Then we have that $\alpha^\top \beta = 0$ and hence $w^\top v = 0$. Simultaneously

$$\begin{aligned} w^\top \Sigma v &= \lambda_i \beta_i \alpha_i + \lambda_j \beta_j \alpha_j \\ &= (\lambda_i - \lambda_j) \frac{\beta_i \beta_j}{\beta_i^2 + \beta_j^2} \neq 0 \end{aligned}$$

which concludes the proof of the first statement.

We now prove the second statement by constructing $\Sigma_{\text{std}} = X_{\text{std}}^\top X_{\text{std}}, \Sigma_{\text{ext}} = X_{\text{ext}}^\top X_{\text{ext}}$ using w, v. We can then obtain $X_{\text{std}}, X_{\text{ext}}$ using any standard decomposition method to obtain $X_{\text{std}}, X_{\text{ext}}$. We construct $\Sigma_{\text{std}}, \Sigma_{\text{ext}}$ using w, v. Without loss of generality, we can make them simultaneously diagonalizable. We construct a set of eigenvectors that is the same for both matrices paired with different eigenvalues. Let the shared eigenvectors include w, v. Then if we set the corresponding eigenvalues $\lambda_w(\Sigma_{\text{ext}}) = 0, \lambda_v(\Sigma_{\text{ext}}) > 0$ and $\lambda_w(\Sigma_{\text{std}}) = 0, \lambda_v(\Sigma_{\text{std}}) = 0$, then $\lambda_w(\Sigma_{\text{aug}}) = 0$ such that $w \in \text{Col}(\Pi_{\text{std}}^\perp \Pi_{\text{aug}}^\perp)$ and $v \in \text{Col}(\Pi_{\text{std}}^\perp \Pi_{\text{aug}})$. This shows the second statement. With this, we can design a $\theta^\star$ for which augmentation increases standard error as in Lemma 2.

B.7. Characterization Corollary 2

A simpler case to analyze is when we only augment with one extra data point. The following corollary characterizes which single augmentation directions lead to higher prediction error for the augmented estimator.

Corollary 2. *The following characterizations hold for augmentation directions that do not cause the standard error of the augmented estimator to be higher than the original estimator.*

(a) (in terms of ratios of inner products) *For a given $\theta^\star$, data augmentation does not increase the standard error of the augmented estimator for a single augmentation direction x_{ext} if*

$$\frac{x_{ext}^\top \Pi_{std}^\perp \Sigma \Pi_{std}^\perp x_{ext}}{x_{ext}^\top \Pi_{std}^\perp x_{ext}} - 2 \frac{(\Pi_{std}^\perp x_{ext})^\top \Sigma \Pi_{std}^\perp \theta^\star}{x_{ext}^\top \Pi_{std}^\perp \theta^\star} \leq 0 \tag{23}$$

(b) (in terms of eigenvectors) *Data augmentation does not increase standard error for any $\theta^\star$ if $\Pi_{std}^\perp x_{ext}$ is an eigenvector of Σ. However if one augments in the direction of a mixture of eigenvectors of Σ with different eigenvalues, there exists $\theta^\star$ such that augmentation increases standard error.*

(c) (depending on well-conditioning of Σ) *If $\frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)} \leq 2$ and $\Pi_{std}^\perp \theta^\star$ is an eigenvector of Σ, then no augmentations x_{ext} increase standard error.*

The form in Equation (23) compares ratios of inner products of $\Pi_{\text{std}}^\perp x_{\text{ext}}$ and $\Pi_{\text{std}}^\perp \theta^\star$ in two spaces: the one in the numerator is weighted by Σ whereas the denominator is the standard inner product. Thus, if Σ scales and rotates rather inhomogeneously, then augmenting with x_{ext} may hurt standard error. Here again, if $\Sigma = \gamma I$ for $\gamma > 0$, then the condition must hold.

B.7.1. Proof of Corollary 2 (a)

Note that for a single augmentation point $X_{\text{ext}} = x_{\text{ext}}^\top$, the orthogonal decomposition of $\Pi_{\text{std}}^\perp \theta^\star$ into $\text{Col}(\Pi_{\text{aug}}^\perp)$ and $\text{Col}(\Pi_{\text{std}}^\perp \Pi_{\text{aug}})$ is defined by $v = \frac{\Pi_{\text{std}}^\perp {x_{\text{ext}}}^\top \theta^\star}{\|\Pi_{\text{std}}^\perp x_{\text{ext}}\|^2} \Pi_{\text{std}}^\perp x_{\text{ext}}$ and $w = \Pi_{\text{std}}^\perp \theta^\star - v$ respectively. Plugging back into into identity (16) then yields the following condition for safe augmentations:

$$\begin{aligned} & 2(v - \Pi_{\text{std}}^\perp \theta^\star)^\top \Sigma v - v^\top \Sigma v \leq 0 \\ & v^\top \Sigma v - 2(\Pi_{\text{std}}^\perp \theta^\star)^\top \Sigma v \leq 0 \\ \iff & \Pi_{\text{std}}^\perp {x_{\text{ext}}}^\top \Sigma \Pi_{\text{std}}^\perp x_{\text{ext}} \leq 2(\Pi_{\text{std}}^\perp \theta^\star)^\top \Sigma \Pi_{\text{std}}^\perp x_{\text{ext}} \cdot \frac{\|\Pi_{\text{std}}^\perp x_{\text{ext}}\|^2}{\Pi_{\text{std}}^\perp {x_{\text{ext}}}^\top \theta^\star} \end{aligned} \tag{24}$$

Rearranging the terms yields inequality (23).

Safe augmentation directions for specific choices of $\theta^\star$ and Σ are illustrated in Figure 3.

B.7.2. Proof of Corollary 2 (b)

Assume that $\Pi_{\text{std}}^{\perp} x_{\text{ext}}$ is an eigevector of Σ with eigenvalue $\lambda > 0$. We have

$$\frac{x_{\text{ext}}^\top \Pi_{\text{std}}^{\perp} \Sigma \Pi_{\text{std}}^{\perp} x_{\text{ext}}}{x_{\text{ext}}^\top \Pi_{\text{std}}^{\perp} x_{\text{ext}}} - 2\frac{(\Pi_{\text{std}}^{\perp} x_{\text{ext}})^\top \Sigma \Pi_{\text{std}}^{\perp} \theta^\star}{x_{\text{ext}}^\top \Pi_{\text{std}}^{\perp} \theta^\star} = -\lambda < 0$$

for any $\theta^\star$. Hence by Corollary 2 (a), the standard error doesn't increase by augmenting with eigenvectors of Σ for any $\theta^\star$.

When the single augmentation direction v is not an eigenvector of Σ, by Lemma 1 one can find w such that $w^\top \Sigma v \neq 0$. The proof in Lemma 1 gives an explicit construction for w such that condition (19) holds and the result then follows directly by Lemma 2.

B.7.3. Proof of Corollary 2 (c)

Suppose $\Sigma \Pi_{\text{std}}^{\perp} \theta^\star = \lambda \Pi_{\text{std}}^{\perp} \theta^\star$ for some $\lambda_{\min}(\Sigma) \leq \lambda \leq \lambda_{\max}(\Sigma)$. Then starting with the expression (23),

$$\begin{aligned}\frac{x_{\text{ext}}^\top \Pi_{\text{std}}^{\perp} \Sigma \Pi_{\text{std}}^{\perp} x_{\text{ext}}}{x_{\text{ext}}^\top \Pi_{\text{std}}^{\perp} x_{\text{ext}}} - 2\frac{(\Pi_{\text{std}}^{\perp} x_{\text{ext}})^\top \Sigma \Pi_{\text{std}}^{\perp} \theta^\star}{x_{\text{ext}}^\top \Pi_{\text{std}}^{\perp} \theta^\star} &= \frac{x_{\text{ext}}^\top \Pi_{\text{std}}^{\perp} \Sigma \Pi_{\text{std}}^{\perp} x_{\text{ext}}}{x_{\text{ext}}^\top \Pi_{\text{std}}^{\perp} x_{\text{ext}}} - 2\lambda \\ &\leq \lambda_{\max}(\Sigma) - 2\lambda < 0\end{aligned}$$

by applying $\frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)} \leq 2$. Thus when $\Pi_{\text{std}}^{\perp} \theta^\star$ is an eigenvector of Σ, there are no augmentations x_{ext} that increase the standard error.

C. Details for spline staircase

We describe the data distribution, augmentations, and model details for the spline experiment in Figure 1 and toy scenario in Figure 2. Finally, we show that we can construct a simplified family of spline problems where the ratio between standard errors of the augmented and standard estimators increases unboundedly as the number of stairs.

C.1. True model

We consider a finite input domain

$$\mathcal{T} = \{0, \epsilon, 1, 1+\epsilon, \ldots, s-1, s-1+\epsilon\} \tag{25}$$

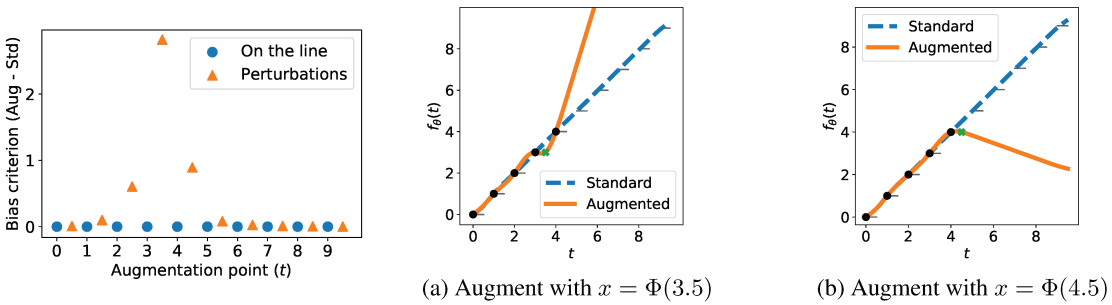
for some integer s corresponding to the total number of "stairs" in the staircase problem. Let $\mathcal{T}_{\text{line}} \subset \mathcal{T} = \{0, 1, \ldots, s-1\}$. We define the underlying function $f^\star : \mathbb{R} \mapsto \mathbb{R}$ as $f^\star(t) = \lfloor t \rfloor$. This function takes a staircase shape, and is linear when restricted to $\mathcal{T}_{\text{line}}$.

Sampling training data X_{std} We describe the data distribution in terms of the one-dimensional input t, and by the one-to-one correspondence with spline basis features $x = X(t)$, this also defines the distribution of spline features $x \in \mathcal{X}$. Let $w \in \Delta_s$ define a distribution over $\mathcal{T}_{\text{line}}$ where Δ_s is the probability simplex of dimension s. We define the data distribution with the following generative process for one sample t. First, sample a point i from $\mathcal{T}_{\text{line}}$ according to the categorical distribution described by w, such that $i \sim \text{Categorical}(w)$. Second, sample t by perturbing i with probability δ such that

$$t = \begin{cases} i & \text{w.p. } 1-\delta \\ i+\epsilon & \text{w.p. } \delta. \end{cases}$$

The sampled t is in $\mathcal{T}_{\text{line}}$ with probability $1-\delta$ and $\mathcal{T}_{\text{line}}^c$ with probability δ, where we choose δ to be small.

Sampling augmented points X_{ext} For each element t_i in the training set, we augment with $\tilde{T}_i = [\overset{u.a.r}{\sim} B(t_i)]$, an input chosen uniformly at random from $B(t_i) = \{\lfloor t_i \rfloor, \lfloor t_i \rfloor + \epsilon\}$. Recall that in our work, we consider data augmentation where the targets associated with the augmented points are from the ground truth oracle. Notice that by definition, $f^\star(\tilde{t}_i) = f^\star(t_i)$ for all $\tilde{t} \in B(t_i)$, and thus we can set the augmented targets to be $\tilde{y}_i = y_i$. This is similar to random data augmentation in images (Yaeger et al., 1996; Krizhevsky et al., 2012), where inputs are perturbed in a way that preserves the label.



(a) Augment with $x = \Phi(3.5)$ (b) Augment with $x = \Phi(4.5)$

Figure 7. Visualization of the effect of single augmentation points in the noiseless spline problem given an initial dataset $X_{\text{std}} = \{\Phi(t) : t \in \{0, 1, 2, 3, 4\}\}$. The standard estimator defined by X_{std} is linear. **(a)** Plot of the difference term in Corollary 2 (a), is positive when augmenting a single point causes higher test error. Augmenting with points on $\mathcal{X}_{\text{line}}$ does not affect the bias, but augmenting with any element of $\{X(t) : t \in \{2.5, 3.5, 4.5\}\}$ hurts the bias of the augmented estimator dramatically. **(b), (c)** Augmenting with $X(3.5)$ or $X(4.5)$ hurts the bias by changing the direction of extrapolation.

C.2. Spline model

We parameterize the spline predictors as $f_\theta(t) = \theta^\top X(t)$ where $X : \mathbb{R} \to \mathbb{R}^d$ is the cubic B-spline feature mapping (Friedman et al., 2001) and the norm of $f_\theta(t)$ can be expressed as $\theta^\top M \theta$ for a matrix M that penalizes a large second derivative norm where $[M]_{ij} = \int X_i^{''}(u) X_j^{''}(u) du$. Notice that the splines problem is a linear regression problem from $\mathbb{R}^d$ to $\mathbb{R}$ in the feature domain $X(t)$, allowing direct application of Theorem 1. As a linear regression problem, we define the finite domain as $\mathcal{X} = \{X(t) : t \in \mathcal{T}\}$ containing $2s$ elements in $\mathbb{R}^d$. There is a one-to-one correspondence between t and $X(t)$, such that X^{-1} is well-defined. We define the features that correspond to inputs in $\mathcal{T}_{\text{line}}$ as $\mathcal{X}_{\text{line}} = \{x : X^{-1}(x) \in \mathcal{T}_{\text{line}}\}$. Using this feature mapping, there exists a $\theta^\star$ such that $f_{\theta^\star}(t) = f^\star(t)$ for $t \in \mathcal{T}$.

Our hypothesis class is the family of cubic B-splines as defined in (Friedman et al., 2001). Cubic B-splines are piecewise cubic functions, where the endpoints of each cubic function are called the knots. In our example, we fix the knots to be $[0, \epsilon, 1, \ldots, s-1, s-1+\epsilon]$, which places a knot on every point in $\mathcal{T}$. This ensures that the function class contains an interpolating function on all $t \in \mathcal{T}$, i.e. for some $\theta^\star$,

$$f_{\theta^\star}(t) = \theta^{\star\top} X(t) = f^\star(t) = \lfloor t \rfloor.$$

We solve the minimum norm problem

$$\hat{\theta}_{\text{std}} = \arg\min_\theta \{\theta^\top M \theta : X_{\text{std}} \theta = y_{\text{std}}\} \tag{26}$$

for the standard estimator and the corresponding augmented problem to obtain the augmented estimator.

C.3. Evaluating Corollary 2 (a) for splines

We now illustrate the characterization for the effect of augmentation with different single points in Theorem 2 (a) on the splines problem. We assume the domain to $\mathcal{T}$ as defined in equation 25 with $s = 10$ and our training data to be $X_{\text{std}} = \{X(t) : t \in \{0, 1, 2, 3, 4\}\}$. Let *local* perturbations be spline features for $\tilde{t} \notin \mathcal{T}_{\text{line}}$ where $\tilde{t} = t + \epsilon$ is ϵ away from some $t \in \{0, 1, 2, 3, 4\}$ from the training set. We examine all possible single augmentation points in Figure 7 (a) and plot the calculated standard error difference as defined in equation (24). Figure 7 shows that augmenting with an additional point from $\{X(t) : t \in \mathcal{T}_{\text{line}}\}$ does not affect the bias, but adding any perturbation point in $\{X(\tilde{t}) : \tilde{t} \in \{2.5, 3.5, 4.5\}\}$ where $\tilde{t} \notin \mathcal{T}_{\text{line}}$ increases the error significantly by changing the direction in which the estimator extrapolates. Particularly, *local* augmentations near the boundary of the original dataset hurt the most while other augmentations do not significantly affect the bias of the augmented estimator.

C.3.1. Local and global structure in the spline staircase

In the spline staircase, the local perturbations can be thought of as fitting high frequency noise in the function space, where fitting them causes a global change in the function.

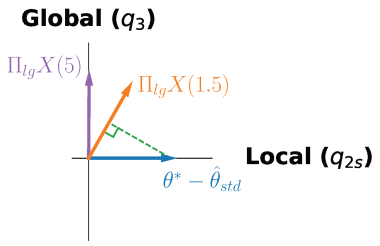


Figure 8. Nullspace projections onto global direction q_3 and local direction q_{2s} in Null(Σ) via Π_{lg}, representing global and local eigenvectors respectively. The local perturbation $\Pi_{\text{lg}}\hat{\Phi}(1.5)$ has both local and global components, creating a high-error component in the global direction.

To see this, we transform the problem to minimum ℓ_2 norm linear interpolation using features $X_M(t) = X(t)M^{-1/2}$ so that the results from Section 3.2 apply directly. Let Σ be the population covariance of X_M for a uniform distribution over the discrete domain consisting of s stairs and their perturbations (Figure 2). Let $Q = [q_i]_{i=1}^{2s}$ be the eigenvectors of Σ in decreasing order of their corresponding eigenvalues. The visualization in Figure 4 shows that q_i are wave functions in the original input space; the "frequency" of the wave increases as i increases.

Suppose the original training set consists of two points, $X_{\text{std}} = [X_M(0), X_M(1)]^\top$. We study the effect of augmenting point x_{ext} in terms of q_i above. First, we find that the first two eigenvectors corresponding to linear functions satisfy $\Pi_{\text{std}}^{\perp} q_1 = \Pi_{\text{std}}^{\perp} q_2 = 0$. Intuitively, this is because the standard estimator is linear. For ease of visualization, we consider the 2D space in Null(Σ) spanned by $\Pi_{\text{std}}^{\perp} q_3$ (global direction, low frequency) and $\Pi_{\text{std}}^{\perp} q_{2s}$ (local direction, high frequency). The matrix $\Pi_{\text{lg}} = [\Pi_{\text{std}}^{\perp} q_3,\ \Pi_{\text{std}}^{\perp} q_{2s}]^\top$ projects onto this space. Note that the same results hold when projecting onto all $\Pi_{\text{std}}^{\perp} q_i$ in Null(Σ).

In terms of the simple 3-D example in Section 3.1, the global direction corresponds to the costly direction with large eigenvalue, as changes in global structure heavily affect the standard error. Figure 8 plots the projections $\Pi_{\text{lg}}\theta^\star$ and $\Pi_{\text{lg}}X_{\text{ext}}$ for different X_{ext}. When $\theta^\star$ has high frequency variations and is complex, $\Pi_{\text{lg}}\theta^\star = (\theta^\star - \hat{\theta}_{\text{std}})$ is aligned with the local dimension. For x_{ext} immediately local to training points, the projection $\Pi_{\text{lg}}x_{\text{ext}}$ (orange vector in Figure 8) has both local and global components. Augmenting these local perturbations introduces error in the global component. For other x_{ext} farther from training points, $\Pi_{\text{lg}}x_{\text{ext}}$ (blue vector in Figure 8) is almost entirely global and perpendicular to $\theta^\star - \hat{\theta}_{\text{std}}$, leaving bias unchanged. Thus, augmenting data close to original data cause estimators to fit local components at the cost of the costly global component which changes overall structure of the predictor like in Figure 2(middle). The choice of inductive bias in the M–norm being minimized results in eigenvectors of Σ that correspond to local and global components, dictating this tradeoff.

C.4. Data augmentation can be quite painful for splines

We construct a family of spline problems such that as the number the augmented estimator has much higher error than the standard estimator. We assume that our predictors are from the full family of cubic splines.

Sampling distribution. We define a modified domain with continuous intervals $\mathcal{T} = \cup_{t=0}^{s-1}[t, t+\epsilon]$. Considering only s which is a multiple of 2, we sample the original data set as described in Section C.1 with the following probability mass w:

$$w(t) = \begin{cases} \frac{1-\gamma}{s/2} & t < s/2,\ t \in \mathcal{T}_{\text{line}} \\ \frac{\gamma}{s/2} & t \geq s/2,\ t \in \mathcal{T}_{\text{line}}. \end{cases} \tag{27}$$

for $\gamma \in [0, 1)$. We define a probability distribution $P_{\mathcal{T}}$ on $\mathcal{T}$ for a random variable T by setting $T = Z + S(Z)$ where $Z \sim$ Categorical(w) and the Z-dependent perturbation $S(z)$ is defined as

$$S(z) \sim \begin{cases} \text{Uniform}([z, z+\epsilon]) & \text{w.p. } \delta \\ z, & \text{w.p. } 1-\delta. \end{cases} \tag{28}$$

We obtain the training dataset $X_{\text{std}} = \{X(t_1), \ldots, X(t_n)\}$ by sampling $t_i \sim P_{\mathcal{T}}$.

Augmenting with an interval. Consider a modified augmented estimator for the splines problem, where for each point t_i we augment with the entire interval $[\lfloor t_i \rfloor, \lfloor t_i \rfloor + \epsilon]$ with $\epsilon \in [0, 1/2)$ and the estimator is enforced to output $f_{\hat{\theta}}(x) = y_i = \lfloor t_i \rfloor$ for all x in the interval $[\lfloor t_i \rfloor, \lfloor t_i \rfloor + \epsilon]$. Additionally, suppose that the ratio $s/n = O(1)$ between the number of stairs s and the number of samples n is constant.

In this simplified setting, we can show that the standard error of the augmented estimator grows while the standard error of the standard estimator decays to 0.

Theorem 3. *Let the setting be defined as above. Then with the choice of* $\delta = \frac{\log(s^7) - \log(s^7 - 1)}{s}$ *and* $\gamma = c/s$ *for a constant* $c \in [0, 1)$*, the ratio between standard errors is lower bounded as*

$$\frac{R(\hat{\theta}_{aug})}{R(\hat{\theta}_{std})} = \Omega(s^2) \tag{29}$$

which goes to infinity as $s \to \infty$*. Furthermore,* $R(\hat{\theta}_{std}) \to 0$ *as* $s \to \infty$*.*

Proof. We first lower bound the standard error of the augmented estimator. Define E_1 as the event that only the lower half of the stairs is sampled, i.e. $\{t : t < s/2\}$, which occurs with probability $(1-\gamma)^n$. Let $t^\star = \max_i \lfloor t_i \rfloor$ be the largest "stair" value seen in the training set. Note that the min-norm augmented estimator will extrapolate with zero derivative for $t \geq \max_i \lfloor t_i \rfloor$. This is because on the interval $[t^\star, t^\star + \epsilon]$, the augmented estimator is forced to have zero derivative, and the solution minimizing the second derivative of the prediction continues with zero derivative for all $t \geq t^\star$. In the event E_1, $t^\star \leq s/2 - 1$, where $t^* = s/2 - 1$ achieves the lowest error in this event. As a result, on the points in the second half of the staircase, i.e. $t = \{t \in \mathcal{T} : t > \frac{s}{2} - 1\}$, the augmented estimator incurs large error:

$$\begin{aligned} R(\hat{\theta}_{\text{aug}} \mid E_1) &\geq \sum_{t=s/2}^{s} (t - (s/2 - 1))^2 \cdot \frac{\gamma}{s/2} \\ &= \sum_{t=1}^{s/2} t^2 \cdot \frac{\gamma}{s/2} = \frac{\gamma}{6}(s^2 + 2s + 1). \end{aligned}$$

Therefore the standard error of the augmented estimator is bounded by

$$\begin{aligned} R(\hat{\theta}_{\text{aug}}) \geq R(\hat{\theta}_{\text{aug}} \mid E_1) P(E_1) &= \frac{\gamma}{6}(s^2 + 2s + 1)(1-\gamma)^n \\ &\geq \frac{1}{6}\gamma(1 - \gamma n)(s^2 + 2s + 1) \\ &= \Omega(\frac{c - c^2}{s}(s^2 + 2s + 1)) = \Omega(s) \end{aligned}$$

where in the first line, we note that the error on each interval is the same and the probability of each interval is $(1-\delta)\frac{\gamma}{s/2} + \epsilon\frac{\delta}{\epsilon} \cdot \frac{\gamma}{s/2} = \frac{\gamma}{s/2}$.

Next we upper bound the standard error of the standard estimator. Define E_2 to be the event where all points are sampled from $\mathcal{T}_{\text{line}}$, which occurs with probability $(1-\delta)^n$. In this case, the standard estimator is linear and fits the points on $\mathcal{T}_{\text{line}}$ with zero error, while incurring error for all points not in $\mathcal{T}_{\text{line}}$. Note that the probability density of sampling a point not in $\mathcal{T}_{\text{line}}$ is either $\frac{\delta}{\epsilon} \cdot \frac{1-\gamma}{s/2}$ or $\frac{\delta}{\epsilon} \cdot \frac{\gamma}{s/2}$, which we upper bound as $\frac{\delta}{\epsilon} \cdot \frac{1}{s/2}$.

$$\begin{aligned} R(\hat{\theta}_{\text{std}} \mid E_2) = \sum_{t=1}^{s-1} \frac{\delta}{\epsilon} \cdot \frac{1}{s/2} \int_0^\epsilon u^2 du &= \frac{\delta}{\epsilon} \cdot \frac{1}{s/2} O(s\epsilon^3) \\ &= O(\delta) \end{aligned}$$

Therefore for event E_2, the standard error is bounded as

$$\begin{aligned} R(\hat{\theta}_{\text{std}} \mid E_2)P(E_2) &= O(\delta)(1-\delta)^n \\ &= O(\delta)e^{-\delta n} \\ &= O(\delta \cdot \frac{s^7-1}{s^7}) \\ &= O(\delta) = O(\frac{\log(s^7)-\log(s^7-1)}{s}) = O(1/s) \end{aligned}$$

since $\log(s^7) - \log(s^7-1) \leq 1$ for $s \geq 2$. For the complementary event E_2^c, note that cubic spline predictors can grow only as $O(t^3)$, with error at most $O(t^6)$. Therefore the standard error for case E_2^c is bounded as

$$\begin{aligned} R(\hat{\theta}_{\text{std}} \mid E_2^c)P(E_2^c) &\leq O(t^6)(1-e^{-\delta n}) \\ &= O(t^6)O(\frac{1}{s^7}) = O(1/s) \end{aligned}$$

Putting the parts together yields

$$\begin{aligned} R(\hat{\theta}_{\text{std}}) &= R(\hat{\theta}_{\text{std}} \mid E_2)P(E_2) + R(\hat{\theta}_{\text{std}} \mid E_2^c)P(E_2^c) \\ &\leq O(1/s) + O(1/s) = O(1/s). \end{aligned}$$

Thus overall, $R(\hat{\theta}_{\text{std}}) = O(1/s)$ and combining the bounds yields the result. □

D. Robust Self-Training

We define the linear robust self-training estimator from Equation (11) and expand all the terms.

$$\begin{aligned} \hat{\theta}_{\text{rst}} \in \arg\min_{\theta} \Big\{ & \mathbb{E}_{P_{\text{x}}}[(x^\top \theta_{\text{int-std}} - x^\top \theta)^2] : \\ & X_{\text{std}}\theta = y_{\text{std}}, \max_{x_{\text{adv}} \in T(x)} (x_{\text{adv}}^\top \theta - y)^2 = 0 \; \forall x, y \in X_{\text{std}}, y_{\text{std}}, \\ & \mathbb{E}_{P_{\text{x}}}[\max_{x_{\text{adv}} \in T(x)} (x_{\text{adv}}^\top \theta - x^\top \theta)^2] = 0 \Big\}. \end{aligned} \tag{30}$$

Notice that for unlabeled components of the estimator, we assume access to the data distribution P_{x} and thus optimize the population quantities.

As we show in the next subsection, we can rewrite the robust self-training estimator into the following reduced form, more directly connecting to the general analysis of adding extra data X_{ext} in min-norm linear regression.

$$\hat{\theta}_{\text{rst}} \in \arg\min_{\theta} \Big\{ (\theta - \theta_{\text{int-std}})^\top \Sigma (\theta - \theta_{\text{int-std}}) : X_{\text{std}}\theta = y_{\text{std}}, X_{\text{ext}}\theta = 0 \Big\} \tag{31}$$

for the appropriate choice of X_{ext}, as shown in Section D.1. Here, we can interpret X_{ext} as the difference between the perturbed inputs and original inputs. These are perturbations which we want the model to be invariant to, and hence output zero.

D.1. Robust self-training algorithm in linear regression

We give an algorithm for constructing X_{ext} which enforces the population robustness constraints. Suppose we are given Σ, the population covariance of P_{x}. In robust self-training, we enforce that the model is consistent over perturbations of the labeled data X_{std} and (infinite) unlabeled data. To do this, we add linear constraints of the form $x_{\text{adv}}^\top \theta - x^\top \theta = 0$, where $x_{\text{adv}} \in T(x)$ for all x. We can view these linear constraints as augmenting the dataset with input-target pairs $(x_{\text{ext}}, 0)$ where $x_{\text{ext}} = x_{\text{adv}} - x$. By assumption, $x_{\text{ext}}^\top \theta^\star = 0$ so these augmentations fit into our data augmentation framework.

However, when we enforce these constraints over the entire population P_{x} or when there are an infinite number of transformations in $T(x)$, a naive implementation requires augmenting with infinitely many points. Noting that the space of

augmentations x_{ext} satisfying $x_{\text{ext}}^\top \theta^\star = 0$ is a linear subspace, we can instead summarize the augmentations with a basis that spans the transformations. Let the space of perturbations be $\mathcal{T} = \cup_{x \in \text{supp}(P_{\mathsf{x}}), x_{\text{adv}} \in T(x)} x_{\text{adv}} - x$. Note that this space of perturbations also contains perturbations of the original data X_{std} if X_{std} is in the support of P_{x}. If X_{std} is not in the support of P_{x}, the behavior of the estimator on these points do not affect standard or robust error. Assuming that we can efficiently optimize over $\mathcal{T}$, we construct the basis by an iterative procedure reminiscent of adversarial training.

1. Set $t = 0$. Initialize $\theta^t = \theta_{\text{int-std}}$ and $(X_{\text{ext}})_0$ as an empty matrix.
2. At iteration t, solve for $x_{\text{ext}}^t = \arg\max_{x_{\text{ext}} \in \mathcal{T}} (x_{\text{ext}}^\top \theta^t)^2$. If the objective is unbounded, choose any x_{ext}^t such that $x_{\text{ext}}^\top \theta^t \neq 0$.
3. If ${\theta^t}^\top x_{\text{ext}}^t = 0$, stop and return $(X_{\text{ext}})_t$.
4. Otherwise, add x_{ext}^t as a row in $(X_{\text{ext}})_t$. Increment t and let θ^t solve (31) with $X_{\text{ext}} = (X_{\text{ext}})_t$.
5. Return to step 2.

In each iteration, we search for a perturbation that the current θ^t is not invariant to. If we can find such a perturbation, we add it to the constraint set in $(X_{\text{ext}})_t$. We stop when we cannot find such a perturbation, implying that the rows of $(X_{\text{ext}})_t$ and X_{std} span $\mathcal{T}$. The final RST estimator solves (31) using X_{ext} returned from this procedure.

This procedure terminates within $O(d)$ iterations. To see this, note that θ^t is orthogonal to all rows of $(X_{\text{ext}})_t$. Any vector in the span of $(X_{\text{ext}})_t$ is orthogonal to θ^t. Thus, if ${\theta^t}^\top x_{\text{ext}}^t \neq 0$, then x_{ext}^t must not be in the span of $(X_{\text{ext}})_t$. At most $d - \text{rank}(X_{\text{std}})$ such new directions can be added until $(X_{\text{ext}})_t$ is full rank. When $(X_{\text{ext}})_t$ is full rank, ${\theta^t}^\top x_{\text{ext}}^t = 0$ must hold and the algorithm terminates.

D.2. Proof of Theorem 2

In this section, we prove Theorem 2, which we reproduce here.

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*

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

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

Proof. We work with the RST estimator in the form from Equation (31). We note that our result applies generally to any extra data $X_{\text{ext}}, y_{\text{ext}}$. We define $\Sigma_{\text{std}} = X_{\text{std}}^\top X_{\text{std}}$. Let $\{u_i\}$ be an orthonormal basis of the kernel $\text{Null}(\Sigma_{\text{std}} + X_{\text{ext}}^\top X_{\text{ext}})$ and $\{v_i\}$ be an orthonormal basis for $\text{Null}(\Sigma_{\text{std}}) \setminus \text{span}(\{u_i\})$. Let U and V be the linear operators defined by $Uw = \sum_i u_i w_i$ and $Vw = \sum_i v_i w_i$, respectively, noting that $U^\top V = 0$. Defining $\Pi_{\text{std}}^\perp := (I - \Sigma_{\text{std}}^\dagger \Sigma_{\text{std}})$ to be the projection onto the null space of X_{std}, we see that there are unique vectors ρ, α such that

$$\theta^\star = (I - \Pi_{\text{std}}^\perp)\theta^\star + U\rho + V\alpha. \tag{32a}$$

As $\theta_{\text{int-std}}$ interpolates the standard data, we also have

$$\theta_{\text{int-std}} = (I - \Pi_{\text{std}}^\perp)\theta^\star + Uw + Vz, \tag{32b}$$

as $X_{\text{std}} Uw = X_{\text{std}} Vz = 0$, and finally,

$$\hat{\theta}_{\text{rst}} = (I - \Pi_{\text{std}}^\perp)\theta^\star + U\rho + V\lambda \tag{32c}$$

where we note the common ρ between Eqs. (32a) and (32c).

Using the representations (32) we may provide an alternative formulation for the augmented estimator (30), using this to prove the theorem. Indeed, writing $\theta_{\text{int-std}} - \hat{\theta}_{\text{rst}} = U(w - \rho) + V(z - \lambda)$, we immediately have that the estimator has the form (32c), with the choice

$$\lambda = \arg\min_{\lambda} \left\{ (U(w - \rho) + V(z - \lambda))^\top \Sigma (U(w - \rho) + V(z - \lambda)) \right\}.$$

The optimality conditions for this quadratic imply that

$$V^\top \Sigma V(\lambda - z) = V^\top \Sigma U(w - \rho). \tag{33}$$

Now, recall that the standard error of a vector θ is $R(\theta) = (\theta - \theta^\star)^\top \Sigma(\theta - \theta^\star) = \|\theta - \theta^\star\|_\Sigma^2$, using Mahalanobis norm notation. In particular, a few quadratic expansions yield

$$\begin{aligned}
&R(\theta_{\text{int-std}}) - R(\hat{\theta}_{\text{rst}}) \\
&= \|U(w-\rho) + V(z-\alpha)\|_\Sigma^2 - \|V(\lambda - \alpha)\|_\Sigma^2 \\
&= \|U(w-\rho) + Vz\|_\Sigma^2 + \|V\alpha\|_\Sigma^2 - 2(U(w-\rho) + Vz)^\top \Sigma V\alpha - \|V\lambda\|_\Sigma^2 - \|V\alpha\|_\Sigma^2 + 2(V\lambda)^\top \Sigma V\alpha \\
&\overset{(i)}{=} \|U(w-\rho) + Vz\|_\Sigma^2 - 2(V\lambda)^\top \Sigma V\alpha - \|V\lambda\|_\Sigma^2 + 2(V\lambda)^\top V\alpha \\
&= \|U(w-\rho) + Vz\|_\Sigma^2 - \|V\lambda\|_\Sigma^2,
\end{aligned} \tag{34}$$

where step (i) used that $(U(w-\rho))^\top \Sigma V = (V(\lambda - z))^\top \Sigma V$ from the optimality conditions (33).

Finally, we consider the rightmost term in equality (34). Again using the optimality conditions (33), we have

$$\|V\lambda\|_\Sigma^2 = \lambda^\top V^\top \Sigma^{1/2}\Sigma^{1/2}(U(w-\rho) + Vz) \le \|V\lambda\|_\Sigma \|U(w-\rho) + Vz\|_\Sigma$$

by Cauchy-Schwarz. Revisiting equality (34), we obtain

$$\begin{aligned}
R(\theta_{\text{int-std}}) - R(\hat{\theta}_{\text{rst}}) &= \|U(w-\rho) + Vz\|_\Sigma^2 - \frac{\|V\lambda\|_\Sigma^4}{\|V\lambda\|_\Sigma^2} \\
&\ge \|U(w-\rho) + Vz\|_\Sigma^2 - \frac{\|V\lambda\|_\Sigma^2 \|U(w-\rho) + Vz\|_\Sigma^2}{\|V\lambda\|_\Sigma^2} = 0,
\end{aligned}$$

as desired.

Finally, we show that $L_{\text{std}}(\hat{\theta}_{\text{rst}}) = L_{\text{rob}}(\hat{\theta}_{\text{rst}})$. Here, choose X_{ext} to contain at most d basis vectors which span $\{x_{\text{adv}} : x_{\text{adv}} \in T(x), \forall x \in \text{supp}(P_{\text{x}})\}$. Thus, the robustness constraint $\mathbb{E}_{P_{\text{x}}}[\max_{x_{\text{adv}} \in T(x)} (x_{\text{adv}}^\top \hat{\theta}_{\text{rst}} - x^\top \hat{\theta}_{\text{rst}})] = 0$ is satisfied by fitting X_{ext}. By fitting X_{ext}, we thus have $x_{\text{adv}}^\top \hat{\theta}_{\text{rst}} - x^\top \hat{\theta}_{\text{rst}} = 0$ for all $x_{\text{adv}} \in T(x), x \in \text{supp}(P_{\text{x}})$ up to a measure zero set of x. Thus, the robust error is

$$L_{\text{rob}}(\hat{\theta}_{\text{rst}}) = \mathbb{E}_{P_{\text{x}}}[\max_{x_{\text{adv}} \in T(x)} (x_{\text{adv}}^\top \hat{\theta}_{\text{rst}} - x_{\text{adv}}^\top \theta^\star)^2] = \mathbb{E}_{P_{\text{x}}}[(x^\top \hat{\theta}_{\text{rst}} - x^\top \theta)] = L_{\text{std}}(\hat{\theta}_{\text{rst}})$$

where we used that $x_{\text{adv}}^\top \theta^\star = x^\top \theta^\star$ by assumption. Since $L_{\text{rob}}(\hat{\theta}_{\text{rst}}) \ge L_{\text{std}}(\hat{\theta}_{\text{rst}})$, $\hat{\theta}_{\text{rst}}$ has perfect consistency, achieving the lowest possible robust error (matching the standard error). □

D.3. Different instantiations of the general RST procedure

The general RST estimator (Equation 10) is simply a weighted combination of some standard loss and some robust loss on the labeled and unlabeled data. Throughout, we assume the same notation as that used in the definition of the general estimator. $X_{\text{std}}, y_{\text{std}}$ denote the standard training set and we have access to m unlabeled points $\tilde{x}_i, i = 1, \dots m$.

D.3.1. Projected Gradient Adversarial Training

In the first variant, RST + PG-AT, we use multiclass logistic loss (cross-entropy) as the standard loss. The robust loss is the maximum cross-entropy loss between any perturbed input (within the set of tranformations $T(\cdot)$) and the label (pseudo-label in the case of unlabeled data). We set the weights such that the estimator can be written as follows.

$$\begin{aligned}
\hat{\theta}_{\text{rst+pg-at}} := \arg\min_\theta \Bigg\{ &\frac{1-\lambda}{n} \sum_{(x,y) \in [X_{\text{std}}, y_{\text{std}}]} (1-\beta)\ell(f_\theta(x), y) + \beta\, \ell(f_\theta(x_{\text{adv}}), y) \\
&+ \frac{\lambda}{m} \sum_{i=1}^m (1-\beta)\ell(f_\theta(\tilde{x}_i), f_{\hat{\theta}_{\text{std}}}(\tilde{x}_i)) + \beta\, \ell(f_\theta(\tilde{x}_{\text{adv}i}), f_{\hat{\theta}_{\text{std}}}(\tilde{x}_i)) \Bigg\},
\end{aligned} \tag{35}$$

In practice, x_{adv} is found by performing a few steps of projected gradient method on $\ell(f_\theta(x), y)$, and similarly $\tilde{x}_{\text{adv}}$ by performing a few steps of projected gradient method on $\ell(f_\theta(\tilde{x}), f_{\hat{\theta}_{\text{std}}}(\tilde{x}))$.

D.3.2. TRADES

TRADES (Zhang et al., 2019) was proposed as a modification of the projected gradient adversarial training algorithm of (Madry et al., 2018). The robust loss is defined slightly differently–it -operates on the normalized logits, which can be thought of as probabilities of different labels. The TRADES loss minimizes the maximum KL divergence between the probability over labels for input x and a perturbaed input $\tilde{x} \in T(x)$. Setting the weights of the different loss of the general RST estimator (10) similar to RST+PG-AT above gives the following estimator.

$$\hat{\theta}_{\text{rst+trades}} := \arg\min_{\theta} \Bigg\{ \frac{(1-\lambda)}{n} \sum_{(x,y)\in[X_{\text{std}},y_{\text{std}}]} \ell(f_\theta(x),y) + \beta\, KL(p_\theta(x_{\text{adv}})||p_\theta(x)) + \frac{\lambda}{m}\sum_{i=1}^{m} \ell(f_\theta(\tilde{x}_i), f_{\hat{\theta}_{\text{std}}}(\tilde{x}_i)) + \beta\, KL(p_\theta(\tilde{x}_{\text{adv}\,i})||p_{\hat{\theta}_{\text{std}}}(\tilde{x}_i)) \Bigg\}. \tag{36}$$

In practice, x_{adv} and $\tilde{x}_{\text{adv}}$ are obtained by performing a few steps of projected gradient method on the respective KL divergence terms.

E. Experimental Details

E.1. Spline simulations

For spline simulations in Figure 2 and Figure 1, we implement the optimization of the standard and robust objectives using the basis described in (Friedman et al., 2001). The penalty matrix M computes second-order finite differences of the parameters θ. We solve the min-norm objective directly using CVXPY (Diamond & Boyd, 2016). Each point in Figure 1(a) represents the average standard error over 25 trials of randomly sampled training datasets between 22 and 1000 samples. Shaded regions represent 1 standard deviation.

E.2. RST experiments

We evaluate the performance of RST applied to ℓ_∞ adversarial perturbations, adversarial rotations, and random rotations.

E.2.1. SUBSAMPLING CIFAR-10

We augment with ℓ_∞ adversarial perturbations of various sizes. In each epoch, we find the augmented examples via Projected Gradient Ascent on the multiclass logistic loss (cross-entropy loss) of the incorrect class. Training the augmented estimator in this setup uses essentially the adversarial training procedure of (Madry et al., 2018), with equal weight on both the "clean" and adversarial examples during training.

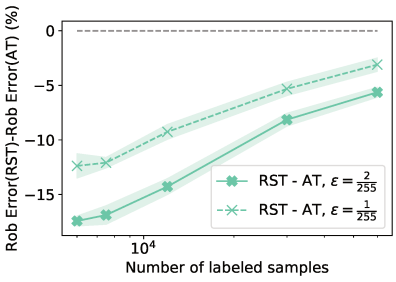
We compare the standard error of the augmented estimator with an estimator trained using RST. We apply RST to adversarial training algorithms in CIFAR-10 using 500k unlabeled examples sourced from Tiny Images, as in (Carmon et al., 2019).

We use Wide ResNet 40-2 models (Zagoruyko & Komodakis, 2016) while varying the number of samples in CIFAR-10. We sub-sample CIFAR-10 by factors of $\{1, 2, 5, 8, 10, 20, 40\}$ in Figure 1(a) and $\{1, 2, 5, 8, 10\}$ in Figure 1(b). We report results averaged from 2 trials for each sub-sample factor. All models are trained for 200 epochs with respect to the size of the labeled training dataset and all achieve almost 100% standard and robust training accuracy.

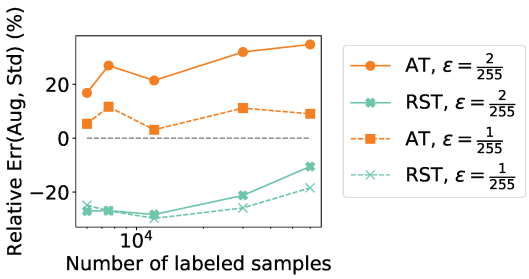
We evaluate the robustness of models to the strong PGD-attack with 40 steps and 5 restarts. In Figure 1(b), we used a simple heuristic to set the regularization strength on unlabeled data λ in Equation (35) to be $\lambda = \min(0.9, p)$ where $p \in [0, 1]$ is the fraction of the original CIFAR-10 dataset sampled. We set $\beta = 0.5$. Intuitively, we give more weight to the unlabeled data when the original dataset is larger, meaning that the standard estimator produces more accurate pseudo-labels.

Figure 9 shows that the robust accuracy of the RST model improves about 5-15% percentage points above the robust model (trained using PGD adversarial training) for all subsamples, including the full dataset (Tables 2,3).

We use a smaller model due to computational constraints enforced by adversarial training. Since the model is small, we could only fit adversarially augmented examples with small $\epsilon = 2/255$, while existing baselines use $\epsilon = 8/255$. Note that even for $\epsilon = 2/255$, adversarial data augmentation leads to an increase in standard error. We show that RST can fix this. While ensuring models are robust is an important goal in itself, in this work, we view adversarial training through the lens of covariate-shifted data augmentation and study how to use augmented data without increasing standard error. We show that



(a) Robust error, CIFAR-10



(b) Relative standard error, CIFAR-10

Figure 9. **(a)** Difference in robust error between the RST adversarial training model and the vanilla adversarial training (AT) model for CIFAR-10. RST improves upon the robust error of the AT model by approximately a 15% percentage point increase for small subsamples and 5% percentage point increase for larger subsamples of CIFAR-10. **(b)** Relative difference in standard error between augmented estimators (the RST model and the AT model) and the standard estimator on CIFAR-10. We achieve up to 20% better standard error than the standard model for small subsamples.

	Standard	AT	RST+AT
Standard Acc	94.63%	94.15%	**95.58%**
Robust Acc ($\epsilon = 1/255$)	-	85.59%	**88.74%**

Table 2. Test accuracies for the standard, vanilla adversarial training (AT), and AT with RST for $\epsilon = 1/255$ on the full CIFAR-10 dataset. Accuracies are averaged over two trials. The robust accuracy of the standard model is near 0%.

RST preserves the other benefits of some kinds of data augmentation like increased robustness to adversarial examples.

E.2.2. ℓ_∞ ADVERSARIAL PERTURBATIONS

In Table 1, we evaluate RST applied to PGD and TRADES adversarial training. The models are trained on the full CIFAR-10 dataset, and models which use unlabeled data (self-training and RST) also use 500k unlabeled examples from Tiny Images. All models except the Interpolated AT and Neural Architecture Search model use the same base model WideResNet 28-10. To evaluate robust accuracy, we use a strong PGD-attack with 40 steps and 5 restarts against ℓ_∞ perturbations of size $8/255$. For RST models, we set $\beta = 0.5$ in Equation (35) and Equation (36), following the heuristic $\lambda = \min(0.9, p)$ with $p = 1$ since we use the entire labeled trainign set. We train for 200 epochs such that 100% training standard accuracy is attained.

E.2.3. ADVERSARIAL AND RANDOM ROTATION/TRANSLATIONS

In Table 1 (right), we use RST for adversarial and random rotation/translations, denoting these transformations as x_{adv} in Equation (35). The attack model is a grid of rotations of up to 30 degrees and translations of up to $\sim 10\%$ of the image size. The grid consists of 31 linearly spaced rotations and 5 linearly spaced translations in both dimensions. The Worst-of-10 model samples 10 uniformly random transformations of each input and augment with the one where the model performs the worst (causes an incorrect prediction, if it exists). The Random model samples 1 random transformation as the augmented input. All models (besides cited models) use the WRN-40-2 architecture and are trained for 200 epochs. We use the same hyperparameters λ, β as in E.2.2 for Equation (35).

F. Comparison to standard self-training algorithms

The main objective of RST is to allow to perform robust training without sacrificing standard accuracy. This is done by regularizing an augmented estimator to provide labels close to a standard estimator on the unlabeled data. This is closely related to but different two broad kinds of semi-supervised learning.

1. Self-training (pseudo-labeling): Classical self-training does not deal with data augmentation or robustness. We view RST as a a generalization of self-training in the context of data augmentations. Here the pseudolabels are generated by a standard non-augmented estimator that is *not* trained on the labeled augmented points. In contrast, standard

	Standard	AT	RST+AT
Standard Acc	94.63%	92.69%	**95.15%**
Robust Acc ($\epsilon = 2/255$)	-	77.87%	**83.50%**

Table 3. Test accuracies for the standard, vanilla adversarial training (AT), and AT with RST for $\epsilon = 2/255$ on the full CIFAR-10 dataset. Accuracies are averaged over two trials. The robust test accuracy of the standard model is near 0%.

self-training would just use all labeled data to generate pseudo-labels. However, since some augmentations cause a drop in standard accuracy, and hence this would generate worse pseudo-labels than RST.

2. Robust consistency training: Another popular semi-supervised learning strategy is based on enforcing consistency in a model's predictions across various perturbations of the unlabeled data (Miyato et al., 2018; Xie et al., 2019; Sajjadi et al., 2016; Laine & Aila, 2017)). RST is similar in spirit, but has an additional crucial component. We generate pseudo-labels first by performing standard training, and rather than enforcing simply consistency across perturbations, RST enforces that the unlabeled data and perturbations are matched with the pseudo-labels generated.

G. Minimum ℓ_1-norm problem where data augmentation hurts standard error

We present a problem where data augmentation increases standard error for minimum ℓ_1-norm estimators, showing that the phenomenon is not special to minimum Mahalanobis norm estimators.

G.1. Setup in 3 dimensions

Define the minimum ℓ_1-norm estimators

$$\hat{\theta}_{\text{std}} = \arg\min_{\theta} \left\{ \|\theta\|_1 : X_{\text{std}}\theta = y_{\text{std}} \right\}$$
$$\hat{\theta}_{\text{aug}} = \arg\min_{\theta} \left\{ \|\theta\|_1 : X_{\text{std}}\theta = y_{\text{std}}, X_{\text{ext}}\theta = y_{\text{ext}} \right\}.$$

We begin with a 3-dimensional construction and then increase the number of dimensions. Let the domain of possible values be $\mathcal{X} = \{\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}\}$ where

$$\mathbf{x_1} = [1+\delta, 1, 0], \quad \mathbf{x_2} = [0, 1, 1+\delta], \quad \mathbf{x_3} = [1+\delta, 0, 1].$$

Define the data distribution through the generative process for the random feature vector $\mathbf{x}$

$$\mathbf{x} = \begin{cases} \mathbf{x_1} & \text{w.p. } 1-p \\ \mathbf{x_2} & \text{w.p. } \epsilon \\ \mathbf{x_3} & \text{w.p. } p-\epsilon \end{cases}$$

where $0 < \delta < 1$ and $\epsilon > 0$. Define the optimal linear predictor $\theta^\star = \mathbf{1}$ to be the all-ones vector, such that in all cases, $\mathbf{x}^\top \theta^\star = 2 + \delta$. We define the consistent perturbations as

$$T(x) = \begin{cases} \{\mathbf{x_1}, \mathbf{x_2}\} & x \in \{\mathbf{x_1}, \mathbf{x_2}\} \\ \{\mathbf{x_3}\} & o.w. \end{cases}$$

The augmented estimator will add all possible consistent perturbations of the training set as extra data X_{ext}. For example, if $\mathbf{x_1}$ is in the training set, then the augmented estimator will add $\mathbf{x_2}$ as extra data since $\mathbf{x_2} \in T(\mathbf{x_1})$. The standard error is measured by mean squared error.

We give some intuition for how augmentation can hurt standard error in this 3-dimensional example. Define E_1 to be the event that we draw n samples with value $\mathbf{x_1}$. Given E_1, the standard and augmented estimators are

$$\hat{\theta}_{\text{std}} = \left[\frac{2+\delta}{1+\delta}, 0, 0 \right], \quad \hat{\theta}_{\text{aug}} = [0, 2+\delta, 0]. \tag{37}$$

Note that the $\hat{\theta}_{\text{aug}}$ has slightly higher norm ($\|\hat{\theta}_{\text{aug}}\|_1 = 2 + \delta > \frac{2+\delta}{1+\delta} = \|\hat{\theta}_{\text{std}}\|_1$). Since $\mathbf{x_3}^\top \hat{\theta}_{\text{aug}} = 0$ in this case, the squared error of $\hat{\theta}_{\text{aug}}$ wrt to $\mathbf{x_3}$ is $(\mathbf{x_3}^\top \hat{\theta}_{\text{aug}} - 2 + \delta)^2 = (2+\delta)^2$. The standard estimator fits $\mathbf{x_3}$ perfectly, but has high error on $\mathbf{x_2}$. If the probability of E_1 occurring is high and the probability of $\mathbf{x_3}$ is higher relative to $\mathbf{x_2}$, then the $\hat{\theta}_{\text{aug}}$ will have high standard error relative to $\hat{\theta}_{\text{std}}$. Here, due to the inductive bias that minimizes the ℓ_1 norm, certain augmentations can cause large changes in the sparsity pattern of the solution, drastically affecting the error. Furthermore, the optimal solution $\theta^\star$ is quite large with respect to the ℓ_1 norm, satisfying the conditions of Proposition 1 in spirit and suggesting that the ℓ_1 inductive bias (promoting sparsity) is mismatched with the problem.

G.2. Construction for general d

We construct the example by sampling $\mathbf{x}$ in 3 dimensions and then repeating the vector d times. In particular, the samples are realizations of the random vector $[\mathbf{x}; \mathbf{x}; \mathbf{x}; \ldots; \mathbf{x}]$ which have dimension $3d$ and every block of 3 coordinates have the same values. Under this setup, we can show that there is a family of problems such that the difference between standard errors of the augmented and standard estimators grows to infinity as $d, n \to \infty$.

Theorem 4. *Let the setting be defined as above, where the dimension d and number of samples n are such that $n/d \to \gamma$ approaches a constant. Let $p = 1/d^2$, $\epsilon = 1/d^3$, and δ be a constant. Then the ratio between standard errors of the augmented and standard estimators grows as*

$$\frac{L_{std}(\hat{\theta}_{aug})}{L_{std}(\hat{\theta}_{std})} = \Omega(d) \tag{38}$$

as $d, n \to \infty$.

Proof. We define an event where the augmented estimator has high error relative to the standard estimator and bound the ratio between the standard errors of the standard and augmented estimators given this event. Define E_1 as the event that we have n samples where all samples are $[\mathbf{x_1}; \mathbf{x_1}; \ldots; \mathbf{x_1}]$. The standard and augmented estimators are the corresponding repeated versions

$$\hat{\theta}_{\text{std}} = \left[\frac{2+\delta}{1+\delta}, 0, 0, \ldots, \frac{2+\delta}{1+\delta}, 0, 0\right], \quad \hat{\theta}_{\text{aug}} = [0, 2+\delta, 0, \ldots, 0, 2+\delta, 0]. \tag{39}$$

The event E_1 occurs with probability $(1-p)^n + (p-\epsilon)^n$. It is straightforward to verify that the respective standard errors are

$$L_{\text{std}}(\hat{\theta}_{\text{std}} \mid E_1) = \epsilon d^2 (2+\delta)^2, \quad L_{\text{std}}(\hat{\theta}_{\text{aug}} \mid E_1) = (p-\epsilon) d^2 (2+\delta)^2$$

and that the ratio between standard errors is

$$\frac{L_{\text{std}}(\hat{\theta}_{\text{aug}} \mid E_1)}{L_{\text{std}}(\hat{\theta}_{\text{std}} \mid E_1)} = \frac{p-\epsilon}{\epsilon}.$$

The ratio between standard errors is bounded by

$$\begin{aligned}
\frac{L_{\text{std}}(\hat{\theta}_{\text{aug}})}{L_{\text{std}}(\hat{\theta}_{\text{std}})} &= \sum_{E \in \{E_1, E_1^c\}} P(E) \frac{L_{\text{std}}(\hat{\theta}_{\text{aug}} \mid E)}{L_{\text{std}}(\hat{\theta}_{\text{std}} \mid E)} \\
&> P(E_1) \frac{L_{\text{std}}(\hat{\theta}_{\text{aug}} \mid E_1)}{L_{\text{std}}(\hat{\theta}_{\text{std}} \mid E_1)} \\
&= ((1-p)^n + (p-\epsilon)^n)(\frac{p-\epsilon}{\epsilon}) \\
&> (1-p)^n (d-1) \\
&\geq (1 - \frac{n}{d^3})(d-1) \\
&= d - \frac{n}{d^2} - 1 + \frac{n}{d^3} = \Omega(d)
\end{aligned}$$

as $n, d \to \infty$, where we used Bernoulli's inequality in the second to last step. □