Adversarial Training Can Hurt Generalization

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

Abstract

While adversarial training can improve robust accuracy (against an adversary), it sometimes hurts standard accuracy (when there is no adversary). Previous work has studied this tradeoff between standard and robust accuracy, but only in the setting where no predictor performs well on both objectives in the infinite data limit. In this paper, we show that even when the optimal predictor with infinite data performs well on both objectives, a tradeoff can still manifest itself with finite data. Furthermore, since our construction is based on a convex learning problem, we rule out optimization concerns, thus laying bare a fundamental tension between robustness and generalization. Finally, we show that robust self-training mostly eliminates this tradeoff by leveraging unlabeled data.

1. Introduction

Neural networks trained using standard training have very low accuracies on perturbed inputs commonly referred to as *adversarial examples* [12]. Even though adversarial training [4, 6] can be effective at improving the accuracy on such examples (*robust accuracy*), these modified training methods decrease accuracy on natural unperturbed inputs (*standard accuracy*) [6, 19]. Table 1 shows the discrepancy between standard and adversarial training on CIFAR-10. While adversarial training improves robust accuracy from 3.5% to 45.8%, standard accuracy drops from 95.2% to 87.3%.

One explanation for a tradeoff is that the standard and robust objectives are fundamentally at conflict. Along these lines, Tsipras et al. [14] and Zhang et al. [19] construct learning problems where the perturbations can change the output of the Bayes estimator. Thus no predictor can achieve both optimal standard accuracy and robust accuracy even in the *infinite data limit*. However, we typically consider perturbations (such as imperceptible ℓ_∞ perturbations) which do not change the output of the Bayes estimator, so that a predictor with both optimal standard and high robust accuracy exists.

	Standard training	**Adversarial training**
Robust test	3.5%	45.8%
Robust train	-	100%
Standard test	95.2%	87.3%
Standard train	100%	100%

Table 1. Train and test accuracies standard and adversarially-trained models on CIFAR-10. Both have 100% training accuracy but very different test accuracies. In particular, adversarial training causes worse standard generalization.

Another explanation could be that the hypothesis class is not rich enough to contain predictors that have optimal standard and high robust accuracy, even if they exist [9]. However, Table 1 shows that adversarial training achieves 100% standard and robust accuracy on the training set, suggesting that the hypothesis class is expressive enough in practice.

Having ruled out a conflict in the objectives and expressivity issues, Table 1 suggests that the tradeoff stems from the worse generalization of adversarial training either due to (i) the statistical properties of the robust objective or (ii) the dynamics of optimizing the robust objective on neural networks. In an attempt to disentangle optimization and statistics, we ask *does the tradeoff indeed disappear if we rule out optimization issues?* After all, from a statistical perspective, the robust objective adds information (constraints on the outputs of perturbations) which should intuitively aid generalization, similar to Lasso regression which enforces sparsity [13].

Contributions. We answer the above question negatively by constructing a learning problem with a *convex loss* where adversarial training hurts generalization even when the optimal predictor has both optimal standard and robust accuracy. Convexity rules out optimization issues, revealing a fundamental statistical explanation for why adversarial training requires more samples to obtain high standard accuracy. Furthermore, we show that we can eliminate the tradeoff in our constructed problem using the recently-proposed robust self-training [15, 1, 8, 18] on additional unlabeled data.

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

ICML 2019 Workshop on Identifying and Understanding Deep Learning Phenomena, Long Beach, California, 2019. Copyright 2019 by the author(s).

In an attempt to understand how predictive this example is of practice, we subsample CIFAR-10 and visualize trends in the performance of standard and adversarially trained models with varying training sample sizes. We observe that the gap between the accuracies of standard and adversarial training decreases with larger sample size, mirroring the trends observed in our constructed problem. Recent results from [1] show that, similarly to our constructed setting, robust self-training also helps to mitigate the trade-off in CIFAR-10.

Standard vs. robust generalization. Recent work [11, 16, 5, 7] has focused on the sample complexity of learning a predictor that has high robust accuracy (robust generalization), a *different objective*. In contrast, we study the finite sample behavior of adversarially trained predictors on the standard learning objective (standard generalization), and show that adversarial training as a particular training procedure could require more samples to attain high standard accuracy.

2. Convex learning problem: the staircase

We construct a learning problem with the following properties. First, fitting the majority of the distribution is statistically easy—it can be done with a *simple* predictor. Second, perturbations of these majority points are low in probability and require *complex* predictors to be fit. These two ingredients cause standard estimators to perform better than their adversarially trained robust counterparts with a few samples. Standard training only fits the training points, which can be done with a simple estimator that generalizes well; adversarial training encourages fitting perturbations of the training points making the estimator complex and generalize poorly.

2.1. General setup

We consider mapping $x \in \mathcal{X} \subset \mathbb{R}$ to $y \in \mathbb{R}$ where (x, y) is a sample from the joint distribution $\mathbb{P}$ and conditional densities exist. We denote by $\mathbb{P}_x$ the marginal distribution on $\mathcal{X}$. We generate the data as $y = f^\star(x) + \sigma v_i$ where $v_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $f^\star : \mathcal{X} \to \mathbb{R}$. For an example (x, y), we measure robustness of a predictor with respect to an *invariance set* $B(x)$ that contains the set of inputs on which the predictor is expected to match the target y.

The central premise of this work is that the optimal predictor is robust. In our construction, we let $f^\star$ be robust by enforcing the invariance property (see Appendix A)

$$f(x) = f(\tilde{x}), \quad \forall \tilde{x} \in B(x). \tag{1}$$

Given training data consisting of n i.i.d. samples $(x_i, y_i) \sim \mathbb{P}$, our goal is to learn a predictor $f \in \mathcal{F}$. We assume that the hypothesis class $\mathcal{F}$ contains $f^\star$ and consider the squared loss. Standard training simply minimizes the empirical risk over the training points. Robust training seeks to enforce invariance to perturbations of training points by penalizing the worst-case loss over the invariance set $B(x_i)$ with respect to target y_i. We consider regularized estimation and obtain the following standard and robust (adversarially trained) estimators for sample size n:

$$\hat{f}_n^{\text{std}} \in \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|^2, \tag{2}$$

$$\hat{f}_n^{\text{rob}} \in \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{i=1}^{n} \max_{\tilde{x}_i \in B(x_i)} (f(\tilde{x}_i) - y_i)^2 + \lambda \|f\|^2. \tag{3}$$

We construct a $\mathbb{P}$ and $f^\star$ such that both estimators above converge to $f^\star$, but such that the error of the robust estimator $\hat{f}_n^{\text{rob}}$ is larger than that of $\hat{f}_n^{\text{std}}$ for small sample size n.

2.2. Construction

In our construction, we consider linear predictors as "simple" predictors that generalize well and *staircase* predictors as "complex" predictors that generalize poorly (Figure 1(a)).

Input distribution. In order to satisfy the property that a simple predictor fits most of the distribution, we define $f^\star$ to be linear on the set $\mathcal{X}_{\text{line}} \subseteq \mathcal{X}$, where

$$\mathcal{X}_{\text{line}} = \{0, 1, 2, \ldots, s-1\}, \quad \mathbb{P}_x(\mathcal{X}_{\text{line}}) = 1 - \delta, \tag{4}$$

for parameters $\delta \in [0, 1]$ and a positive integer s. Any predictor that fits points in $\mathcal{X}_{\text{line}}$ will have low (but not optimal) standard error when δ is small.

Perturbations. We now define the perturbations such that that fitting perturbations of the majority of the distribution requires complex predictors. We can obtain a staircase by flattening out the region around the points in $\mathcal{X}_{\text{line}}$ locally (Figure 1(a)). This motivates our construction where we treat points in $\mathcal{X}_{\text{line}}$ as anchor points and the set $\mathcal{X}_{\text{line}}^c$ as local perturbations of these points: $x \pm \epsilon$ for $x \in \mathcal{X}_{\text{line}}$. This is a simpler version of the commonly studied ℓ_∞ perturbations in computer vision. For a point that is not an anchor point, we define $B(x)$ as the invariance set of the closest anchor point $\lfloor x \rceil$. Formally, for some $\epsilon \in (0, \frac{1}{2})$,

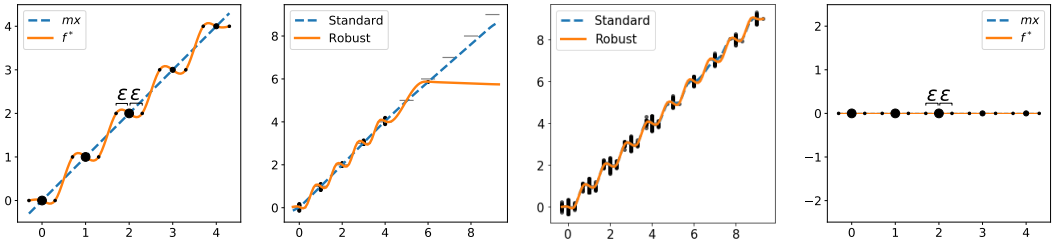
$$B(x) = \{\lfloor x \rceil, \lfloor x \rceil + \epsilon, \lfloor x \rceil - \epsilon\}. \tag{5}$$

Output distribution. For any point in the support $\mathcal{X}$,

$$f^\star(x) = m \lfloor x \rceil, \forall x \in \mathcal{X}, \tag{6}$$

for some parameter m. Setting the slope as $m = 1$ makes $f^\star$ resemble a staircase. Such an $f^\star$ satisfies the invariance property (1) that ensures that the optimal predictor for standard error is also robust. Note that $f^\star(x) = mx$ (a simple linear function) when restricted to x in $\mathcal{X}_{\text{line}}$. Note also that the invariance sets $B(x)$ are disjoint. This is in contrast to the example in [19], where any invariant function is also globally constant. Our construction allows a non-trivial robust and accurate estimator.

We generate the output by adding Gaussian noise to the optimal predictor $f^\star$, i.e., $y = f^\star(x) + \sigma v_i$ where $v_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$.



(a) Slope $m=1$ (b) Small sample size (c) Large sample size (d) Slope $m=0$

Figure 1. **(a):** An illustration of our convex problem with slope $m = 1$, with size of the circles proportional to probability under the data distribution. The dashed blue line shows a simple linear predictor that has low test error but not robust to perturbations to nearby low-probability points, while the solid orange line shows the complex optimal predictor $f^\star$ that is both robust and accurate. **(b):** With small sample size ($n=40$), any robust predictor that fits the sets $B(x)$ is forced to be a staircase that generalizes poorly. **(c):** With large sample size ($n=25000$), the training set contains all the points from $\mathcal{X}_{\text{line}}$ and the robust predictor is close to $f^\star$ by enforcing the right invariances. The standard predictor also has low error, but higher than the robust predictor. **(d):** An illustration of our convex problem when the slope $m=0$. The optimal predictor $f^\star$ that is robust is a simple linear function. This setting sees no tradeoff for any sample size.

2.3. Simulations

We empirically validate the intuition that the staircase problem is sensitive to robust training by simulating training with various sample sizes and comparing the test MSE of the standard and robust estimators (2) and (3). We report final test errors here; trends in generalization gap (difference between train and test error) are nearly identical. See Appendix D for more details.

Figure 2 shows the difference in test errors of the two estimators. For each sample size n, we compare the standard and robust estimators by performing a grid search over regularization parameters λ that individually minimize the test MSE of each estimator. With few samples, most training samples are from $\mathcal{X}_{\text{line}}$ and standard training learns a simple linear predictor that fits all of $\mathcal{X}_{\text{line}}$. On the other hand, robust estimators fit the low probability perturbations $\mathcal{X}_{\text{line}}^c$, leading to staircases that generalize poorly. Figure 1(b) visualizes the two estimators for small samples. However, as we increase the size of the training set, the training set contains all points from $\mathcal{X}_{\text{line}}$, and robust estimators also generalize well despite being more complex. Furthermore, in this regime, robust estimators indeed see the expected "regularization" benefit where the robust objective helps fit points in the low probability regions $\mathcal{X}_{\text{line}}^c$, even when they are not yet sampled in the training points. In general, we see that robust training has higher test error with a small sample size, but the difference in the test error of standard and robust estimators decreases as sample size increases, and robust training eventually obtains lower test error.

Another common approach to encoding invariances is data augmentation, where perturbations are *sampled* from $B(x)$ and added to the dataset. Data augmentation is less demanding than adversarial training which minimizes loss on the *worst-case* point within the invariance set. We find that for our staircase example, an estimator trained even with the less demanding data augmentation sees a similar tradeoff with small training sets, due to increased complexity of the augmented estimator.

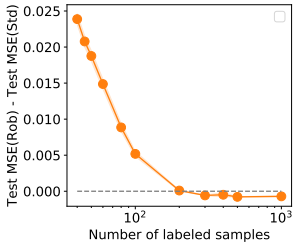
2.4. Robust self-training mostly eliminates the tradeoff

Section 2.3 shows that the gap between the standard errors of robust and standard estimators decreases as training sample size increases. Moreover, if we obtained training points spanning $\mathcal{X}_{\text{line}}$, then the robust estimator (staircase) would also generalize well and have lower error than the standard estimator. Thus, a natural strategy to eliminate the tradeoff is to sample more training points. In fact, we do not need additional labels for the points on $\mathcal{X}_{\text{line}}$—a standard trained estimator fits points on $\mathcal{X}_{\text{line}}$ with just a few labels, and can be used to generate labels on additional unlabeled points. Recent works have proposed robust self-training (RST) to leverage unlabeled data for robustness [10, 1, 15, 8, 18]. RST is a robust variant of the popular self-training algorithm for semi-supervised learning [10], which uses a standard estimator trained on a few labels to generate psuedo-labels for unlabeled data as described above. See Appendix C for details on RST.

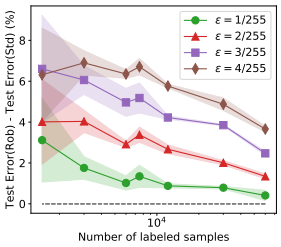
For the staircase problem ($m=1$), RST mostly eliminates the tradeoff and achieves similar test error to standard training (while also being robust, see Appendix C.2) as shown in Figure 2.

3. Experiments on CIFAR-10

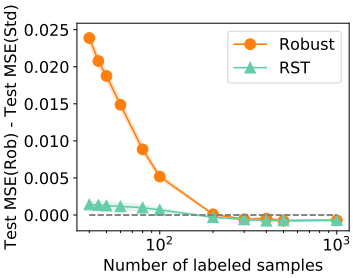
In our staircase problem from Section 2, robust estimators perform worse on the standard objective because these predictors are more complex, thereby generalizing poorly. Does this also explain the drop in standard accuracy we see for adversarially trained models on real datasets like CIFAR-10?



(a) Staircase, $m=1$



(b) WRN40-2 on CIFAR-10



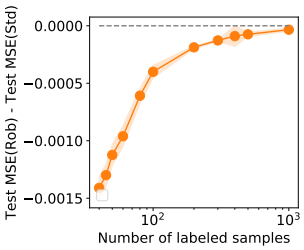
(c) Staircase ($m=1$): RST vs. Robust

Figure 2. Difference between test errors (robust - standard) as a function of the # of training samples n. For each n, we choose the best regularization parameter λ for each of robust and standard training and plot the difference. Positive numbers show that the robust estimator has higher MSE than the standard estimator. **(a)** For the staircase problem with slope $m=1$, we see that for small n, test loss of the robust estimator is larger. As n increases, the gap closes, and eventually the robust estimator has smaller MSE. **(b)** On subsampling CIFAR-10, we see that the gap between test errors (%) of standard and adversarially trained models decreases as the number of samples increases, just like the staircase construction in (a). Extrapolating, the gap should close as we have more samples. **(c)** Robust self-training (RST), using 1000 additional unlabeled points, achieves comparable test MSE to standard training (with the same amount of labeled data) and mostly eliminates the tradeoff seen in robust training. The shaded regions represent 1 STD.

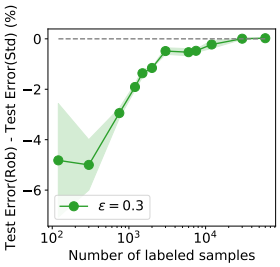
We subsample CIFAR-10 by various amounts to study the effect of sample size on the standard test errors of standard and robust models. To train a robust model, we use the adversarial training procedure from [6] against ℓ_∞ perturbations of varying sizes (see Figure 2). The gap in the errors of the standard and adversarially trained models decreases as sample size increases, mirroring the trends in the staircase problem. Extrapolating the trends, more training data should eliminate the tradeoff in CIFAR-10. Similarly to the staircase example, [1] showed that robust self-training with additional unlabeled data improves robust accuracy and standard accuracy in CIFAR-10. See Appendix C for more details.

4. Adversarial training can also help

One of the key ingredients that causes the tradeoff in the staircase problem is the complexity of robust predictors.



(a) Staircase, $m=0$



(b) Small CNN on MNIST

Figure 3. Difference between test errors (robust - standard) as a function of the # of training samples n. For each n, we choose the best regularization parameter λ for each of robust and standard training and take the difference. Negative numbers mean that robust training has a lower test MSE than standard training. **(a)** In the staircase problem with slope $m=0$, the robust estimator consistently outperforms the standard estimator, showing a regularization benefit. **(b)** On MNIST , the adversarially trained model has lower test error (%) than the standard model. The difference in test errors is largest for small sample sizes and closes with more training samples. Shaded regions represent 1 STD.

If we change our construction such that robust predictors are also simple, we see that adversarial training instead offers a regularization benefit. When $m=0$, the optimal predictor (which is robust) is linear (Figure 1(d)). We find that adversarial training has lower standard error by enforcing invariance on $B(x)$ making the robust estimator less sensitive to target noise (Figure 4(a)).

Similarly, on MNIST , the adversarially trained model has lower test error than standard trained model. As we increase the sample size, both standard and adversarially trained models converge to obtain same small test error. We remark that our observation on MNIST is contrary to that reported in [14], due to a different initialization that led to better optimization (see Appendix Section D.2).

5. Conclusion

In this work, we shed some light on the counter-intuitive phenomenon where enforcing invariance respected by the optimal function could actually degrade performance. Being invariant could require complex predictors and consequently more samples to generalize well. Our experiments support that the tradeoff between robustness and accuracy observed in practice is indeed due to insufficient samples and additional unlabeled data is sufficient to mitigate this tradeoff.

Acknowledgements

We are grateful to Tengyu Ma for several helpful discussions. This work was funded by an Open Philanthropy Project Award. AR was supported by Google Fellowship and Open Philanthropy AI Fellowship. FY was supported by the Institute for Theoretical Studies ETH Zurich and the Dr. Max Rössler and the Walter Haefner Foundation. FY and JCD were supported by the Office of Naval Research Young Investigator Award N00014-19-1-2288.

References

[1] Y. Carmon, A. Raghunathan, L. Schmidt, P. Liang, and J. C. Duchi. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019.

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

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

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

[5] J. Khim and P. Loh. Adversarial risk bounds for binary classification via function transformation. *arXiv preprint arXiv:1810.09519*, 2018.

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

[7] O. Montasser, S. Hanneke, and N. Srebro. VC classes are adversarially robustly learnable, but only improperly. *arXiv preprint arXiv:1902.04217*, 2019.

[8] A. Najafi, S. Maeda, M. Koyama, and T. Miyato. Robustness to adversarial perturbations in learning from incomplete data. *arXiv preprint arXiv:1905.13021*, 2019.

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

[10] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision*, 2005.

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

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

[13] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

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

[15] J. Uesato, J. Alayrac, P. Huang, R. Stanforth, A. Fawzi, and P. Kohli. Are labels required for improving adversarial robustness? *arXiv preprint arXiv:1905.13725*, 2019.

[16] D. Yin, K. Ramchandran, and P. Bartlett. Rademacher complexity for adversarially robust generalization. *arXiv preprint arXiv:1810.11914*, 2018.

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

[18] R. Zhai, T. Cai, D. He, C. Dan, K. He, J. Hopcroft, and L. Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019.

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

A. Consistency of robust and standard estimators

We show that the invariance condition (restated, (7)) is a sufficient condition for the minimizers of the standard and robust objectives under $\mathbb{P}$ in the infinite data limit to be the same.

$$f^\star(x) = f^\star(\tilde{x}) \;\; \forall \tilde{x} \in B(x), \tag{7}$$

for all $x \in \mathcal{X}$.

Recall that $y = f^\star(x) + \sigma v_i$ where $v_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, with $f^\star(x) = \mathbb{E}[y \,|\, x]$. Therefore, if $f^\star$ is in the hypothesis class $\mathcal{F}$, then $f^\star$ minimizes the standard objective for the square loss.

If both $\hat{f}_n^{\text{std}}$ (2) and $\hat{f}_n^{\text{rob}}$ (3) converge to the same Bayes optimal $f^\star$ as $n \to \infty$, we say that the two estimators $\hat{f}_n^{\text{std}}$ and $\hat{f}_n^{\text{rob}}$ are *consistent*. In this section, we show that the invariance condition (7) implies consistency of $\hat{f}_n^{\text{rob}}$ and $\hat{f}_n^{\text{std}}$.

Intuitively, from (7), since $f^\star$ is invariant for all x in $B(x)$, the maximum over $B(x)$ in the robust objective is achieved by the unperturbed input x (and also achieved by any other element of $B(x)$). Hence the standard and robust loss of $f^\star$ are equal. For any other predictor, the robust loss upper bounds the standard loss, which in turn is an upper bound on the standard loss of $f^\star$ (since $f^\star$ is Bayes optimal). Therefore $f^\star$ also obtains optimal robust loss and $\hat{f}_n^{\text{std}}$ and $\hat{f}_n^{\text{rob}}$ are consistent and converge to $f^\star$ with infinite data.

Formally, let ℓ be the square loss function, and the population loss be $\mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(f(x),y)]$. In this section, all expectations are taken over the joint distribution $\mathbb{P}$.

Theorem 1. *(Regression) Consider the minimizer of the standard population squared loss, $f^* = \operatorname{argmin}_f \mathbb{E}[\ell(f(x),y)]$ where $\ell(f(x), y) = (f(x) - y)^2$. Assuming (7) holds, we have that for any f, $\mathbb{E}[\max_{\tilde{x}\in B(x)} \ell(f(x), y)] \geq \mathbb{E}[\max_{\tilde{x}\in B(x)} \ell(f^*(x),y)]$, such that f^* is also optimal for the robust population squared loss.*

Proof. Note that the optimal standard model is the Bayes estimator, such that $f^*(x) = \mathbb{E}[y \,|\, x]$. Then by condition (7), $f^*(\tilde{x}) = \mathbb{E}[y \mid \tilde{x}] = \mathbb{E}[y \mid x] = f^*(x)$ for all $\tilde{x} \in B(x)$. Thus the robust objective for f^* is

$$\begin{aligned}\mathbb{E}[\max_{\tilde{x}\in B(x)} \ell(f^*(x),y)] &= \mathbb{E}\left[\max_{\tilde{x}\in B(x)} (\mathbb{E}[y \,|\, \tilde{x}] - y)^2\right] \\ &= \mathbb{E}\left[(\mathbb{E}[y \,|\, x] - y)^2\right] \\ &= \mathbb{E}[\ell(f^*(x),y)]\end{aligned}$$

where the first equality follows because f^* is the Bayes estimator and the second equality is from (7). Noting that for any classifier f, $\mathbb{E}[\max_{\tilde{x}\in B(x)} \ell(f(x),y)] \geq \mathbb{E}[\ell(f(x),y)] \geq \mathbb{E}[\ell(f^*(x),y)]$, the theorem statement follows. □

For the classification case, consistency requires label invariance, which is that

$$\operatorname*{argmax}_y p(y \,|\, x) = \operatorname*{argmax}_y p(y \,|\, \tilde{x}) \;\; \forall \tilde{x} \in B(x), \tag{8}$$

such that the adversary cannot change the label that achieves the maximum but can perturb the distribution.

The optimal standard classifier here is the Bayes optimal classifier $f_c^\star = \operatorname{argmax}_y p(y \mid x)$. Assuming that $f_c^\star = \operatorname{argmax}_y p(y \,|\, x)$ is in $\mathcal{F}$, then consistency follows by essentially the same argument as in the regression case.

Theorem 2. *(Classification) Consider the minimizer of the standard population 0-1 loss, $f_c^\star = \operatorname{argmin}_f \mathbb{E}[\ell(f(x),y)]$ where $\ell(f(x), y) = \mathbf{1}\{\operatorname{argmax}_j f(x)_j = y\}$. Assuming (8) holds, we have that for any f, $\mathbb{E}[\max_{\tilde{x}\in B(x)} \ell(f(x),y)] \geq \mathbb{E}[\max_{\tilde{x}\in B(x)} \ell(f_c^\star(x),y)]$, such that $f_c^\star$ is also optimal for the robust population 0-1 loss.*

Proof. Replacing $f^\star$ with $f_c^\star$ and $\ell(f(x), y)$ with the zero-one loss $\mathbf{1}\{\operatorname{argmax}_j f(x)_j = y\}$ in the proof of Theorem 1 gives the result. □

In our staircase problem, from (1), we assume that the target y is generated as follows: $y = f^\star(x) + \sigma v_i$ where $v_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$, we see that the points within an invariance sets $B(x)$ have the same target distribution (target distribution invariance).

$$f^\star(x) = f^\star(\tilde{x}) \;\; \forall \tilde{x} \in B(x) \tag{9}$$

$$\implies p(y \,|\, x) = p(y \,|\, \tilde{x}) \quad \forall \tilde{x} \in B(x), \tag{10}$$

for all $x \in \mathcal{X}$.

The target invariance condition above implies consistency in both the regression and classification case.

B. Convex staircase example

B.1. Data distribution

Distribution of $\mathcal{X}$. We focus on a 1-dimensional regression case. Let s be the total number of "stairs" in the staircase problem. Let $s_0 \leq s$ be the number of stairs that have a large weight in the data distribution. Define $\delta \in [0,1]$ to be the probability of sampling a perturbation point, i.e. $x \in \mathcal{X}_{\text{line}}^c$, which we will choose to be close to zero. The size of the perturbations is $\epsilon \in [0, \frac{1}{2})$, which is bounded by $\frac{1}{2}$ so that $\lfloor x \pm \epsilon \rceil = x$, for any $x \in \mathcal{X}_{\text{line}}$. The standard deviation of the noise in the targets is $\sigma > 0$. Finally, $m \in [0,1]$ is a parameter controlling the slope of the points in $\mathcal{X}_{\text{line}}$.

Let $w \in \Delta_s$ be a distribution over $\mathcal{X}_{\text{line}}$ where Δ_s is the probability simplex of dimension s. We define the data distribution with the following generative process for one sample x. First, sample a point i from $\mathcal{X}_{\text{line}}$ according to the categorical

distribution described by w, such that $i \sim \text{Categorical}(w)$. Second, sample x by perturbing i with probability δ such that

$$x = \begin{cases} i & \text{w.p. } 1-\delta \\ i-\epsilon & \text{w.p. } \delta/2 \\ i+\epsilon & \text{w.p. } \delta/2. \end{cases}$$

Note that this is just a formalization of the distribution described in Section 2. The sampled x is in $\mathcal{X}_{\text{line}}$ with probability $1-\delta$ and $\mathcal{X}_{\text{line}}^c$ with probability δ, where we choose δ to be small.

In addition, in order to exaggerate the difference between robust and standard estimators for small sample sizes, we set w such that the first s_0 stairs have the majority of probability mass. To achieve this, we set the unnormalized probabilities of w as

$$\hat{w}_j = \begin{cases} 1/s_0 & j < s_0 \\ 0.01 & j \geq s_0 \end{cases}$$

and define w by normalizing $w = \hat{w}/\sum_j \hat{w}_j$. For our examples, we fix $s_0 = 5$. In general, even though we can increase s to create versions of our example with more stairs, s_0 is fixed to highlight the bad extrapolation behavior of the robust estimator.

Distribution of $\mathcal{Y}$. We define the target distribution as $(Y \mid X = x) \sim \mathcal{N}(m\lfloor x \rceil, \sigma^2)$, where $\lfloor x \rceil$ rounds x to the nearest integer. The invariance sets are $B(x) = \{\lfloor x \rceil - \epsilon, \lfloor x \rceil, \lfloor x \rceil + \epsilon\}$. We define the distribution such that for any x, all points in $B(x)$ have the same mean target value $m\lfloor x \rceil$. See Figure 1 for an illustration.

Note that $B(x)$ is defined such that ((9)) holds, since for any $x_1, x_2 \in B(x)$, $\lfloor x_1 \rceil = \lfloor x_2 \rceil$ and thus $p(y \mid x_1) = p(y \mid x_2)$. The conditional distributions are defined since $p(\tilde{x}) > 0$ for any $\tilde{x} \in B(x)$.

B.2. Model

Our hypothesis class is the family of cubic B-splines as defined in [3]. 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 $\tau = [-\epsilon, 0, \epsilon, \ldots, (s-1)-\epsilon, s-1, (s-1)+\epsilon]$, which places a knot on every point on the support of $\mathcal{X}$. This ensures that the family is expressive enough to include $f^\star$, which is any function in $\mathcal{F}$ which satisfies $f^\star(x) = m\lfloor x \rceil$ for all x in $\mathcal{X}$. Cubic B-splines can be viewed as a kernel method with kernel feature map $\Phi : \mathcal{X} \to \mathbb{R}^{3s+2}$, where s is the number of stairs in the example.

For some regularization parameter $\lambda \geq 0$ we optimize with the penalized smoothing spline loss function over parameters θ,

$$\ell(f_\theta(x), y) = (y - f_\theta(x))^2 + \lambda \int (f_\theta''(t))^2 dt \tag{11}$$

$$= (y - \Phi(x)^T\theta)^2 + \lambda\theta^T\Omega\theta, \tag{12}$$

where $\Omega_{i,j} = \int \Phi''(t)_i \Phi''(t)_j dt$ measures smoothness in terms of the second derivative.With respect to the regularized objectives (2) and (3), the norm regularizer is $\|f\|^2 = \theta^T\Omega\theta$.

We implement the optimization of the standard and robust objectives using the basis described in [3]. The regularization penalty matrix Ω computes second-order finite differences of the parameters θ. Suppose we have n samples of training inputs $X = \{x_1, \ldots, x_n\}$ and targets $\mathbf{y} = \{y_1, \ldots, y_n\}$ drawn from $\mathbb{P}$. The standard spline objective solves the linear system

$$\hat{\theta}_{\text{std}} = (\Phi(X)^T\Phi(X) + \lambda\Omega)^{-1}\Phi(X)^T\mathbf{y},$$

where the i-th row of $\Phi(X) \in \mathbb{R}^{n \times (3s+2)}$ is $\Phi(x_i)$. The standard estimator is then $\hat{f}_n^{\text{std}}(x) = \Phi(x)^T\hat{\theta}_{\text{std}}$. We solve the robust objective directly as a pointwise maximum of squared losses over the invariance sets (which is still convex) using CVXPY [2].

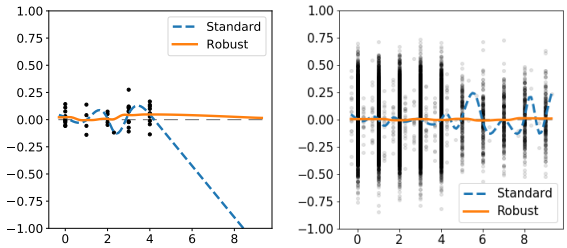
B.3. Role of different parameters

To construct an example where robustness hurts generalization, the main parameters needed are that the slope m is large and that the probability δ of drawing samples from perturbation points $\mathcal{X}_{\text{line}}^c$ is small. When slope m is large, the complexity of the true function increases such that good generalization requires more samples. A small δ ensures that a low-norm linear solution has low test error. This example is insensitive to whether there is label noise, meaning that $\sigma = 0$ is sufficient to observe that robustness hurts generalization.

If $m \approx 0$, then the complexity of the true function is low and we observe that robustness helps generalization. In contrast, this example relies on the fact that there is label noise ($\sigma > 0$) so that the noise-cancelling effect of robust training improves generalization. In the absence of noise, robustness neither hurts nor helps generalization since both the robust and standard estimators converge to the true function ($f^*(x) = 0$) with only one sample.

B.4. Plots of other values

We show plots for a variety of quantities against number of samples n. For each n, we pick the best regularization parameter λ with respect to standard test MSE individually for robust and standard training. in the $m = 1$ (robustness hurts) and $m = 0$ (robustness helps) cases, with all the same parameters as before. In both cases, the test MSE and generalization gap (difference between training MSE and test MSE) are almost identical due to robust and standard training having similar training errors. In the $m = 1$ case where robustness hurts (Figure 6), robust training finds higher norm estimators for all sample sizes. With enough samples, standard training begins to increase the norm of its solution as it starts to converge to the true function (which is complex) and the robust train MSE starts to drop accordingly.



(a) Small sample ($n = 40$) (b) Large sample ($n = 25000$)

Figure 4. **Left:** With small samples, the standard solution may overfit to noise, while adversarial training has a noise cancelling effect. **Right:** With large samples, both the robust and standard predictors have low test error, but the standard predictor is still more susceptible to noise.

In the $m = 0$ case where robustness helps (Figure 7), the optimal predictor is the line $f(x) = 0$, which has 0 norm. The robust estimator has consistently low norm. With small sample size, the standard estimator has low norm but has high test MSE. This happens when the standard estimator is close to linear (has low norm), but the estimator has the wrong slope, causing high test MSE. However, in the infinite data limit, both standard and robust estimators converge to the optimal solution.

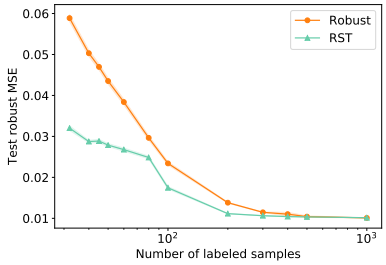
C. Robust self-training algorithm

We describe the robust self-training procedure, which performs robust training on a dataset augmented with unlabeled data. The targets for the unlabeled data are generated from a standard estimator trained on the labeled training data. Since the standard estimator has good standard generalization, the generated targets for the unlabeled data have low error on expectation. Robust training on the augmented dataset seeks to improve both the standard and robust test error of robust training (over just the labeled training data). Intuitively, robust self-training achieves these gains by mimicking the standard estimator on more of the data distribution (by using unlabeled data) while also optimizing the robust objective.

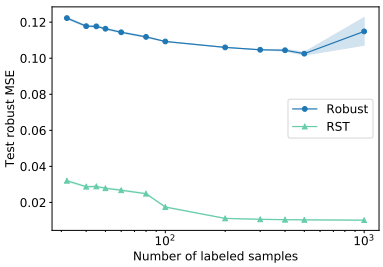
In robust self-training, we are given n samples of training inputs $X = \{x_1, \ldots, x_n\}$ and targets $\mathbf{y} = \{y_1, \ldots, y_n\}$ drawn from $\mathbb{P}$. Suppose that we have additional m unlabeled samples X_u drawn from $\mathbb{P}_x$. Robust self-training uses the following steps for a given regularization λ:

1. Compute the standard estimator $\hat{f}_n^{\text{std}}$ (2) on the labeled data $(X, \mathbf{y})$ with regularization parameter λ.

2. Generate pseudo-targets $\mathbf{y}_u = \hat{f}_n^{\text{std}}(X_u)$ by evaluating the standard estimator obtained above on the unlabeled data X_u.



(a) Robust training vs. RST



(b) Standard training vs. RST

Figure 5. Robust self-training (RST) improves test robust MSE (not just standard test MSE) over both standard and robust training. For each n, the regularization parameter λ is chosen with respect to the best test MSE over a grid search for each of robust, RST, and standard training. **(a)** shows that robust self-training improves robust error over robust training. **(b)** confirms that robust self-training also improves robust test error over standard training.

3. Construct an augmented dataset $X_{\text{aug}} = X \cup X_u$, $\mathbf{y}_{\text{aug}} = \mathbf{y} \cup \mathbf{y}_u$.

4. Return a robust estimator $\hat{f}_n^{\text{rob}}$ (3) with the augmented dataset $(X_{\text{aug}}, \mathbf{y}_{\text{aug}})$ as training data.

C.1. Results on CIFAR-10

We present relevant results from the recent work of [1] on robust self-training applied on CIFAR-10 augmented with unlabeled data in Table 2. The procedure employed in [1] is identical to the procedure describe above, using a modified version of adversarial training (TRADES) [19] as the robust estimator.

C.2. Robust self-training doesn't sacrifice robustness

In Section 2.4, we show that if we have access to additional unlabeled samples from the data distribution, robust self-training (RST) can mitigate the tradeoff in standard error between robust and standard estimators. It is important that we do not sacrifice robustness in order to have better standard error. Figure 5 shows that in the case where robustness hurts generalization in our convex construction ($m = 1$), RST improves over robust training not only in standard test error

	Standard training	Adversarial training	RST [1]
Robust test	3.5%	45.8%	62.5%
Standard test	95.2%	87.3%	89.7%

Table 2. Robust and standard accuracies for different training methods. Robust self-training (RST) leverages unlabeled data in addition to the CIFAR-10 training set to see an increase in both standard and robust accuracies over traditional adversarial training. To mitigate the tradeoff between robustness and accuracy, all we need is (possibly large amounts of) unlabeled data.

(Section 2.4), but also in robust test error. Therefore, by leveraging some unlabeled data, we can recover the standard generalization performance of standard training using RST while simultaneously improving robustness.

D. Experimental details

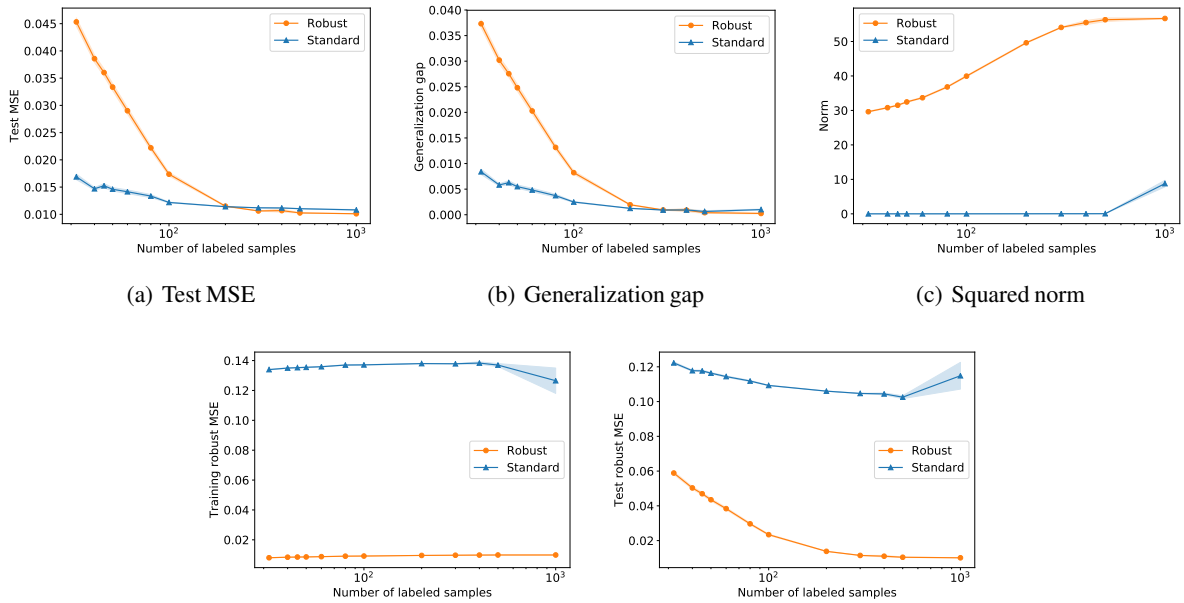
D.1. CIFAR-10

We train Wide ResNet 40-2 models [17] using standard and adversarial training while varying the number of samples in CIFAR-10. We sub-sample CIFAR-10 by factors of $\{1, 2, 5, 8, 10, 20, 40\}$. For sub-sample factors 1 to 20, we report results averaged from 2 trials each for standard and adversarial training. For sub-sample factors greater than 20, we average over 5 trials. We train adversarial models under the ℓ_∞ attack model with ℓ_∞-norm constraints of sizes $\epsilon = \{1/255, 2/255, 3/255, 4/255\}$ using PGD adversarial training [6]. The models are trained for 200 epochs using minibatched gradient descent with momentum, such that 100% standard training accuracy is achieved for both standard and adversarial models in all cases and $> 98\%$ adversarial training accuracy is achieved by adversarially trained models in most cases. We did not include reuslts for subsampling factors greater than 50, since the test accuracies are very low (20-50%). However, we note that for very small sample sizes (subsampling factor 500), the robust estimator can have slightly better test accuracy than the standard estimator. While this behavior is not captured by our example, we focus on capturing the observation that standard and robust test errors converge with more samples.

D.2. MNIST

The MNIST dataset consists of 60000 labeled examples of digits. We sub-sample the dataset by factors of $\{1, 2, 5, 8, 10, 20, 40, 50, 80, 200, 500\}$ and report results for a small 3-layer CNN averaged over 2 trials for each sub-sample factor. All models are trained for 200 epochs and achieve 100% standard training accuracy in all cases. The adversarial models achieve $> 99\%$ adversarial training accuracy in all cases. We train the adversarial models under the ℓ_∞ attack model with PGD adversarial training and $\epsilon = 0.3$. For computing the max in each training step, we use 40 steps of PGD, with step size 0.01 (the parameters used in [6]). We use the Adam optimizer. The final robust test accuracy when training with the full training set was 91%.

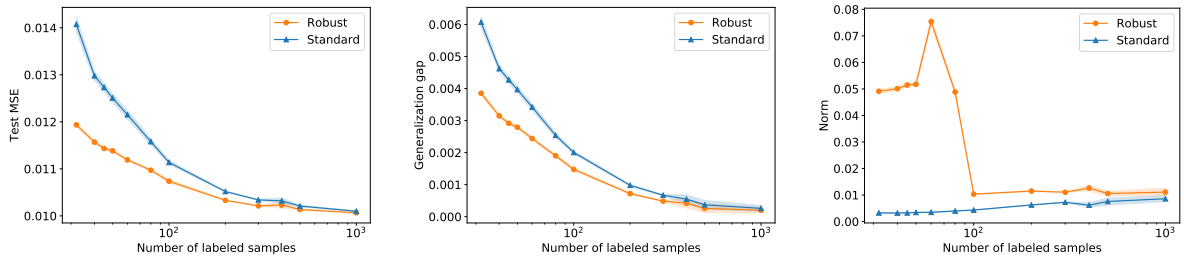
Initialization and trade-off for MNIST . We note here that the tradeoff for adversarial training reported in [14] is because the adversarially trained model hasn't converged (even after a large number of epochs). Using the Xavier initialization, we get faster convergence with adversarial training and see no drop in clean accuracy at the same level of robust accuracy. Interestingly, standard training is not affected by initialization, while adversarial training is dramatically affected.



(a) Test MSE (b) Generalization gap (c) Squared norm

(d) Robust train MSE (e) Robust test MSE

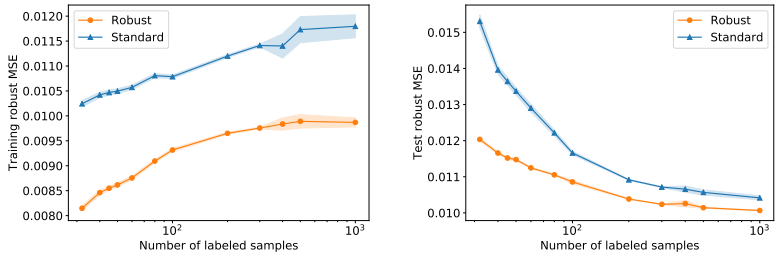
Figure 6. Plots as number of samples varies for the case where robustness hurts ($m = 1$). For each n, we pick the best regularization parameter λ with respect to standard test MSE individually for robust and standard training. **(a),(b)** The standard estimator has lower test MSE, but the gap shrinks with more samples. Note that the trend in test MSE is almost identical to generalization gap. **(c)** The robust estimator has higher norm throughout training due to learning a more complex estimator. The norm of the standard estimator increases as sample size increases as it starts to converge to the true function, which is complex. **(d, e)** The robust train and test MSE is smaller for the robust estimator throughout. With larger sample size, the standard estimator improves in robust (train and test) MSE as it converges to the true function, which is robust. Shaded regions are 1 STD.



(a) Test MSE

(b) Generalization gap (train MSE - test MSE)

(c) Squared norm



(d) Training robust MSE

(e) Test robust MSE

Figure 7. Plots as number of samples varies for the case where robustness helps ($m = 0$). For each n, we pick the best regularization parameter λ with respect to standard test MSE individually for robust and standard training. **(a),(b)** The robust estimator has lower test MSE, and the gap shrinks with more samples. Note that the trend in test MSE is almost identical to generalization gap. **(c)** The robust estimator has consistent norm throughout due to the noise-cancelling behavior of optimizing the robust objective. While the standard estimator has low norm for small samples, it has high test MSE due to finding a low norm (close to linear) solution with the wrong slope. **(d, e)** The robust train and test MSE is smaller for the robust estimator throughout. Shaded regions are 1 STD.