Regularization properties of adversarially-trained linear regression

Antônio H. Ribeiro
Uppsala University
`antonio.horta.ribeiro@it.uu.se`

Dave Zachariah
Uppsala University
`dave.zachariah@it.uu.se`

Francis Bach
École normale supérieure / INRIA
`francis.bach@inria.fr`

Thomas B. Schön
Uppsala University
`thomas.schon@it.uu.se`

Abstract

State-of-the-art machine learning models can be vulnerable to very small input perturbations that are adversarially constructed. Adversarial training is an effective approach to defend against it. Formulated as a min-max problem, it searches for the best solution when the training data were corrupted by the worst-case attacks. Linear models are among the simple models where vulnerabilities can be observed and are the focus of our study. In this case, adversarial training leads to a convex optimization problem and can be formulated as the minimization of a finite sum. We provide a comparative analysis between the solution of adversarial training in linear regression and other regularization methods. Our main findings are that: (A) adversarial training yields the minimum-norm interpolating solution in the overparameterized regime (more parameters than data), as long as the maximum disturbance radius is smaller than a threshold. And, conversely, the minimum-norm interpolator is the solution to adversarial training with a given radius. (B) Adversarial training can be equivalent to parameter shrinking methods (ridge regression and Lasso). This happens in the underparametrized region, for an appropriate choice of adversarial radius and zero-mean symmetrically distributed covariates. (C) For ℓ_∞-adversarial training—as in square-root Lasso—the choice of adversarial radius for optimal bounds does not depend on the additive noise variance. We confirm our theoretical findings with numerical examples.

1 Introduction

Adversarial attacks generated striking examples of the brittleness of modern machine learning. It considers inputs contaminated with disturbances deliberately chosen to maximize the model error and, even for small disturbances, can cause a substantial performance drop in otherwise state-of-the-art models [1]–[6]. Adversarial training [7] is one of the most effective approaches for deep learning models to defend against adversarial attacks [7]–[9]. It considers training models on samples that have been modified by an adversary, with the goal of obtaining a model that will be more robust when faced with new adversarially perturbed samples. The training procedure is formulated as a min-max problem, searching for the best solution to the worst-case attacks.

Despite its success in producing state-of-the-art results in adversarial defense benchmarks [10], there are still major challenges in using adversarial training. The min-max problem is a hard optimization problem to solve and it is still an open question how to solve it efficiently. Moreover, these methods still produce large errors in new adversarially disturbed test points, often much larger than the adversarial error obtained during training.

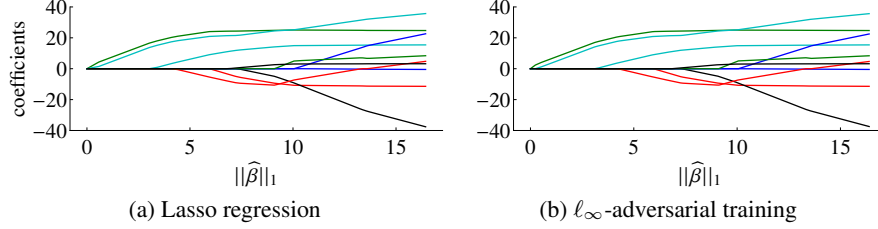|	
(a) Lasso regression	(b) ℓ_∞-adversarial training

Figure 1: **Regularization paths** estimated in the Diabetes dataset [18]. Regularization paths are plots showing the coefficient estimates for varying regularization parameter λ or, in adversarial training, perturbation radius δ. This type of plot is commonly used in analysing Lasso and variants, i.e. [18], [19]. On the horizontal axis, we give the norm of the estimated parameter. On the vertical axis, the coefficients obtained for each method. The dataset has $p = 10$ baseline variables (age, sex, body mass index, average blood pressure, and six blood serum measurements), which were obtained for $n = 442$ diabetes patients. The model output is a quantitative measure of the disease progression.

To get insight into adversarial training and try to tackle these challenges, a growing body of work [11]–[15] studies fundamental properties of adversarial attacks and adversarial training in linear models. Linear models allow for analytical analysis while still reproducing phenomena observed in state-of-the-art models. Consider a training dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ consisting of n data points of dimension $\mathbb{R}^p \times \mathbb{R}$, adversarial training in linear regression corresponds to minimizing (in $\boldsymbol{\beta} \in \mathbb{R}^p$)

$$R^{\text{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|) = \frac{1}{n}\sum_{i=1}^n \max_{\|\boldsymbol{\Delta x}_i\| \le \delta} |y_i - (\boldsymbol{x}_i + \boldsymbol{\Delta x}_i)^\top \boldsymbol{\beta}|^2. \tag{1}$$

Restricting the analysis to linear models allows for some important simplifications. In this case, the problem is convex. Moreover, the next proposition allow us to express R^{adv} in terms of the dual norm, $\|\boldsymbol{\beta}\|_* = \sup_{\|\boldsymbol{x}\| \le 1} |\boldsymbol{\beta}^\top \boldsymbol{x}|$.

> **Proposition 1** (Dual formulation). Let $\|\cdot\|_*$ be the dual norm of $\|\cdot\|$, then
>
> $$R^{\text{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|) = \frac{1}{n}\sum_{i=1}^n \left(|y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}| + \delta\|\boldsymbol{\beta}\|_*\right)^2. \tag{2}$$

This simple reformulation removes one of the major challenges: it removes the inner optimization problem and yields a model that is more amenable to analysis. Similar reformulations can be obtained for classification. This simplification made the analysis of adversarial attacks in linear models fruitful to get insights into the properties of adversarial training and examples. From giving a counterexample to the idea that deep neural networks' vulnerabilities to adversarial attacks were caused by their nonlinearity [2], [5], [16], to help explaining how implicitly regularized overparametrized models can be robust, see [17].

In this paper, we further explore this reformulation and *provide a thorough characterization of adversarial training in linear regression problems*. We do this in a comparative fashion, establishing when the solution of adversarially-trained linear models coincides with the solution of traditional regularization methods. For instance, a simple inspection of (2) hints at the similarities with other parameter shrinking methods. This similarity is confirmed in Figure 1, which illustrates how the solutions of ℓ_∞-adversarial training solutions can be almost indistinguishable from Lasso. We explain the similarities and also differences with other methods. Our contributions are:

A. In the overparametrized region, we prove that the *minimum-norm interpolator is the solution to adversarial training for δ smaller than a certain threshold* (Section 4).
B. We establish *conditions under which the solution coincides with Lasso and ridge regression* (Section 5).
C. We show that adversarial training can be framed in the *robust regression* framework, and use it to establish connections with the *square-root Lasso*. We show that adversarial training (like the *square-root Lasso*) can obtain bounds on the prediction error that do not require the noise variance to be known (Section 6).
D. We prove a more *general version of Proposition 1*, valid for general lower-semicontinuous and convex loss functions (Section 8).

2

2 Related work

Generalization of minimum-norm interpolators. The study of minimum-norm interpolators played a key role explaining why overparametrized models generalize—an important open question for which traditional theory failed to explain empirical results [20]—and for which significant progress has been made over the past few years [21]. These estimates provide a simple scenario where we can interpolate noisy data and still generalize well. Minimum-norm interpolators have indeed been quite a fruitful setting to study the phenomenon of *benign overfitting*, i.e., when the model interpolates the training data but still generalizes well to new samples. In [22] the authors use these estimates to prove consistency, a development that later had several extensions [23]. From another angle, in a series of insightful papers, Belkin *et al.* [24], [25] explore the phenomena of *double-descent*: where a double-descent curve subsumes the textbook U-shaped bias–variance trade-off curve, with a second decrease in the error occurring beyond the point where the model has reached the capacity of interpolating the training data. Interestingly, minimum-norm interpolators are also basic scenarios for observing this phenomena [24]–[26]. *Our research connects the research on robustness to adversarial attacks to the study of generalization of minimum-norm interpolators.*

Adversarial training in linear models. The generalization of adversarial attacks in linear models is well-studied. Tsipras *et al.* [16] and Ilyas *et al.* [5] use linear models to explain the conflict between robustness and high-performance models observed in neural networks; Ribeiro *et al.* [17] use these models to show how overparameterization affects robustness to perturbations; Taheri *et al.* [11] derive asymptotics for adversarial training in binary classification. Javanmard *et al.* [12] provide asymptotics for adversarial training in linear regression, Javanmard *et al.* [27] in classification settings and Hassani *et al.* [13] for random feature regressions. Min *et al.* [14] investigate how the dataset size affects adversarial performance. Yin *et al.* [15] provide an analysis of ℓ_∞-attack on linear classifiers based on Rademacher complexity. These works, however, focus on the generalization properties. *We provide a fresh perspective on the problem by clarifying the connection of adversarial training to other regularization methods.*

Robust regression and square-root Lasso. The properties of Lasso [28] for recovering sparse parameters in noisy settings have been extensively studied, and bounds on the parameter estimation error are well-known [29]. These results however rely on choices of regularization parameters that depend on the variance of additive noise. Square-root Lasso [30] was proposed with the purpose of circumventing this difficulty. We show that for ℓ_∞-adversarial, *similarly to the square-root Lasso*, bounds can be obtained with the adversarial radius set without knowledge about the variance of additive noise. We also show that both methods fit into the robust regression framework [31].

Dual formulation. Variations of Proposition 1 are presented in [12], [17], [32]. Equivalent results in the context of classification are presented in [2], [15]. We use the reformulation extensively in our developments and also present a generalized statement for it.

3 Background

Different estimators will be relevant to our developments and will be compared in this paper.

Adversarially-trained linear regression. Our main object of study is the minimization of

$$R^{\mathrm{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|) = \frac{1}{n} \sum_{i=1}^{n} \max_{\|\boldsymbol{\Delta x}_i\| \leq \delta} |y_i - (\boldsymbol{x}_i + \boldsymbol{\Delta x}_i)^\top \boldsymbol{\beta}|^2 \overset{\text{(a)}}{=} \frac{1}{n} \sum_{i=1}^{n} \left(|y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}| + \delta \|\boldsymbol{\beta}\|_* \right)^2,$$

where equality (a) follows from Proposition 1. We use $\|\boldsymbol{\beta}\|_* = \sup_{\|\boldsymbol{x}\| \leq 1} |\boldsymbol{\beta}^\top \boldsymbol{x}|$ to denote the dual norm of $\|\cdot\|$. We highlight that the ℓ_2-norm, $\|\boldsymbol{\beta}\|_2 = \sum_i |\boldsymbol{\beta}_i|^2$ is dual to itself. The ℓ_1-norm, $\|\boldsymbol{\beta}\|_1 = \sum_i |\boldsymbol{\beta}_i|$, is the dual norm of the ℓ_∞-norm, $\|\boldsymbol{\beta}\|_\infty = \max_i |\boldsymbol{\beta}_i|$. When the adversarial disturbances are constrained to the ℓ_p ball: $\{\boldsymbol{\Delta x} : \|\boldsymbol{\Delta x}\|_p \leq \delta\}$ we will call it ℓ_p-adversarial training. Our focus will be on ℓ_∞ and ℓ_2-adversarial training.

Parameter-shrinking methods. Parameter-shrinking methods explicitly penalize large values of $\boldsymbol{\beta}$. Types of regression that involve shrinkage include Lasso [28] and ridge regression, which minimize, respectively

$$R^{\mathrm{lasso}}(\boldsymbol{\beta}; \lambda) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}|^2 + \lambda \|\boldsymbol{\beta}\|_1 \quad \text{and} \quad R^{\mathrm{ridge}}(\boldsymbol{\beta}; \lambda) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \boldsymbol{x}_i^\mathsf{T} \beta|^2 + \lambda \|\boldsymbol{\beta}\|_2^2.$$

3

Square-root Lasso: Setting (with guarantees) the lasso regularization parameters requires knowledge about the variance of the noise. Square-root Lasso [30] circumvent this difficulty by minimizing:

$$R^{\sqrt{\text{lasso}}}(\boldsymbol{\beta}, \lambda) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}|^2} + \lambda\|\boldsymbol{\beta}\|_1. \tag{3}$$

Minimum-norm interpolator: Let $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ denote the matrix of stacked vectors $\boldsymbol{x}_i$ and $\boldsymbol{y} \in \mathbb{R}^n$ is the vector of stacked outputs. When matrix $\boldsymbol{X}$ has full row rank, the linear system $\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{y}$ has multiple solutions. In this case the minimum $\|\cdot\|_*$-norm interpolator is the solution of

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_* \quad \text{subject to} \quad \boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{y}. \tag{4}$$

4   Adversarial training in the overparametrized regime

Our first contribution is to give conditions for when the minimum-norm interpolator is equivalent to adversarial training in the overparametrized case (we will assume that rank$(\boldsymbol{X}) = n$ and $n < p$). On the one hand, our result gives further insight into adversarial training. On the other hand, it allows us to see minimum-norm interpolators as a solution of the adversarial training problem.

> **Theorem 1.** Assume the matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ has full row rank, let $\widehat{\boldsymbol{\alpha}}$ denote the solution of the dual problem $\max_{\|\boldsymbol{\alpha}^\top \boldsymbol{X}\| \leq 1} \boldsymbol{\alpha}^\top \boldsymbol{y}$, and $\bar{\delta}$ denote the threshold
>
> $$\bar{\delta} = (n\|\widehat{\boldsymbol{\alpha}}\|_\infty)^{-1}. \tag{5}$$
>
> The minimum $\|\cdot\|_*$-norm interpolator minimizes the adversarial risk $R^{\text{adv}}(\theta, \delta, \|\cdot\|)$ iff $\delta \in (0, \bar{\delta}]$.

Remark 1 (Bounds on $\bar{\delta}$). The theorem allows us to directly compute $\bar{\delta}$, but it does require solving the dual problem to the minimum-norm solution. In Appendix A.3, we provide bounds on $\bar{\delta}$ that avoid solving the optimization problem. For instance, for ℓ_∞-adversarial attacks, we have the following bounds depending on the singular values of $\boldsymbol{X}$: $\frac{1}{\sqrt{p}}\sigma_n(\boldsymbol{X}) \leq n\bar{\delta} \leq \sqrt{p}\sigma_1(\boldsymbol{X})$. Where σ_1 and σ_n denote the largest and smallest *positive* singular values of $\boldsymbol{X}$.

Proof of Theorem 1. Let $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\alpha}}$ be the minimum $\|\cdot\|_*$-norm solution and the solution of the associated dual problem. Along the proof we denote $R^{\text{adv}}(\boldsymbol{\beta}) = R^{\text{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|)$, dropping the last two arguments. The subgradient of $R^{\text{adv}}(\boldsymbol{\beta})$ evaluated at $\widehat{\boldsymbol{\beta}}$ is:[1]

$$\partial R^{\text{adv}}(\widehat{\boldsymbol{\beta}}) = \frac{2\delta}{n} \sum_{i=1}^{n} \|\widehat{\boldsymbol{\beta}}\|_* (\epsilon_i \boldsymbol{x}_i + \delta\partial\|\widehat{\boldsymbol{\beta}}\|_*), \tag{6}$$

for any $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n) \in \mathbb{R}^n$ that satisfies $\|\boldsymbol{\epsilon}\|_\infty \leq 1$. We have that any element of $\partial\|\widehat{\boldsymbol{\beta}}\|_*$ can be written as $\boldsymbol{X}^T\widehat{\boldsymbol{\alpha}}$ (we prove that in Lemma 1 in the Appendix). Hence, we can rewrite:

$$\partial R^{\text{adv}}(\widehat{\boldsymbol{\beta}}) = \frac{2\delta}{n} \sum_{i=1}^{n} \|\widehat{\boldsymbol{\beta}}\|_* (\epsilon_i \boldsymbol{x}_i + \delta\boldsymbol{X}^T\widehat{\boldsymbol{\alpha}}), = 2\delta\|\widehat{\boldsymbol{\beta}}\|_* \left(\frac{\boldsymbol{X}^\top \boldsymbol{\epsilon}}{n} + \delta\boldsymbol{X}^\top\widehat{\boldsymbol{\alpha}}\right).$$

If $\delta \leq \bar{\delta} = 1/(n\|\widehat{\boldsymbol{\alpha}}\|_\infty)$, we can take $\boldsymbol{\epsilon} = -n\delta\widehat{\boldsymbol{\alpha}}$ and the subderivative contains zero. If $\delta > 1/(n\|\widehat{\boldsymbol{\alpha}}\|_\infty)$ then $\left(\frac{\boldsymbol{X}^\top \boldsymbol{\epsilon}}{n} + \delta\boldsymbol{X}^\top\widehat{\boldsymbol{\alpha}}\right)$ is not zero for $\|\boldsymbol{\epsilon}\|_\infty \leq 1$. $\square$

The minimum ℓ_1-norm interpolator is well studied in the context of 'basis pursuit' and allows the recovery of low-dimensional representations of sparse signals [36]. The interest in the minimum ℓ_2-norm is more recent. It had an important role on the study of the interplay between interpolation and generalization, being used in many recent papers where the double descent [24], [26] and the benign overfitting phenomena [22] were observed.

---

[1]The subgradient of a function $\omega : \mathbb{R}^p \to \mathbb{R}$ is the set:

$$\partial\omega(\boldsymbol{\beta}_0) = \{\boldsymbol{v} \in \mathbb{R}^p : \omega(\boldsymbol{\beta}) - \omega(\boldsymbol{\beta}_0) \geq \boldsymbol{v}(\boldsymbol{\beta} - \boldsymbol{\beta}_0), \forall\boldsymbol{\beta} \in \mathbb{R}^p\}.$$

See [33]–[35] for properties. We use $\partial\|\widehat{\boldsymbol{\beta}}\|_*$ to denote the subgradient of $\omega(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_*$ evaluated at $\widehat{\boldsymbol{\beta}}$.

Theorem 1 gives a new perspective on the robustness of the minimum-norm interpolator, allowing us to see it as a solution to adversarial training[2] with $\delta_{\text{train}} = \bar{\delta}$. Figure 2 shows that the radius $\bar{\delta}$ of the adversarial training problem corresponding to the minimum-norm interpolator increases with the ratio p/n. It is natural to expect that training with a larger radius would yield more adversarially robust models on new test points. Let us denote
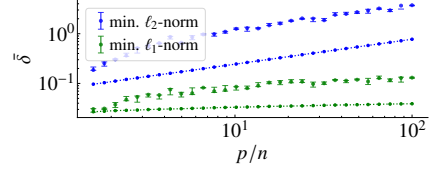

Figure 2: **Threshold $\bar{\delta}$ vs. number of features.** We fix $n = 60$ and show the value of $\bar{\delta}$ as a function of the number of features p. The dotted lines give the reference $\delta = 0.01\mathbb{E}\left[\|\boldsymbol{x}\|\right]$ for comparison.

$\mathcal{R}_*^{\text{adv}}(\boldsymbol{\beta}; \delta_{\text{test}}, \|\cdot\|) = \mathbb{E}_{y_0, \boldsymbol{x}_0}\left[\max_{\|\boldsymbol{\Delta}\boldsymbol{x}_i\| \leq \delta_{\text{test}}}(y_0 - (\boldsymbol{x}_0 + \boldsymbol{\Delta}\boldsymbol{x}_0)^\top \boldsymbol{\beta})^2\right]$ i.e., the expected adversarial squared error, and $\mathcal{R}_*(\boldsymbol{\beta}) = \mathbb{E}_{y_0, \boldsymbol{x}_0}\left[(y_0 - \boldsymbol{x}_0^\top \boldsymbol{\beta})^2\right]$ the expected squared error in the absence of an adversary on a new test point. In Figure 2, $\delta_{\text{train}} = \bar{\delta}$ increases faster than $\delta_{\text{test}} \propto \mathbb{E}\left[\|\boldsymbol{x}\|\right]$ as we add more features, hence we expect the model to become more robust. The next proposition formalizes this intuition by establishing an upper bound on the robustness gap based on the adversarial training error.

> **Proposition 2.** Assume $\boldsymbol{X}$ has full row rank and let $\widehat{\boldsymbol{\beta}}$ be the minimum $\|\cdot\|_*$-norm interpolator and $\delta_{\text{train}} = \bar{\delta}$,
>
> $$\sqrt{\mathcal{R}_*^{\text{adv}}(\widehat{\boldsymbol{\beta}}; \delta_{\text{test}}, \|\cdot\|)} - \sqrt{\mathcal{R}_*(\widehat{\boldsymbol{\beta}})} \leq \frac{\delta_{\text{test}}}{\delta_{\text{train}}}\sqrt{R^{\text{adv}}(\widehat{\boldsymbol{\beta}}; \delta_{\text{train}}, \|\cdot\|)}. \quad (7)$$

Remark 2. The example in Figure 2 is one case where adding more features makes the minimum-norm interpolator more robust. Examples showing the opposite and illustrating that adding more features to linear models can make them less adversarially robust abound in the literature [16], [17]. Indeed, examples that consider the minimum ℓ_2-interpolator subject to ℓ_∞-adversarial attacks during test-time result in this scenario, as discussed in [17]. Proposition 7 in the Appendix is the equivalent of Proposition 2 for this case (mismatched norms in train and test). There, the upper bound grows with $\sqrt{p}$, which explains the vulnerability in this scenario.

Remark 3. A pitfall of analyzing the minimum-norm solution properties in linear models is that it requires the analysis of nested problems: different choices of p require different covariates $\boldsymbol{x}$. The random projection model proposed by Bach [37] avoids this pitfall by considering the input $\boldsymbol{x}$ is fixed, but only a projected version of it $\boldsymbol{S}\boldsymbol{x}$ is observed, with the number of parameters being estimated changing with the number of projections observed. Adversarially training this model consists of finding a parameter $\widehat{\boldsymbol{\beta}}$ that minimizes: $\frac{1}{n}\sum_{i=1}^n \max_{\|\boldsymbol{\Delta}\boldsymbol{x}_i\| \leq \delta}|y_i - (\boldsymbol{x}_i + \boldsymbol{\Delta}\boldsymbol{x}_i)^\top \boldsymbol{S}^\top \boldsymbol{\beta}|^2$. We generalize our results so that we can also study this case (See Appendix A.4).

5 Adversarial training and parameter-shrinking methods

Theorem 1 stated an equivalence with the minimum-norm solution when δ is small. The next result applies to the other extreme of the spectrum, characterizing the case when δ is large.

> **Proposition 3** (Zero solution of adversarial training). The zero solution $\widehat{\boldsymbol{\beta}} = \boldsymbol{0}$ minimizes the adversarial training if and only if $\delta \geq \frac{\|\boldsymbol{X}^\top \boldsymbol{y}\|}{\|\boldsymbol{y}\|_1}$.

Hence for very small or for very large δ the solution of adversarial training is known. Our intent in this section is to characterize the solution between these two extremes. In this section, we compare adversarial training to Lasso and ridge regression.

5.1 Relation to Lasso and ridge regression

Proposition 1 makes it clear by inspection that adversarial training is similar to other well-known parameter-shrinking regularization methods. Indeed, for ℓ_∞-adversarial attacks, the cost function

---

[2]About the notation in Proposition 2, δ_{train} denote the adversarial radius used during the adversarial training, the underscript is used to make the clear contrast with δ_{test}, the test-time adversarial radius.
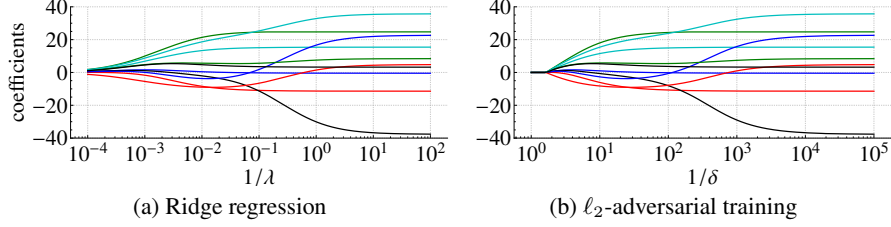
Figure 3: **Regularization paths** in the Diabetes dataset [18]. On the horizontal axis, we give the inverse of the regularization parameter (in log scale). On the vertical axis, the coefficients.

$R^{\mathrm{adv}}(\beta; \delta, \|\cdot\|_\infty)$ in its dual form is remarkably similar to Lasso [28] and $R^{\mathrm{adv}}(\beta; \delta, \|\cdot\|_2)$, to ridge regression (the cost function are presented in Section 3). In Figures 1 and 3, we show the regularization paths of these methods (the dataset is described in [18]). We observe that ℓ_∞-*adversarial training produces sparse solutions* and that Lasso and ℓ_∞-adversarial training have extremely similar regularization paths. There are also striking similarities between ridge regression and ℓ_2-adversarial training, with the notable difference that for large δ the solution of ℓ_2-adversarial training is zero (which is explained by Proposition 3). The next proposition justifies why we can observe such similarities in part of the regularization path. It applies to data that has been normalized and for positive responses y (as it is the case in our example). It is proved in Appendix B.

Proposition 4. If $y \geq 0$ and $X^\top \mathbf{1} = \mathbf{0}$, i.e., the data is normalized. Let $\widehat{\beta}$ be the solution of adversarial training then for $\|\widehat{\beta}\|_* \leq \min_i \frac{|y_i|}{\|x_i\|}$ it is also the solution of

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^n |y_i - x_i^\top \beta|^2 + \left(\delta\|\beta\|_* + \frac{1}{n}\|y\|_1\right)^2.$$

For instance, if we consider the ℓ_∞-adversarial problem. Under the necessary assumptions, $\widehat{\beta}$ is the solution of

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^n (y_i - x_i^\top \beta)^2 + \left(\delta\|\beta\|_1 + \frac{1}{n}\|y\|_1\right)^2.$$

Which in turn, is the Lagrangian formulation of the following constrained optimization problem

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^n (y_i - x_i^\top \beta)^2 \text{ subject to } \left(\delta\|\beta\|_1 + \frac{1}{n}\|y\|_1\right)^2 \leq \Delta,$$

for some $\Delta \geq \frac{1}{n}\|y\|_1$. And, we can rewrite the constraint as $\delta\|\beta\|_1 \leq \sqrt{\Delta} - \frac{1}{n}\|y\|_1$. In turn, the Lagrangian formulation of this modified problem is

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^n (y_i - x_i^\top \beta) + \lambda'\|\beta\|_1.$$

for some λ'. Hence, Proposition 4 establishes the equivalence between ℓ_∞-adversarial training and Lasso (even though there is not a closed-formula expression for the map between δ and λ'). A similar reasoning could be used to connect the result for ℓ_2-adversarial training with ridge regression.

More generally, even when the condition on $\widehat{\beta}$ is not satisfied and if y is negative, we show in Appendix B that for zero-mean and symmetrically distributed covariates, i.e., $\mathbb{E}[x] = 0$ and $(x \sim -x)$, adversarial training has approximately the same solution as

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^n |y_i - x_i^\top \beta|^2 + \left(\delta\|\beta\|_* + \frac{1}{n} s^\top y\right)^2,$$

for $s = \mathrm{sign}(y - X^\top \widehat{\beta})$. This explains the similarities in the regularization paths.

5.2 Transition into the interpolation regime

The discussion above hints at the similarities between adversarial training and parameter-shrinking methods. Interestingly, Lasso and ridge regression are also connected to minimum-norm interpolators.
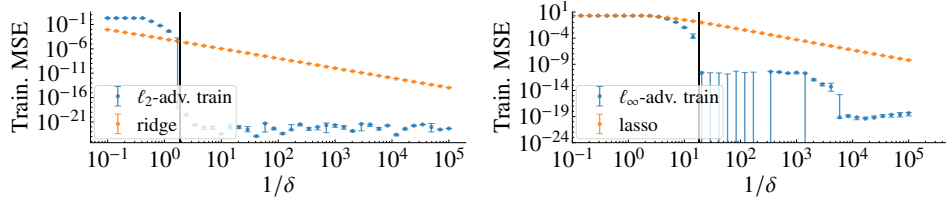
Figure 4: **Training MSE *vs* adversarial radius**. *Left:* for ridge and ℓ_2-adversarial training. *Right:* for Lasso and ℓ_∞-adversarial training. The error bars give the median and the 0.25 and 0.75 quantiles from 5 realizations. Vertical black lines give $\bar{\delta}$, see (5). Figure S.8 (appendix) shows the test MSE.

The ridge regression solution converges to the minimum-norm solution as the parameter vanishes i.e., $\widehat{\boldsymbol{\beta}}^{\text{ridge}}(\lambda) \to \widehat{\boldsymbol{\beta}}^{\min-\ell_2}$ as $\lambda \to 0^+$. Similarly, there is a relation between the minimum ℓ_1-norm solution and Lasso. The relation requires additional constraints because, for the overparameterized case, Lasso does not necessarily have a unique solution. Nonetheless, it is proved in [38, Lemma 7] that the Lasso solution by the LARS algorithm satisfies $\widehat{\boldsymbol{\beta}}^{\text{lasso}}(\lambda) \to \widehat{\boldsymbol{\beta}}^{\min-\ell_1}$ as $\lambda \to 0^+$.

For a sufficiently small δ, the solution of ℓ_2-adversarial training equals the minimum ℓ_2-norm solution; and, the solution of ℓ_∞-adversarial training equals the minimum ℓ_1-norm solution. There is a notable difference though: while this happens only in the limit for ridge regression and Lasso, for adversarial training this happens for all values δ smaller than the threshold $\bar{\delta}$. We illustrate this phenomenon next. Unlike ridge regression and Lasso which converge towards the interpolation solution, adversarial training goes through abrupt transitions and suddenly starts to interpolate the data. Figure 4 illustrates this phenomenon in synthetically generated data (with isotropic Gaussian feature, see Section 7).

Intuitive explanation for abrupt transitions. To get insight into why the abrupt transition occurs we present the case where $n = 1$, i.e., the dataset has a single data point $(\boldsymbol{x}, y)$, where $y = 1$ and $\|\boldsymbol{x}\|_2 = 1$. We can write $R^{\text{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|_2) = (\mathcal{L}(\boldsymbol{\beta}))^2$, where

$$\mathcal{L}(\boldsymbol{\beta}) = |y - \boldsymbol{x}^{\mathsf{T}}\boldsymbol{\beta}| + \delta\|\boldsymbol{\beta}\|_2.$$

Figure 5: **Function** $\mathcal{L}(\boldsymbol{\beta})$ for $\boldsymbol{\beta} = \alpha\boldsymbol{x}$.

The function $\mathcal{L}$ is necessarily minimized along the subspace spanned by the vector $\boldsymbol{x}$. Now, along this line, the function $\mathcal{L}$ is piecewise linear with three segments (See Figure 5). Thus explaining why abrupt transitions occur: one of the three segments changes the slope sign as we change δ and the minimum of the function changes abruptly.
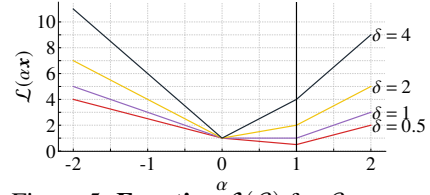
6 Relation to robust regression and square-root Lasso

In this section we establish both adversarial training and square-root Lasso can be integrated into the robust regression framework. We further look into the similarities between the methods by analyzing the prediction error upper bounds.

Robust regression framework. Robust linear regression considers the minimization of the cost function:

$$R^{\text{robust}}(\boldsymbol{\beta}; \mathcal{S}) = \max_{\boldsymbol{\Delta} \in \mathcal{S}} \|\boldsymbol{y} - (\boldsymbol{X} + \boldsymbol{\Delta})\boldsymbol{\beta}\|_2, \tag{8}$$

where the disturbance matrix $\boldsymbol{\Delta}$ is constrained to belong to the 'disturbance set' $\mathcal{S}$. We connect robust regression, square-root Lasso and adversarial training. The next proposition gives the equivalence between robust regression for a row-bounded disturbance sets $\mathcal{R}_{p,\delta}$ and ℓ_p-adversarial training, see the appendix for the proof.

Proposition 5. For a disturbance set with the rows bounded by δ:

$$\mathcal{R}_{p,\delta} = \left\{ \begin{bmatrix} \text{---} & \boldsymbol{\Delta}\boldsymbol{x}_1 & \text{---} \\ & \vdots & \\ \text{---} & \boldsymbol{\Delta}\boldsymbol{x}_n & \text{---} \end{bmatrix} : \|\boldsymbol{\Delta}\boldsymbol{x}_i\| \le \delta, \forall i \right\},$$

7

we have $\arg\min_{\boldsymbol{\beta}} R^{\text{robust}}(\boldsymbol{\beta}, \mathcal{R}_{p,\delta}) = \arg\min_{\boldsymbol{\beta}} R^{\text{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|)$.

On the other hand, there is an equivalence between square-root Lasso and robust regression for column-bounded disturbance sets $\mathcal{C}_{2,\delta}$. It was established by Xu *et al.* [31, Theorem 1] and we repeat it below.

> **Proposition 6.** [31] For a disturbance set with columns bounded by δ:
>
> $$\mathcal{C}_{2,\delta} = \left\{ \begin{bmatrix} | & & | \\ \boldsymbol{\zeta}_1 & \cdots & \boldsymbol{\zeta}_m \\ | & & | \end{bmatrix} : \|\boldsymbol{\zeta}_i\|_2 \leq \delta, \forall i \right\},$$
>
> we have $\arg\min_{\boldsymbol{\beta}} R^{\text{robust}}(\boldsymbol{\beta}; \mathcal{C}_{2,\delta}) = \arg\min_{\boldsymbol{\beta}} R^{\sqrt{\text{lasso}}}(\boldsymbol{\beta}, \delta)$.

The above discussion hints at the similarities between adversarial training and square-root Lasso, as instances of robust regression under different constraints. Square-root Lasso minimizes $R^{\sqrt{\text{lasso}}}(\boldsymbol{\beta}, \lambda) = n^{-1/2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2 + \lambda\|\boldsymbol{\beta}\|_1$ (See Section 3). The main motivation for the square-root Lasso is that it is a 'pivotal' method for sparse recovery: that is, it attains near-oracle performance without knowledge of the variance levels to set the regularization parameter [30]. As we will show in the next section, a similar property applies to ℓ_∞-adversarial training.

Fixed-design analysis and similarities with square-root Lasso. In this section, we assume that the data was generated as: $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}^* + \varepsilon_i$ where $\boldsymbol{\beta}^*$ is the parameter vector used to generate the data. Under these assumptions, we can derive the upper bound for the (in-sample) prediction error:

> **Theorem 2.** Let $\delta > \delta^* = 3\frac{\|\boldsymbol{X}^\top\boldsymbol{\varepsilon}\|_\infty}{\|\boldsymbol{\varepsilon}\|_1}$, the prediction error of ℓ_∞-adversarial training satisfies the bound:
> $$\frac{1}{n}\|\boldsymbol{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2^2 \leq 8\delta\|\boldsymbol{\beta}^*\|_1 \left(\frac{1}{n}\|\boldsymbol{\varepsilon}\|_1 + 10\delta\|\boldsymbol{\beta}^*\|_1\right). \qquad (9)$$

For comparison, we also provide the result for Lasso: (Adapted from [29, Thm. 7.13, p. 210])

> **Theorem 3.** [29] Let $\lambda > \lambda^* = 3\|\frac{\boldsymbol{X}^\top\boldsymbol{\varepsilon}}{n}\|_\infty$, the prediction error of Lasso satisfies the bound:
> $$\frac{1}{n}\|\boldsymbol{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2^2 \leq 8\lambda\|\boldsymbol{\beta}^*\|_1. \qquad (10)$$

Without additional constraints (see Remark 5), one can show that the above bound for Lasso is unimprovable. To satisfy this bound, however, Lasso requires knowledge of the magnitude of the noise $\boldsymbol{\varepsilon}$. In Theorem 3, $\lambda^* = \frac{3}{n}\|\boldsymbol{X}^\top\boldsymbol{\varepsilon}\|$. Hence, if we rescale $\boldsymbol{\varepsilon}$ (i.e., $\boldsymbol{\varepsilon} \to \eta\boldsymbol{\varepsilon}$) then a correspondent change in magnitude follows in λ^* (i.e., $\lambda^* \to \eta\lambda^*$). Square-root Lasso [30] avoids this problem and achieves a similar rate even without knowing the variance: i.e., it is a 'pivotal' method. Our method has similar properties and allows us to set δ without estimating the variance. This can be seen in Theorem 2, where re-scaling $\boldsymbol{\varepsilon}$ does not alter the value of $\delta^* = 3\|\boldsymbol{X}^\top\boldsymbol{\varepsilon}\|/\|\boldsymbol{\varepsilon}\|_1$ because it affects the numerator and denominator simultaneously.

For instance, if we assume $\boldsymbol{\varepsilon}$ has i.i.d. $\mathcal{N}(0, \sigma^2)$ entries and that the matrix $\boldsymbol{X}$ is fixed with $\max_{j=1,\ldots,m} \|\boldsymbol{x}_j\|_\infty \leq R$. For $\lambda \propto R\sigma\sqrt{(\log p)/n}$, we (with high-probability) satisfy the condition in Theorem 3, obtaining: $\frac{1}{n}\|\boldsymbol{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2^2 \lesssim R\sigma\sqrt{(\log p)/n}$. For adversarial training, we can set: $\delta \propto R\sqrt{(\log p)/n}$, and (with high-probability) satisfy the theorem condition, obtaining the same bound. Notice that the choice of δ is not dependent on σ. We provide a full analysis in Appendix C.3. We conclude this section with two remarks and by mentioning that a potential advantage of ℓ_∞-adversarial training over square-root Lasso is its cost function, which has a finite sum structure.

Remark 4 (On the relation between $\bar{\delta}$ and δ^*). For sufficiently large n, we have $\delta^* > \bar{\delta}$ in the scenario above—the noise $\boldsymbol{\varepsilon}$ has i.i.d. normal entries $\mathcal{N}(0, \sigma^2)$ and the matrix $\boldsymbol{X}$ is fixed. We can prove it by contradiction: if $\delta^* \leq \bar{\delta}$ then we could apply the bound from Theorem 2 to the minimum ℓ_1-norm interpolator (due to Theorem 1). Hence, $\sigma^2 \approx \frac{1}{n}\|\boldsymbol{\varepsilon}\|_2^2 = \frac{1}{n}\|\boldsymbol{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2^2 \lesssim R\sigma\sqrt{(\log p)/n}$ and we can always choose a sufficiently large value of n for which the inequality is false.

Remark 5. In the original square-root Lasso paper [30], a bound is obtained for the ℓ_2 parameter distance: $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2^2$ for the case $\boldsymbol{X}$ satisfy the restricted eigenvalue condition and $\boldsymbol{\beta}^*$ is sparse. Under these more strict assumptions (restricted eigenvalue condition) one can also obtain a faster

convergence rate for the prediction error. Here we focus on the fixed-design prediction error without additional assumptions on $\boldsymbol{X}$. For other predictors, these other proofs follow similar steps and yield similar requirements on λ^*, see [29, Chapter 7]. But we leave these and other analyses (such as variable selection consistency) in the case of adversarial training for future work.

7 Numerical Experiments

We study five different examples. Our main goal is to experimentally confirm our theoretical findings. For each scenario, we compute and plot: train and test MSE for different choices of p, n and δ. For comparison purposes, we also compute and plot train and test MSE for Lasso and ridge regression and minimum-norm interpolators. Finally, in line with our discussion in Section 4, we compute and plot $\bar{\delta}$ as a function of p/n. The scenarios under consideration are: (Appendix D contain more details)

1. **Isotropic Gaussian feature model**. The output is a linear combination of the features plus additive noise: $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \epsilon_i$, for Gaussian noise and covariates: $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_p)$.
2. **Latent-space feature model** [26, Section 5.4]. The features $\boldsymbol{x}$ are noisy observations of a lower-dimensional subspace of dimension d. A vector in this *latent space* is represented by $\boldsymbol{z} \in \mathbb{R}^d$. $\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{u}$. The output is a linear combination of the latent space plus noise.
3. **Random Fourier features model** [39] We apply a random Fourier feature map to inputs of the Diabetes dataset [18]. Random fourier features are obtained by the transformation $\boldsymbol{x}_i = \sqrt{2/m}\cos(\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{b})$ where the entries of $\boldsymbol{W}$ and $\boldsymbol{b}$ are independently sampled. It can be seen as a one-layer untrained neural network and approximates the Gaussian kernel feature map [39].
4. **Random projection model.** [37] The data is generated as in the isotropic Gaussian feature model. But we only observe $\boldsymbol{S}\boldsymbol{x}_i$, i.e., a projection of the inputs.
5. **Phenotype prediction from genotype.** We illustrate our method on the Diverse MAGIC wheat dataset [40] from the National Institute for Applied Botany. We use a subset of the genotype to predict one of the continuous phenotypes.

Results. We experimentally confirm Theorem 1 in all scenarios: In Figures 4 and S.7 (Appendix) we show the train MSE as we change the adversarial radius δ, confirming the abrupt transitions into interpolation regime and also that $\bar{\delta}$ is the transition point. In Figures S.8 and S.9 we show the corresponding test MSE without and in the presence of an adversary. Interestingly, the optimal parameter δ_{train} for the adversarial training is not always equal to the strength of the attack the model will be evaluated on δ_{test}. Figures 2 and S.4 (Appendix) display $\bar{\delta}$ as a function of the ratio p/n. We observe $\bar{\delta}/\mathbb{E}\left[\|\boldsymbol{x}\|\right]$ is growing in all examples and we would still expect improved robustness in light of Proposition 2. Random Fourier features model is the only case where $\bar{\delta}$ seems to decrease (in absolute value). However, $\mathbb{E}\left[\|\boldsymbol{x}\|_1\right]$ is also decreasing (and at a faster rate). Figures Figures S.5 and S.6 give test MSE without and in the presence of an adversary for the minimum-norm interpolator as a function of the ratio p/n, confirming improved adversarial robustness as p grows.

Figure 6 provides a comparison of the test error of different methods under study. For Lasso, ridge and adversarial training, we use the best δ or λ available for each method (obtained via grid search). We note that while for $p = 1000$ optimally tuned Lasso and ℓ_∞-adversarial training significantly outperform the corresponding minimum ℓ_1-norm interpolator. As p increases, the performance of the three different methods becomes quite similar. It is also an example where the minimum ℓ_1-norm outperforms the minimum ℓ_2-norm interpolator (for large p).

In the same setup, Figure S.11 study a choice of adversarial radius inspired by Theorem 2. We use $\delta \propto \|\boldsymbol{X}\boldsymbol{\xi}\|_\infty/\|\boldsymbol{\xi}\|_1$ for $\boldsymbol{\xi}$ a vector with zero-mean normal entries. We use a random $\boldsymbol{\xi}$, since we do not know the true additive noise. Even with this approximation, ℓ_∞-adversarial training performs comparably with Lasso with the regularization parameter set using 5-fold cross-validation doing a full search among the hyperparameter space. The figure also provides a comparison with square-root Lasso under a similar setting.
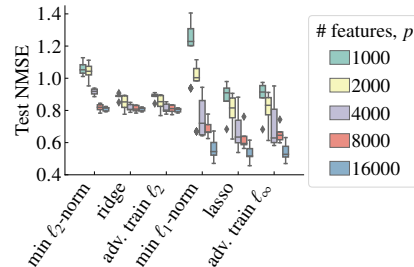

Figure 6: **Test normalized MSE (NMSE) in MAGIC dataset.**

9

8 Results for general loss functions

Let $\mathcal{X}$ be a Banach space, with norm $\|\cdot\|$. And let $\boldsymbol{\beta} \in \mathcal{X}^*$ be a continuous linear map $\beta : \mathcal{X} \to \mathbb{R}$. We will use the notation $\boldsymbol{x}^\top \boldsymbol{\beta} := \boldsymbol{\beta}(\boldsymbol{x})$, to keep the analogy with the previous section more clear. Let $\|\cdot\|_*$ denote the dual norm. The following theorem can be used to generalize Proposition 1.

Theorem 4. Let $\mathcal{L} : \mathbb{R} \to \mathbb{R}$ be a convex and lower-semicontinuous function, for every $\delta \geq 0$,

$$\max_{\|\boldsymbol{\Delta x}\| \leq \delta} \mathcal{L}\left((\boldsymbol{x} + \boldsymbol{\Delta x})^\top \boldsymbol{\beta}\right) = \max_{s \in \{-1,1\}} \mathcal{L}\left(\boldsymbol{x}^\top \boldsymbol{\beta} + \delta s \|\boldsymbol{\beta}\|_*\right). \tag{11}$$

We can find a closed-formula expression for $s_* = \arg\max_{s \in \{-1,1\}} \mathcal{L}(\boldsymbol{\beta}^\top x + \delta s \|\boldsymbol{\beta}\|_*)$ in many cases of interest. For **regression problems**, given any non-decreasing function $\ell : \mathbb{R}^+ \to \mathbb{R}^+$, and $\mathcal{L}(\boldsymbol{x}^\top \boldsymbol{\beta}) = \ell(|\boldsymbol{x}^\top \boldsymbol{\beta} - y|)$. If ℓ is lower semicontinuous and convex so will be $\mathcal{L}$ and the result of the theorem holds (the squared loss $\ell(z) = z^2$ is a special case). Then $s_* = -\text{sign}(y - \boldsymbol{x}^\top \boldsymbol{\beta})$ and

$$\max_{\|\boldsymbol{\Delta x}\| \leq \delta} \ell(|(\boldsymbol{x} + \boldsymbol{\Delta x})^\top \boldsymbol{\beta} - y|) = \ell(|y - x^\top \boldsymbol{\beta}| + \delta \|\boldsymbol{\beta}\|_*).$$

For **classification**, let $y \in \{-1,1\}$ and $\mathcal{L}(\boldsymbol{x}^\top \boldsymbol{\beta}) = \ell(y(\boldsymbol{x}^\top \boldsymbol{\beta}))$ where ℓ is a non-increasing function. If ℓ is lower semicontinuous and convex so will be $\mathcal{L}$ and the result of the theorem holds. Then $s_* = -y$ and we obtain:

$$\max_{\|\boldsymbol{\Delta x}\| \leq \delta} \ell(y((\boldsymbol{x} + \boldsymbol{\Delta x})^\top \boldsymbol{\beta})) = \ell(y(\boldsymbol{x}^\top \boldsymbol{\beta}) - \delta \|\boldsymbol{\beta}\|_*).$$

The above result also can be applied to unsupervised learning. In the Appendix, we illustrate how Theorem 4 can be used for **dimensionality reduction**. We consider the problem of finding $\boldsymbol{P}$ that minimizes the reconstruction error $\|\boldsymbol{x} + \boldsymbol{P}\boldsymbol{P}^\top \boldsymbol{x}\|_2^2$ (that yields PCA algorithm) and derive an adversarial version of it.

9 Conclusion

We study adversarial training in linear regression. Adversarial training allows us to depart from the traditional regularization setting where we can decompose the loss and the regularization terms, minimizing $\frac{1}{n} \sum_{i=0}^n \ell(\boldsymbol{x}_i^\top \boldsymbol{\beta}, y_i) + \Omega(\boldsymbol{\beta})$ for some penalization function Ω. We showcase how it provides a new angle into the minimum-norm interpolator, by proving a new equivalence between the two methods (Section 4). While adversarially-trained linear regression arrives in similar solutions to parameter shrinking methods in some scenarios (Section 5), it also has some advantages, for instance, the adversarial radius might be set without the knowledge of the noise variance (Section 6). Unlike, squared-root Lasso it achieves this while still solving the minimization of a sum of squared terms. We believe a natural next step is to provide a tailored solver that can allow for efficient solution for models with many features, rendering adversarially-trained linear regression useful, for instance, for modeling genetics data (see one minimal example of phenotype prediction from genotype in Section 7). This is a natural direction we are interested in further developing. Another interesting direction is generalizing our results to classification and nonlinear models. We believe the results in Section 8 might be especially useful to this second development.

Acknowledgment

References

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

[2]   I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *International Conference on Learning Representations (ICLR)*, 2015.

[3]   A. Kurakin, I. Goodfellow, S. Bengio, *et al.*, "Adversarial attacks and defences competition," in *The NIPS '17 competition: Building Intelligent Systems*, 2018, pp. 195–231.

[4]   A. Fawzi, O. Fawzi, and P. Frossard, "Analysis of classifiers' robustness to adversarial perturbations," *Machine Learning*, vol. 107, no. 3, pp. 481–508, 2018.

[5]   A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial Examples Are Not Bugs, They Are Features," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[6]   X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[7]   A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," *International Conference for Learning Representations (ICLR)*, 2018.

[8]   R. Huang, B. Xu, D. Schuurmans, and C. Szepesvari, "Learning with a Strong Adversary," *arXiv:1511.03034*, 2016.

[9]   T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent Advances in Adversarial Training for Adversarial Robustness," *arXiv:2102.01356*, 2021.

[10] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein, "RobustBench: A standardized adversarial robustness benchmark," in *NeurIPS Datasets and Benchmarks track*, 2021.

[11] H. Taheri, R. Pedarsani, and C. Thrampoulidis, "Asymptotic Behavior of Adversarial Training in Binary Classification," *IEEE International Symposium on Information Theory (ISIT)*, vol. 127-132, 2022.

[12] A. Javanmard, M. Soltanolkotabi, and H. Hassani, "Precise tradeoffs in adversarial training for linear regression," in *Proceedings of the Conference on Learning Theory*, vol. 125, 2020, pp. 2034–2078.

[13] H. Hassani and A. Javanmard, "The curse of overparametrization in adversarial training: Precise analysis of robust generalization for random features regression," *arXiv:2201.05149*, 2022.

[14] Y. Min, L. Chen, and A. Karbasi, "The Curious Case of Adversarially Robust Models: More Data Can Help, Double Descend, or Hurt Generalization," *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, vol. 161, pp. 129–139, 2021.

[15] D. Yin, R. Kannan, and P. Bartlett, "Rademacher Complexity for Adversarially Robust Generalization," in *Proceeding of the International Conference on Machine Learning*, 2019, pp. 7085–7094.

[16] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Ma, "Robustness May Be At Odds with Accuracy," *International Conference for Learning Representations*, 2019.

[17] A. H. Ribeiro and T. B. Schön, "Overparameterized Linear Regression under Adversarial Attacks," *IEEE Transactions on Signal Processing*, 2023.

[18] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[19] T. Hastie, R. Tibshirani, and J. Friedman, *Elements of Statistical Learning*, Second. Springer Science & Business Media, 2009.

[20] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *International Conference on Learning Representations*, 2017.

[21] P. L. Bartlett, A. Montanari, and A. Rakhlin, "Deep learning: A statistical viewpoint," *arXiv:2103.09177*, 2021.

[22] P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler, "Benign overfitting in linear regression," *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30 063–30 070, 2020.

[23] F. Koehler, L. Zhou, D. J. Sutherland, and N. Srebro, "Uniform Convergence of Interpolators: Gaussian Width, Norm Bounds and Benign Overfitting," in *NeurIPS*, 2021.

[24] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proceedings of the National Academy of Sciences*, vol. 116, no. 32, pp. 15 849–15 854, 2019.

[25] M. Belkin, D. Hsu, and J. Xu, "Two Models of Double Descent for Weak Features," *SIAM Journal on Mathematics of Data Science*, vol. 2, no. 4, pp. 1167–1180, 2020.

[26] T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani, "Surprises in high-dimensional ridgeless least squares interpolation," *The Annals of Statistics*, vol. 50, no. 2, pp. 949–986, 2022.

[27] A. Javanmard and M. Soltanolkotabi, "Precise statistical analysis of classification accuracies for adversarial training," *The Annals of Statistics*, vol. 50, no. 4, pp. 2127–2156, 2022.

[28] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[29] M. J. Wainwright, *High-Simensional Statistics: a non-Asymptotic Viewpoint*, ser. Cambridge series on statistical and probabilistic mathematics 48. Cambridge University Press, 2019.

[30] A. Belloni, V. Chernozhukov, and L. Wang, "Square-Root Lasso: Pivotal Recovery of Sparse Signals via Conic Programming," *Biometrika*, vol. 98, no. 4, pp. 791–806, 2011.

[31] H. Xu, C. Caramanis, and S. Mannor, "Robust regression and lasso," *Advances in Neural Information Processing Systems*, vol. 21, 2008.

[32] Y. Xing, Q. Song, and G. Cheng, "On the Generalization Properties of Adversarial Training," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2021, pp. 505–513.

[33] D. P. Bertsekas, A. Nedi, and A. E. Ozdaglar, *Convex Analysis and Optimization*. 2003.

[34] F. H. Clarke, *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1990.

[35] S. Boyd, J. Duchi, M. Pilanci, and L. Vandenberghe, "Subgradients (Lecture Notes)," Tech. Rep. Notes for EE364b, Stanford University, Spring 2021-22, 2022.

[36] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[37] F. Bach, "High-dimensional analysis of double descent for linear regression with random projections," *arXiv:2303.01372*, 2023.

[38] R. J. Tibshirani, "The Lasso Problem and Uniqueness," *Electronic Journal of Statistics*, vol. 7, pp. 1456–1490, 2013.

[39] A. Rahimi and B. Recht, "Random Features for Large-Scale Kernel Machines," in *Advances in Neural Information Processing Systems 20*, 2008, pp. 1177–1184.

[40] M. F. Scott, N. Fradgley, A. R. Bentley, *et al.*, "Limited haplotype diversity underlies polygenic trait architecture across 70 years of wheat breeding," *Genome Biology*, vol. 22, no. 1, p. 137, 2021.

Appendix

Table of Contents

A Adversarial training in the overparametrized regime

A.1 Additional proofs

The next lemma was used in the proof of Theorem 1. It formalizes the equivalence between the minimum-norm solution of $y = X\beta$ and the dual problem we need to solve to obtain $\widehat{\alpha}$.

> **Lemma 1.** The following equivalence hold
>
> $$\min_{y=X\beta} \|\boldsymbol{\beta}\|_* = \max_{\|\boldsymbol{\alpha}^\top X\| \leq 1} \boldsymbol{\alpha}^\top \boldsymbol{y}. \qquad (\text{S.1})$$
>
> Furthermore, $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\alpha}}$ be the arguments at optimality if and only if $\widehat{\boldsymbol{\alpha}}^\top X \in \partial \|\widehat{\boldsymbol{\beta}}\|_*$.

Proof of Lemma 1. By strong duality:

$$\min_{\boldsymbol{y}=X\theta} \|\theta\|_* = \max_\alpha \min_\theta (\|\theta\|_* + \alpha^T(y - X\theta)) = \max_\alpha \left(\alpha^T \boldsymbol{y} + \min_\theta (\|\theta\|_* - \alpha^T X\theta) \right)$$

Now, the Fenchel conjugate of $\|\cdot\|_*$ is the indicator function on the ball of the norm $\|\cdot\|$. Hence, we obtain the result. $\qquad\square$

A.2 Results on the robustness gap and adversarial training

Next we provide a proof of Proposition 2.

Proof of Proposition 2. Using Theorem 4, for any distribution on $(\boldsymbol{x}, y)$

$$\mathbb{E}_{\boldsymbol{x},y}\left[\max_{\|\boldsymbol{\Delta x}\|\leq\delta}(y - (\boldsymbol{x} + \boldsymbol{\Delta x})^{\top}\boldsymbol{\beta}|^2)\right] = \mathbb{E}_{\boldsymbol{x},y}\left[(|y - \boldsymbol{x}^{\top}\boldsymbol{\beta}| + \delta\|\boldsymbol{\beta}\|_*)^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x},y}\left[(y - \boldsymbol{x}^{\top}\boldsymbol{\beta})^2\right] + 2\delta\|\boldsymbol{\beta}\|_*\mathbb{E}_{\boldsymbol{x},y}\left[|y - \boldsymbol{x}^{\top}\boldsymbol{\beta}|\right] + \delta^2\|\boldsymbol{\beta}\|_*^2,$$

Now, since $0 \leq \mathbb{E}_{\boldsymbol{x},y}\left[|y_i - \boldsymbol{x}_i^{\top}\boldsymbol{\beta}|\right] \leq \sqrt{\mathbb{E}_{\boldsymbol{x},y}\left[|y_i - \boldsymbol{x}_i^{\top}\boldsymbol{\beta}|^2\right]}$ (by Jensen inequality), we have that

$$\mathbb{E}_{\boldsymbol{x},y}\left[(y - \boldsymbol{x}^{\top}\boldsymbol{\beta})^2\right] + \delta^2\|\boldsymbol{\beta}\|_*^2 \leq \mathbb{E}_{\boldsymbol{x},y}\left[(|y - \boldsymbol{x}^{\top}\boldsymbol{\beta}| + \delta\|\boldsymbol{\beta}\|_*)^2\right] \leq \left(\sqrt{\mathbb{E}_{\boldsymbol{x},y}\left[(y_i - \boldsymbol{x}_i^{\top}\boldsymbol{\beta})^2\right]} + \delta\|\boldsymbol{\beta}\|_*\right)^2.$$

If we denote, $R_*^{\mathrm{adv}}(\boldsymbol{\beta}; \delta_{\mathrm{test}}, \|\cdot\|) = \mathbb{E}_{y,\boldsymbol{x}}\left[\max_{\|\boldsymbol{\Delta x}_i\|\leq\delta_{\mathrm{test}}}(y - (\boldsymbol{x} + \boldsymbol{\Delta x})^{\top}\boldsymbol{\beta})^2\right]$ i.e., the expected adversarial squared error and $R_*(\boldsymbol{\beta}) = (y_0 - (\boldsymbol{x}_0 + \boldsymbol{\Delta x}_0)^{\top}\boldsymbol{\beta})^2$ the expected squared error in the absence of an adversary on a new test point. We can rewrite it as:

$$R_*(\boldsymbol{\beta}) + \delta^2\|\boldsymbol{\beta}\|_*^2 \leq R_*^{\mathrm{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|) \leq \left(\sqrt{R_*(\boldsymbol{\beta})} + \delta\|\boldsymbol{\beta}\|_*\right)^2.$$

Rearranging:

$$\sqrt{R_*^{\mathrm{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|)} - \sqrt{R_*(\boldsymbol{\beta})} \leq \delta\|\boldsymbol{\beta}\|_* \leq \sqrt{R_*^{\mathrm{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|) - R_*(\boldsymbol{\beta})}. \qquad (\text{S.2})$$

Now for the empirical adversarial distribution:

$$R^{\mathrm{adv}}(\widehat{\boldsymbol{\beta}}; \bar{\delta}, \|\cdot\|) = \frac{1}{n}\sum_{i=1}^{n}(|y_i - \boldsymbol{x}_i^{\top}\widehat{\boldsymbol{\beta}}| + \bar{\delta}\|\boldsymbol{\beta}\|_*)^2$$

$$= \bar{\delta}^2\|\boldsymbol{\beta}\|_*^2$$

where the last step follows from considering that $\widehat{\boldsymbol{\beta}}$ is an interpolator (hence $y_i - \boldsymbol{x}_i^{\top}\widehat{\boldsymbol{\beta}} = 0, \forall i$). Plugging this result back into (S.2). Let $\delta = \delta_{\mathrm{test}}$ and $\delta' = \delta_{\mathrm{train}}$ $\qquad\square$

The next result holds for mismatched ℓ_{∞} and ℓ_2-adversarial attacks

> **Proposition 7.** Let $\widehat{\boldsymbol{\beta}}$ be the minimum $\|\cdot\|_*$-norm interpolator and $\delta_{\mathrm{train}} = \bar{\delta}$,
>
> $$\sqrt{R_*^{\mathrm{adv}}(\boldsymbol{\beta}; \delta_{\mathrm{test}}, \|\cdot\|_{\infty})} - \sqrt{R_*(\boldsymbol{\beta})} \leq \sqrt{p}\frac{\delta_{\mathrm{test}}}{\delta_{\mathrm{train}}}\sqrt{R^{\mathrm{adv}}(\boldsymbol{\beta}; \delta_{\mathrm{train}}, \|\cdot\|_2)} \qquad (\text{S.3})$$

Proof. A similar derivation as in the proof of Proposition 2 yields:

$$\sqrt{R_*^{\mathrm{adv}}(\widehat{\boldsymbol{\beta}}; \delta, \|\cdot\|_{\infty})} - \sqrt{R_*(\boldsymbol{\beta})} \leq \delta\|\widehat{\boldsymbol{\beta}}\|_1 \leq \sqrt{R_*^{\mathrm{adv}}(\widehat{\boldsymbol{\beta}}; \delta, \|\cdot\|_{\infty}) - R_*(\widehat{\boldsymbol{\beta}})}$$

$$R^{\mathrm{adv}}(\widehat{\boldsymbol{\beta}}; \bar{\delta}, \|\cdot\|_2) = \bar{\delta}^2\|\widehat{\boldsymbol{\beta}}\|_2^2.$$

Now since $\|\widehat{\boldsymbol{\beta}}\|_2 \leq \|\widehat{\boldsymbol{\beta}}\|_1 \leq \sqrt{p}\|\widehat{\boldsymbol{\beta}}\|_2$ the result follows for $\delta = \delta_{\mathrm{test}}$ and $\bar{\delta} = \delta_{\mathrm{train}}$. $\qquad\square$

A.3 Bounds on $\bar{\delta}$

We point that although the exact value of $\bar{\delta}$ requires the solution of the dual problem. It is easy to come up with upper and lower bounds that depend only on $\boldsymbol{X}$. On the one hand, by construction $\|\boldsymbol{X}^{\top}\widehat{\boldsymbol{\alpha}}\| \leq 1$, hence,

$$\frac{1}{\|\widehat{\boldsymbol{\alpha}}\|_{\infty}} \geq \frac{\|\boldsymbol{X}^{\top}\widehat{\boldsymbol{\alpha}}\|}{\|\widehat{\boldsymbol{\alpha}}\|_{\infty}}.$$

On the other hand, $\boldsymbol{X}^\top\widehat{\boldsymbol{\alpha}} \in \partial\|\widehat{\boldsymbol{\beta}}\|_*$ such that $\widehat{\boldsymbol{\alpha}}^\top\boldsymbol{X}\widehat{\boldsymbol{\beta}} = \|\widehat{\boldsymbol{\beta}}\|_*$. Hence a simple application of Hölder inequality yields:

$$\frac{\|\boldsymbol{X}\widehat{\boldsymbol{\beta}}\|_1}{\|\widehat{\boldsymbol{\beta}}\|_*} \geq \frac{1}{\|\widehat{\boldsymbol{\alpha}}\|_\infty}$$

Now since, $n\bar{\delta} = \frac{1}{\|\widehat{\boldsymbol{\alpha}}\|_\infty}$ we have that:

$$\inf_{\boldsymbol{u}\in\mathbb{R}^n} \frac{\|\boldsymbol{X}^\top\boldsymbol{u}\|}{\|\boldsymbol{u}\|_\infty} \leq n\bar{\delta} \leq \sup_{\boldsymbol{v}\in\mathbb{R}^m} \frac{\|\boldsymbol{X}\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_*}$$

For instance, for ℓ_∞-adversarial attacks, this speciallize to

$$\inf_{\boldsymbol{u}\in\mathbb{R}^n} \frac{\|\boldsymbol{X}^\top\boldsymbol{u}\|_\infty}{\|\boldsymbol{u}\|_\infty} \leq n\bar{\delta} \leq \sup_{\boldsymbol{v}\in\mathbb{R}^m} \frac{\|\boldsymbol{X}\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_1}$$

Let, $\sigma_1(\boldsymbol{X}) \geq \cdots \geq \sigma_n(\boldsymbol{X})$ singular values of the matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$. We have that $\sigma_1(\boldsymbol{X}) = \sup_{\boldsymbol{v}\in\mathbb{R}^m} \frac{\|\boldsymbol{X}\boldsymbol{v}\|_2}{\|\boldsymbol{v}\|_2}$ and $\sigma_n(\boldsymbol{X}) = \inf_{\boldsymbol{u}\in\mathbb{R}^n} \frac{\|\boldsymbol{X}^\top\boldsymbol{u}\|_2}{\|\boldsymbol{u}\|_2}$ Now we can use standard norm inequalities, let $\boldsymbol{w} \in \mathbb{R}^m$: $\|\boldsymbol{w}\|_2 \leq \|\boldsymbol{w}\|_1 \leq \sqrt{m}\|\boldsymbol{w}\|_2$ and $\|\boldsymbol{w}\|_\infty \leq \|\boldsymbol{w}\|_2 \leq \sqrt{m}\|\boldsymbol{w}\|_\infty$ to obtain, that for for ℓ_∞-adversarial attacks:

$$\frac{1}{\sqrt{m}}\sigma_n(\boldsymbol{X}) \leq n\bar{\delta} \leq \sqrt{m}\sigma_1(\boldsymbol{X})$$

Similarly ℓ_2-adversarial attacks one can show that $\sigma_n \leq n\bar{\delta} \leq \sqrt{m}\sigma_1$.

It is also possible to establish a relationship with the minimum-norm interpolator. Let $\widehat{\boldsymbol{\beta}}$ be the minimum norm interpolators, using Hölder inequality

$$\|\widehat{\boldsymbol{\beta}}\|_* = \widehat{\boldsymbol{\alpha}}^\top\boldsymbol{y} \leq \|\widehat{\boldsymbol{\alpha}}\|_\infty\|\boldsymbol{y}\|_1$$

And it follows that :

$$\bar{\delta}\|\widehat{\boldsymbol{\beta}}\|_* \leq \frac{1}{n}\|\boldsymbol{y}\|_1$$

Now, if we assume that the data was generated as $y_i = \boldsymbol{x}_i^\top\boldsymbol{\beta}^* + \varepsilon_i$ for $i = 1,\cdots,n$. From Hölder inequality and the definition of the dual problem

$$\widehat{\boldsymbol{\alpha}}^\top\boldsymbol{X}\boldsymbol{\beta}^* \leq \|\widehat{\boldsymbol{\alpha}}^\top\boldsymbol{X}\|\|\boldsymbol{\beta}^*\|_* \leq \|\boldsymbol{\beta}^*\|_* \tag{S.4}$$

Now:

$$\begin{aligned}
\|\widehat{\boldsymbol{\beta}}\|_* &\overset{(a)}{=} \widehat{\boldsymbol{\alpha}}^\top\boldsymbol{y} \overset{(b)}{=} \widehat{\boldsymbol{\alpha}}^\top\boldsymbol{X}\boldsymbol{\beta}^* + \widehat{\boldsymbol{\alpha}}^\top\boldsymbol{\varepsilon} \\
&\overset{(c)}{\leq} \|\boldsymbol{\beta}^*\|_* + \widehat{\boldsymbol{\alpha}}^\top\boldsymbol{\varepsilon} \\
&\leq \|\boldsymbol{\beta}^*\|_* + \|\widehat{\boldsymbol{\alpha}}\|_\infty\|\boldsymbol{\varepsilon}\|_1 \\
&= \|\boldsymbol{\beta}^*\|_* + \frac{1}{n}\frac{\|\boldsymbol{\varepsilon}\|_1}{\bar{\delta}}
\end{aligned}$$

where (a) follows Lemma 1, (b) follows from the data model definition and finaly (c) follows from Eq. (S.4).

A.4 Linear maps and random projections

As an extension, we consider the following framework: We consider a matrix $\boldsymbol{S} \in \mathbb{R}^{p\times d}$ that maps from the input space $\mathbb{R}^d$ to a feature space $\mathbb{R}^p$. For instance, in Bach [37] this scenario—with the entries of $\boldsymbol{S}$ sampled from a Rademacher distribution—is used to study the phenomena of double-descent. For this case, the adversarial problem consists of finding a parameter $\widehat{\boldsymbol{\beta}}$ that minimizes:

$$R_{\boldsymbol{S}}^{\text{adv}}(\boldsymbol{\beta};\delta,\|\cdot\|) = \frac{1}{n}\sum_{i=1}^n \max_{\|\boldsymbol{\Delta}\boldsymbol{x}_i\|\leq\delta} |y_i - (\boldsymbol{x}_i + \boldsymbol{\Delta}\boldsymbol{x}_i)^\top\boldsymbol{S}^\top\boldsymbol{\beta}|^2. \tag{S.5}$$

Equation (S.5) also allow for reformulation:

Proposition 8 (Dual formulation: linear maps). Let $\|\cdot\|_*$ be the dual norm of $\|\cdot\|$, then

$$R_{\boldsymbol{S}}^{\mathrm{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|) = \frac{1}{n} \sum_{i=1}^{n} \left(|y_i - \boldsymbol{x}_i^\top \boldsymbol{S}^\top \boldsymbol{\beta}| + \delta \|\boldsymbol{S}^\top \boldsymbol{\beta}\|_* \right)^2. \tag{S.6}$$

this is a direct consequence of the more general result Theorem 7. We also have the following theorem relating the adversarial training solution to the minimum $\|\cdot\|_*$-norm interpolator.

Theorem 5. Assume the matrix $\boldsymbol{X}\boldsymbol{S}^\top \in \mathbb{R}^{n \times p}$ has full row rank (i.e., $\mathrm{rank}(\boldsymbol{X}\boldsymbol{S}^\top) = n$). The minimum-norm solution

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{S}^\top \boldsymbol{\beta}\|_* \quad \text{subject to} \quad \boldsymbol{X}\boldsymbol{S}^\top \boldsymbol{\beta} = \boldsymbol{y}, \tag{S.7}$$

minimizes the adversarial risk $R_{\boldsymbol{S}}^{\mathrm{adv}}(\theta, \delta, \|\cdot\|)$ if and only if $\delta \in (0, \bar{\delta}]$. For $\bar{\delta} = \frac{1}{n\|\widehat{\boldsymbol{\alpha}}\|_\infty}$ where $\widehat{\boldsymbol{\alpha}}$ denote the solution of the dual problem $\max_{\|\boldsymbol{\alpha}^\top \boldsymbol{X}\boldsymbol{P}\| \le 1} \boldsymbol{\alpha}^\top y$, where $\boldsymbol{P} = \boldsymbol{S}^\top (\boldsymbol{S}\boldsymbol{S}^\top)^{-1} \boldsymbol{S}$.

B Adversarial training and parameter shrinking methods

B.1 Background

The subderivative of a function $\omega : \mathbb{R}^p \to \mathbb{R}$ evaluated at a point $\boldsymbol{\beta}_0$ is the set

$$\partial \omega(\boldsymbol{\beta}_0) = \{\boldsymbol{v} \in \mathbb{R}^p : \omega(\boldsymbol{\beta}) - \omega(\boldsymbol{\beta}_0) \ge \boldsymbol{v}(\boldsymbol{\beta} - \boldsymbol{\beta}_0) \, \forall \boldsymbol{\beta} \in \mathbb{R}^p\}.$$

In this section, for convenience, we will drop the two last arguments of $R^{\mathrm{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|)$ and denote it only by $R^{\mathrm{adv}}(\boldsymbol{\beta})$. Let $\mathcal{L}_i(\boldsymbol{\beta}) = |y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}| + \delta\|\boldsymbol{\beta}\|_*$, then the partial derivative of R^{adv} with respect to $\boldsymbol{\beta}$ is

$$\partial R^{\mathrm{adv}}(\boldsymbol{\beta}) = \frac{2}{n} \sum_{i=1}^{n} \mathcal{L}_i(\boldsymbol{\beta}) \partial \mathcal{L}_i(\boldsymbol{\beta}), \text{ where } \partial L_i(\boldsymbol{\beta}) = \boldsymbol{x}_i \partial |\boldsymbol{x}_i^\top \boldsymbol{\beta} - y_i| + \delta \partial \|\boldsymbol{\beta}\|_*, \tag{S.8}$$

where

$$\partial |a| = \begin{cases} \{1\} \text{ if } a > 0 \\ \{-1\} \text{ if } a < 0 \\ \{\gamma : \gamma \in [-1, 1]\} \text{ if } a = 0, \end{cases}$$

and:

$$\partial \|\boldsymbol{\beta}\|_* = \left\{ \boldsymbol{\alpha} : \|\boldsymbol{\alpha}\| \le 1, \boldsymbol{\alpha}^\top \boldsymbol{\beta} = \|\boldsymbol{\beta}\|_* \right\}. \tag{S.9}$$

Since, $R^{\mathrm{adv}}(\boldsymbol{\beta})$ is a convex function of $\boldsymbol{\beta}$, we have that $\widehat{\boldsymbol{\beta}}$ is a solution of $\min R^{\mathrm{adv}}(\boldsymbol{\beta})$ iff $\boldsymbol{0} \in \partial R^{\mathrm{adv}}(\widehat{\boldsymbol{\beta}})$.

B.2 Zero solution to adversarial training: Proposition 3

Proposition 3 stated that *The zero solution $\widehat{\boldsymbol{\beta}} = \boldsymbol{0}$ minimizes the adversarial training iff $\delta \ge \frac{\|\boldsymbol{X}^\top \boldsymbol{y}\|}{\|\boldsymbol{y}\|_1}$.* Indeed, one can notice from Figure 3 and Figures S.2 and S.3 there is a threshold of δ, such that for all δ above such threshold the solution is identical to zero. We provide proof for this theorem next.

Proof of Proposition 3. On the one hand,

$$\partial R^{\mathrm{adv}}(\boldsymbol{0}) = \frac{2}{n} \sum_{i=1}^{n} |y_i| (\boldsymbol{x}_i \partial |y_i| + \delta \partial \|\boldsymbol{0}\|_*).$$

In matrix form:

$$\partial R^{\mathrm{adv}}(\boldsymbol{0}) = \frac{2}{n} \left(\boldsymbol{X}^\top \boldsymbol{y} + \delta \|\boldsymbol{y}\|_1 \partial \|\boldsymbol{0}\|_* \right)$$

where we used that $|y_i| \partial |y_i| = y_i$. Now, $\partial \|\boldsymbol{0}\|_* = \{\boldsymbol{\alpha} : \|\boldsymbol{\alpha}\| \le 1\}$, hence for $\boldsymbol{z} \in \partial \|\boldsymbol{0}\|_*$, we obtain from triangular inequality that:

$$\|\boldsymbol{X}^\top \boldsymbol{y}\| - \delta \|\boldsymbol{y}\|_1 \le \left\| \boldsymbol{X}^\top \boldsymbol{y} + \delta \|\boldsymbol{y}\|_1 \boldsymbol{z} \right\|.$$

On the one hand, if $\frac{\|\boldsymbol{X}^\top \boldsymbol{y}\|}{\|\boldsymbol{y}\|_1} > \delta$, the left-hand-side of the above inequality is larger than zero, and $\boldsymbol{0} \notin \partial R^{\text{adv}}(\boldsymbol{0})$. On the other hand, if $\frac{\|\boldsymbol{X}^\top \boldsymbol{y}\|}{\|\boldsymbol{y}\|_1} \leq \delta$, we have $\boldsymbol{z} = \boldsymbol{X}^T \boldsymbol{y}/(\delta\|\boldsymbol{y}\|_1) \in \partial\|\boldsymbol{0}\|_*$ (since $\|\boldsymbol{z}\| \leq 1$), and:

$$\boldsymbol{X}^\top \boldsymbol{y} + \delta\|\boldsymbol{y}\|_1 \boldsymbol{z} = \boldsymbol{0}.$$

Therefore $\boldsymbol{0} \in \partial R^{\text{adv}}(\boldsymbol{0})$ and the proof is complete. $\qquad\square$

B.3 On the similarities of regularization paths: Proposition 4 and extensions

The next phenomenon we want to explain is the similarity between Lasso and ℓ_∞-adversarial attacks regularization paths. As well as the similarities between ridge and ℓ_2-adversarial attacks regularization paths.

> **Theorem 6.** Let $\widehat{\boldsymbol{\beta}}(\delta)$ be the minimizer of $R^{\text{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|)$, define the vector $\boldsymbol{s}(\delta) = \text{sign}(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}(\delta))$. Assume that $y_i \neq \boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}(\delta)$ for every i. If $\boldsymbol{X}^\top \boldsymbol{s}(\delta) = \boldsymbol{0}$ than $\widehat{\boldsymbol{\beta}}(\delta)$ is a minimizer of
>
> $$\frac{1}{n}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \left(\delta\|\boldsymbol{\beta}\|_* + \frac{1}{n}\boldsymbol{s}(\delta)^\top \boldsymbol{y}\right)^2, \qquad (S.10)$$

Proof. We will prove first that if $\boldsymbol{X}^\top \boldsymbol{s}(\delta) = 0$, than a minimizer of $R^{\text{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|)$ is also a minimizer of:

$$\frac{1}{n}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \left(\delta\|\boldsymbol{\beta}\|_* + \frac{1}{n}\boldsymbol{s}(\delta)^\top \boldsymbol{y}\right)^2 \qquad (S.11)$$

is also a minimizer of $R^{\text{adv}}(\boldsymbol{\beta}; \delta, \|\cdot\|)$. Multiplying the terms in (S.8) and putting into matrix form:

$$\frac{1}{2}\partial R^{\text{adv}}(\boldsymbol{\beta}) = \frac{1}{n}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}) + \frac{\delta\|\boldsymbol{\beta}\|_*}{n}\boldsymbol{X}^\top\partial\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_1 + \left(\frac{1}{n}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_1 + \delta\|\boldsymbol{\beta}\|_*\right)\delta\partial\|\boldsymbol{\beta}\|_*$$

Now, we have that $\partial\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{y}\|_1 = \{-\boldsymbol{s}(\delta)\}$ and that $\|\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{y}\|_1 = \boldsymbol{s}(\delta)^\top(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}) = \boldsymbol{s}(\delta)^\top \boldsymbol{y}$, hence:

$$\frac{1}{2}\partial R^{\text{adv}}(\widehat{\boldsymbol{\beta}}) = \frac{1}{n}\boldsymbol{X}^\top(\boldsymbol{X}\widehat{\boldsymbol{\beta}} - \boldsymbol{y}) + \left(\delta\|\widehat{\boldsymbol{\beta}}\|_* + \frac{1}{n}\boldsymbol{s}(\delta)^\top \boldsymbol{y}\right)\delta\partial\|\widehat{\boldsymbol{\beta}}\|_*$$

The left-hand side is the subderivative of (S.11) evaluated at $\widehat{\boldsymbol{\beta}}$. Since $\widehat{\boldsymbol{\beta}}$ is the minimizer of $R^{\text{adv}}(\widehat{\boldsymbol{\beta}})$, than $\boldsymbol{0} \in \partial R^{\text{adv}}(\widehat{\boldsymbol{\beta}})$ and it follows that $\widehat{\boldsymbol{\beta}}$ is also a minimizer of (S.11). $\qquad\square$

Proposition 4 is weaker than necessary. But we choose it to be part of the main text instead instead of Theorem 6 because it has a cleaner interpretation: it only depends on the norm of $\widehat{\boldsymbol{\beta}}$ and not on its direction. And $\boldsymbol{X}^\top \boldsymbol{1} = 0$ has the interpretation that the data has been normalized. The proof follows directly from the Theorem.

Proof of Proposition 4. We can use the Hölder inequality $|\boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}| \leq \|\widehat{\boldsymbol{\beta}}\|_*\|\boldsymbol{x}_i\|$, to show that if $\|\widehat{\boldsymbol{\beta}}\|_*\|\boldsymbol{x}_i\| \leq |y_i|$ than $|\boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}(\delta)| \leq |y_i|$ for all i. In this case, $\boldsymbol{s}(\delta) = \text{sign}(\boldsymbol{y})$. If additionally if $\boldsymbol{y} > \boldsymbol{0}$, the Theorem implies that as long as $\boldsymbol{X}^\top \boldsymbol{1} = 0$ then $\widehat{\boldsymbol{\beta}}(\delta)$ is the minimizer of:

$$\frac{1}{n}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \left(\delta\|\boldsymbol{\beta}\|_* + \frac{1}{n}\|\boldsymbol{y}\|_1\right)^2.$$

$\qquad\square$

We notice that the theorem conclusion holds even under the (less strict) condition, $|\boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}(\delta)| \leq |y_i|$ for every i. Figure 1 highlights the part of the regularization path for which this condition holds. Showing that even for values of δ for which this condition does not hold, the regularization paths are still extremely similar.
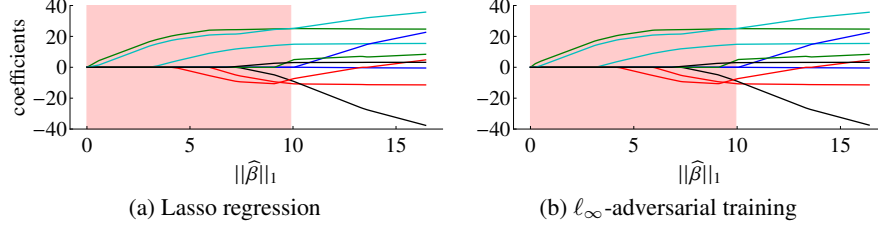
(a) Lasso regression　　　　(b) ℓ_∞-adversarial training

Figure S.1: **Regularization paths: diabetes dataset**. On the horizontal axis, we give the $\|\widehat{\boldsymbol{\beta}}\|_1$. On the vertical axis, we show the coefficients of the learned linear model. The hashed area gives the values of $\widehat{\boldsymbol{\beta}}$ for which $|\boldsymbol{x}_i^\top \widehat{\boldsymbol{\beta}}(\delta)| \leq |y_i|$ for every i.

This can be explained from the same argument use in Theorem 6. Assume the hypothesis that the covariates $\{\boldsymbol{x}_i\}_{i=1}^n$ are i.i.d. and are sampled from a symmetric and zero-mean distribution, i.e. $\boldsymbol{x} \sim -\boldsymbol{x}$ and $\mathbb{E}\,[\boldsymbol{x}] = 0$. In this case, we notice that if $s \in \{-1, 1\}$ then by symmetry of the distribution $s\boldsymbol{x} \sim \boldsymbol{x}$, and we have that $\mathbb{E}\,[s\boldsymbol{x}] = \boldsymbol{0}$. Now, from the law of large numbers, $\frac{1}{n}\sum s_i \boldsymbol{x}_i \to 0$ as $n \to 0$. Let $\widehat{\boldsymbol{\beta}}$ be the solution of

$$\frac{1}{n}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \left(\delta\|\boldsymbol{\beta}\|_* + \frac{1}{n}\boldsymbol{s}^\top \boldsymbol{y}\right)^2.$$

The same argument used in the proof of Theorem can than be used to show that in this case $\partial R^{\mathrm{adv}}(\widehat{\boldsymbol{\beta}}) \to 0$ as $n \to \infty$. Hence, for large enought n both problems have approximately the same solution.
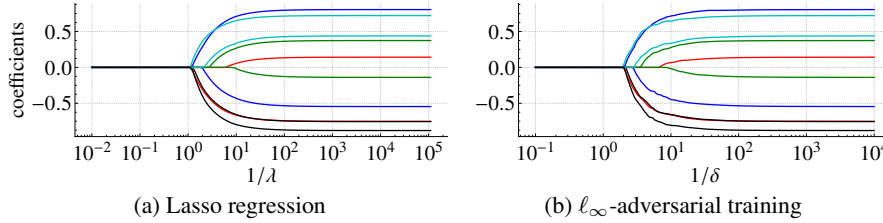
B.4　Regularization path for Gaussian covariates

(a) Lasso regression　　　　(b) ℓ_∞-adversarial training

Figure S.2: **Regularization paths: Gaussian covariates.** Lasso and ℓ_∞-adversarial training. On the horizontal axis, we give the inverse of the regularization parameter (in log scale). On the vertical axis we show the coefficients of the learned linear model.
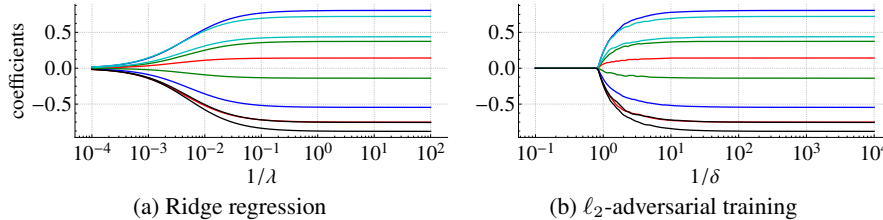

(a) Ridge regression　　　　(b) ℓ_2-adversarial training

Figure S.3: **Regularization paths: Gaussian covariates**. Ridge regression and ℓ_2-adversarial training. On the horizontal axis, we give the inverse of the regularization parameter (in log scale). On the vertical axis we show the coefficients of the learned linear model.

C   Relation to robust regression and square-root Lasso

C.1   Proof of Proposition 5

Proof. It follows from noting that

$$R_p^{\mathrm{adv}}(\boldsymbol{\beta};\delta) = \frac{1}{n}\max_{\boldsymbol{\Delta}\in\mathcal{R}_p(\delta)}\|\boldsymbol{y}-(\boldsymbol{X}+\boldsymbol{\Delta})\boldsymbol{\beta}\|_2^2 = \frac{1}{n}\left(\max_{\boldsymbol{\Delta}\in\mathcal{R}_p(\delta)}\|\boldsymbol{y}-(\boldsymbol{X}+\boldsymbol{\Delta})\boldsymbol{\beta}\|_2\right)^2.$$

Where the first equality follows from the definition of adversarial training and the last equality from the fact that the function $h(z) = \frac{1}{n}z^2$ is monotonically increasing for $z \geq 0$. Repeating the same argument, but now for the minimization, implies that $R_p^{\mathrm{adv}}(\boldsymbol{\beta};\delta)$ has the same minimizer as $\max_{\boldsymbol{\Delta}\in\mathcal{R}_p}\|y-(\boldsymbol{X}+\boldsymbol{\Delta})\boldsymbol{\beta}\|_2$. $\qquad\square$

C.2   Error bounds for ℓ_∞-adversarial training for sparse recovery

In this section, we will consider exclusively the empirical ℓ_∞-adversarial risk:

$$R^{\mathrm{adv}}(\boldsymbol{\beta};\delta,\|\cdot\|_\infty) = \frac{1}{n}\sum_{i=1}^n\left(|y_i - \boldsymbol{x}_i^\top\boldsymbol{\beta}| + \delta\|\boldsymbol{\beta}\|_1\right)^2.$$

For convenience, we will denote it only $R^{\mathrm{adv}}(\boldsymbol{\beta})$ in this section and we denote $\widehat{\boldsymbol{\beta}}$ the minimizer of this optimization problem. Moreover, we assume that the data was generated as:

$$y_i = \boldsymbol{x}_i^\top\boldsymbol{\beta}^* + \varepsilon_i, \tag{S.12}$$

where $\boldsymbol{\beta}^*$ is the parameter vector used to generate the data. Again we denote $\boldsymbol{X}$ the matrix of vectors $\boldsymbol{x}_i$ stacked and $\boldsymbol{y}$ and $\boldsymbol{\varepsilon}$ the vectors of stacked outputs and noise components respectively. The follow result holds for the (in-sample) prediction error of adversarial training.

Theorem 1. *Let* $\delta > \delta^* = 3\frac{\|\boldsymbol{X}^\top\boldsymbol{\varepsilon}\|_\infty}{\|\boldsymbol{\varepsilon}\|_1}$, *then the prediction error of* ℓ_∞-*adversarial training satisfies the bound:*

$$\frac{1}{n}\|\boldsymbol{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2^2 \leq 8\delta\|\boldsymbol{\beta}^*\|_1\left(\frac{1}{n}\|\boldsymbol{\varepsilon}\|_1 + 10\delta\|\boldsymbol{\beta}^*\|_1\right). \tag{S.13}$$

Proof. Let $\mathcal{L}_i(\boldsymbol{\beta}) = |y_i - \boldsymbol{x}_i^\top\boldsymbol{\beta}| + \delta\|\boldsymbol{\beta}\|_1$, then

$$\partial R^{\mathrm{adv}}(\boldsymbol{\beta}) = \frac{2}{n}\sum_{i=1}^n\mathcal{L}_i(\boldsymbol{\beta})\partial\mathcal{L}_i(\boldsymbol{\beta}), \text{ where } \partial L_i(\boldsymbol{\beta}) = \boldsymbol{x}_i\partial|y_i - \boldsymbol{x}_i^\top\boldsymbol{\beta}| + \delta\partial\|\boldsymbol{\beta}\|_1.$$

Multiplying the terms and expanding

$$\frac{1}{2}\partial R^{\mathrm{adv}}(\boldsymbol{\beta}) = \frac{1}{n}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}) + \frac{\delta\|\boldsymbol{\beta}\|_1}{n}\boldsymbol{X}^\top\partial\|\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\|_1 + \left(\frac{1}{n}\|\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\|_1 + \delta\|\boldsymbol{\beta}\|_1\right)\delta\partial\|\boldsymbol{\beta}\|_1.$$

Now, since $\boldsymbol{0} \in \partial R^{\mathrm{adv}}(\widehat{\boldsymbol{\beta}})$ we must have $\widehat{\boldsymbol{z}} \in \partial\|\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y}\|_1$ and $\widehat{\boldsymbol{w}} \in \partial\|\boldsymbol{\beta}\|_1$ such that:

$$\boldsymbol{0} = \frac{1}{n}\boldsymbol{X}^\top(\boldsymbol{X}\widehat{\boldsymbol{\beta}}-\boldsymbol{y}) + \frac{\delta\|\widehat{\boldsymbol{\beta}}\|_1}{n}\boldsymbol{X}^\top\widehat{\boldsymbol{z}} + \delta\left(\frac{1}{n}\|\boldsymbol{X}\widehat{\boldsymbol{\beta}}-\boldsymbol{y}\|_1 + \delta\|\widehat{\boldsymbol{\beta}}\|_1\right)\widehat{\boldsymbol{w}}. \tag{S.14}$$

Let us denote,

$$\widehat{\boldsymbol{\Delta}} = \widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*.$$

Taking the dot product of both sides of Eq (S.14) with $\widehat{\boldsymbol{\Delta}}$, making the substitution $\boldsymbol{X}\widehat{\boldsymbol{\beta}}-\boldsymbol{y} = \boldsymbol{X}\widehat{\boldsymbol{\Delta}}-\boldsymbol{\varepsilon}$ and rearranging we find that

$$\frac{1}{n}\|\boldsymbol{X}\widehat{\boldsymbol{\Delta}}\|_2^2 = \frac{1}{n}(\boldsymbol{X}\widehat{\boldsymbol{\Delta}})^\top\boldsymbol{\varepsilon} - \frac{\delta\|\widehat{\boldsymbol{\beta}}\|_1}{n}(\boldsymbol{X}\widehat{\boldsymbol{\Delta}})^\top\widehat{\boldsymbol{z}} - \delta\left(\frac{1}{n}\|\boldsymbol{X}\widehat{\boldsymbol{\Delta}}-\boldsymbol{\varepsilon}\|_1 + \delta\|\widehat{\boldsymbol{\beta}}\|_1\right)\widehat{\boldsymbol{\Delta}}^\top\widehat{\boldsymbol{w}}. \tag{S.15}$$

Next, we bound each of the terms highlighted above. From (S.9), since $\widehat{\boldsymbol{w}} \in \partial\|\widehat{\boldsymbol{\beta}}\|_1$, we have that $\widehat{\boldsymbol{w}}^\top\widehat{\boldsymbol{\beta}} = \|\widehat{\boldsymbol{\beta}}\|_1$ and $\widehat{\boldsymbol{w}}^\top\boldsymbol{\beta}^* \leq \|\widehat{\boldsymbol{w}}\|_\infty\|\boldsymbol{\beta}^*\|_1 \leq \|\boldsymbol{\beta}^*\|_1$. Therefore:

$$-\widehat{\boldsymbol{w}}^\top\widehat{\boldsymbol{\Delta}} \leq \|\boldsymbol{\beta}^*\|_1 - \|\widehat{\boldsymbol{\beta}}\|_1.$$

Similarly, since $\widehat{z} \in \partial \|X^\top \beta - y\|_1$, we have that

$$-\widehat{z}^\top X \widehat{\Delta} = -\widehat{z}^\top (X \widehat{\Delta} - \varepsilon) - \widehat{z}^\top \varepsilon \le -\|X \widehat{\Delta} - \varepsilon\|_1 + \|\varepsilon\|_1.$$

Moreover, we have that:

$$\frac{1}{n}(X \widehat{\Delta})^\top \varepsilon \overset{(a)}{\le} \frac{1}{n}\|X^\top \varepsilon\|_\infty \|\widehat{\Delta}\|_1 \overset{(b)}{\le} \frac{\delta}{3n}\|\varepsilon\|_1 \|\widehat{\Delta}\|_1,$$

where step (a) follows from Hölder inequality and step (b) from the condition imposed in the theorem that $\|\varepsilon\|_1 \delta > 3\|X \varepsilon\|_\infty$. Plugging these results back into (S.15) and rearranging

$$\frac{1}{n}\|X \widehat{\Delta}\|_2^2 \le \frac{\delta}{3n}\|\varepsilon\|_1 \|\widehat{\Delta}\|_1 + \frac{\delta}{n}(\|\varepsilon\|_1 - 2\|X \widehat{\Delta} - \varepsilon\|_1)\|\widehat{\beta}\|_1 +$$

$$\frac{\delta}{n}\|X \widehat{\Delta} - \varepsilon\|_1 \|\beta^*\|_1 + \delta^2 \|\widehat{\beta}\|_1 (\|\beta^*\|_1 - \|\widehat{\beta}\|_1).$$

On the one hand:

$$\|\varepsilon\|_1 - 2\|X \widehat{\Delta} - \varepsilon\|_1 \overset{(a)}{\le} \|\varepsilon\|_1 - \tfrac{3}{2}\|X \widehat{\Delta} - \varepsilon\|_1 \overset{(b)}{\le} \tfrac{3}{2}\|X \widehat{\Delta}\|_1 - \tfrac{1}{2}\|\varepsilon\|_1,$$

where, in step (a) we use the trivial fact that $-\frac{3}{2} > -2$. The number $\frac{3}{2}$ is arbitrary and other values $-2 < \alpha < -1$ should also work. In step (b), we use the triangular inequality: $\|X \widehat{\Delta} - \varepsilon\|_1 \ge \|\varepsilon\|_1 - \|X \widehat{\Delta}\|_1$. On the other hand, also using the triangular inequality, we have that: $\|X \widehat{\Delta} - \varepsilon\|_1 \le \|X \widehat{\Delta}\|_1 + \|\varepsilon\|_1$. Hence:

$$\frac{1}{n}\|X \widehat{\Delta}\|_2^2 \le \frac{\delta}{3n}\|\varepsilon\|_1 \|\widehat{\Delta}\|_1 + \frac{\delta}{n}\left(\tfrac{3}{2}\|X \widehat{\Delta}\|_1 - \tfrac{1}{2}\|\varepsilon\|_1\right)\|\widehat{\beta}\|_1 +$$

$$\frac{\delta}{n}(\|X \widehat{\Delta}\|_1 + \|\varepsilon\|_1)\|\beta^*\|_1 + \delta^2 \|\widehat{\beta}\|_1 (\|\beta^*\|_1 - \|\widehat{\beta}\|_1).$$

Rearranging the right-hand-side of the above inequality:

$$\frac{1}{n}\|X \widehat{\Delta}\|_2^2 \le \frac{\delta}{3n}\|\varepsilon\|_1 \left(\|\widehat{\Delta}\|_1 + 3\|\beta^*\| - \frac{3}{2}\|\widehat{\beta}\|_1\right) +$$

$$\frac{\delta}{n}\|X \widehat{\Delta}\|_1 \left(\|\beta^*\|_1 + \frac{3}{2}\|\widehat{\beta}\|_1\right) + \delta^2 \|\widehat{\beta}\|_1 (\|\beta^*\|_1 - \|\widehat{\beta}\|_1).$$

Finally, using that the norm inequality : $\|X \widehat{\Delta}\|_1 \le \sqrt{n}\|X \widehat{\Delta}\|_2$, we obtain

$$\frac{1}{n}\|X \widehat{\Delta}\|_2^2 \le \frac{\delta}{3n}\|\varepsilon\|_1 \left(\|\widehat{\Delta}\|_1 + 3\|\beta^*\| - \frac{3}{2}\|\widehat{\beta}\|_1\right) +$$

$$\frac{\delta}{\sqrt{n}}\|X \widehat{\Delta}\|_2 \left(\|\beta^*\|_1 + \frac{3}{2}\|\widehat{\beta}\|_1\right) + \delta^2 \|\widehat{\beta}\|_1 (\|\beta^*\|_1 - \|\widehat{\beta}\|_1).$$

Notice that the above inequality is a second-order inequality of the type $y^2 \le by + c$, where $y = \frac{1}{\sqrt{n}}\|X \widehat{\Delta}\|_2$ and b and c can be read by inspection. Since $y \ge 0$ we must have $b^2 + 4c \ge 0$, hence:

$$0 \le \frac{4}{3}\frac{\delta}{n}\|\varepsilon\|_1 \left(\|\widehat{\Delta}\|_1 + 3\|\beta^*\| - \frac{3}{2}\|\widehat{\beta}\|_1\right) + 4\delta^2 \|\widehat{\beta}\|_1 \left(\|\beta^*\|_1 - \|\widehat{\beta}\|_1\right) + \delta^2 \left(\|\beta^*\|_1 + \frac{3}{2}\|\widehat{\beta}\|_1\right)^2$$

$$\le \frac{4}{3}\frac{\delta}{n}\|\varepsilon\|_1 \left(\|\widehat{\Delta}\|_1 + 3\|\beta^*\| - \frac{3}{2}\|\widehat{\beta}\|_1\right) + \delta^2 \left(\|\beta^*\|_1^2 + 7\|\widehat{\beta}\|_1 \|\beta^*\|_1 - \frac{7}{4}\|\widehat{\beta}\|_1^2\right).$$

Factoring the leftmost term we obtain:

$$0 \le \frac{4}{3}\frac{\delta}{n}\|\varepsilon\|_1 \left(\|\widehat{\Delta}\|_1 + 3\|\beta^*\| - \frac{3}{2}\|\widehat{\beta}\|_1\right) + \delta^2 \left(\|\beta^*\|_1 + c_1\|\widehat{\beta}\|_1\right)\left(\|\beta^*\|_1 - c_2\|\widehat{\beta}\|_1\right), \quad \text{(S.16)}$$

where $c_1 = \sqrt{14} + \frac{7}{2} \leq 7.5$ and $c_2 = \sqrt{14} - \frac{7}{2} \geq 0.2$. Using the triangular inequality $\|\widehat{\boldsymbol{\Delta}}\|_1 \leq \|\boldsymbol{\beta}^*\|_1 + \|\widehat{\boldsymbol{\beta}}\|_1$, and the above inequality can be written as

$$0 \leq \frac{4}{3}\frac{\delta}{n}\|\boldsymbol{\varepsilon}\|_1 \left(4\|\boldsymbol{\beta}^*\| - \frac{1}{2}\|\widehat{\boldsymbol{\beta}}\|_1\right) + \delta^2 \left(\|\boldsymbol{\beta}^*\|_1 + 7.5\|\widehat{\boldsymbol{\beta}}\|_1\right)\left(\|\boldsymbol{\beta}^*\|_1 - 0.2\|\widehat{\boldsymbol{\beta}}\|_1\right).$$

Hence:

$$\|\widehat{\boldsymbol{\beta}}\|_1 \leq 8\|\boldsymbol{\beta}^*\|_1, \tag{S.17}$$

otherwise, the right-hand side of the inequality would be negative. Now we use the following proposition to obtain a bound on $y^2 = \frac{1}{n}\|\boldsymbol{X}\widehat{\boldsymbol{\Delta}}\|_2^2$.

Proposition 9. Let $y, b, c \in \mathbb{R}$ and $b^2 + 4c \geq 0$, if $y^2 \leq by + c$ then $y^2 \leq b^2 + 2c$.

Which yields:

$$\frac{1}{n}\|\boldsymbol{X}\widehat{\boldsymbol{\Delta}}\|_2^2 \leq \frac{2}{3}\frac{\delta}{n}\|\boldsymbol{\varepsilon}\|_1 \left(\|\widehat{\boldsymbol{\Delta}}\|_1 + 3\|\boldsymbol{\beta}^*\| - \frac{3}{2}\|\widehat{\boldsymbol{\beta}}\|_1\right) + $$

$$2\delta^2\|\widehat{\boldsymbol{\beta}}\|_1 \left(\|\boldsymbol{\beta}^*\|_1 - \|\widehat{\boldsymbol{\beta}}\|_1\right) + \delta^2 \left(\|\boldsymbol{\beta}^*\|_1 + \frac{3}{2}\|\widehat{\boldsymbol{\beta}}\|_1\right)^2.$$

Rearranging it we obtain:

$$\frac{1}{n}\|\boldsymbol{X}\widehat{\boldsymbol{\Delta}}\|_2^2 \leq \frac{2}{3}\frac{\delta}{n}\|\boldsymbol{\varepsilon}\|_1 \left(\|\widehat{\boldsymbol{\Delta}}\|_1 + 3\|\boldsymbol{\beta}^*\| - \frac{3}{2}\|\widehat{\boldsymbol{\beta}}\|_1\right) + \delta^2 \left(\|\boldsymbol{\beta}^*\|_1^2 + 7\|\widehat{\boldsymbol{\beta}}\|_1\|\boldsymbol{\beta}^*\|_1 + \frac{1}{4}\|\widehat{\boldsymbol{\beta}}\|_1^2\right)$$

$$\leq \frac{2}{3}\frac{\delta}{n}\|\boldsymbol{\varepsilon}\|_1 \left(\|\widehat{\boldsymbol{\Delta}}\|_1 + 3\|\boldsymbol{\beta}^*\| - \frac{3}{2}\|\widehat{\boldsymbol{\beta}}\|_1\right) + \delta^2 \left(\|\boldsymbol{\beta}^*\|_1 + c_1\|\widehat{\boldsymbol{\beta}}\|_1\right)\left(\|\boldsymbol{\beta}^*\|_1 + c_2\|\widehat{\boldsymbol{\beta}}\|_1\right),$$

where $c_1 = \frac{7}{2} + \sqrt{12} \leq 7$ and $c_2 = \frac{7}{2} - \sqrt{12} \geq 0.04$. Now since $\|\widehat{\boldsymbol{\beta}}\|_1 \leq 8\|\boldsymbol{\beta}^*\|_1$ and $\|\widehat{\boldsymbol{\Delta}}\|_1 \leq \|\boldsymbol{\beta}^*\|_1 + \|\widehat{\boldsymbol{\beta}}\|_1 \leq 9\|\boldsymbol{\beta}^*\|_1$,

$$\frac{1}{n}\|\boldsymbol{X}\widehat{\boldsymbol{\Delta}}\|_2^2 \leq \frac{2}{3}\frac{\delta}{n}\|\boldsymbol{\varepsilon}\|_1 \left(9\|\boldsymbol{\beta}^*\|_1 + 3\|\boldsymbol{\beta}^*\|\right) + \delta^2 \left(\|\boldsymbol{\beta}^*\|_1 + (7 \cdot 8)\|\boldsymbol{\beta}^*\|_1\right)\left(\|\boldsymbol{\beta}^*\|_1 + (0.04 \cdot 8)\|\boldsymbol{\beta}^*\|_1\right)$$

$$\leq \frac{8}{n}\delta\|\boldsymbol{\beta}^*\|_1\|\boldsymbol{\varepsilon}\|_1 + 76\delta^2\|\boldsymbol{\beta}^*\|_1^2$$

$$\leq 8\delta\|\boldsymbol{\beta}^*\|_1 \left(\frac{1}{n}\|\boldsymbol{\varepsilon}\|_1 + 10\delta\|\boldsymbol{\beta}^*\|_1\right).$$

$\square$

Proof of Proposition 9. For completeness, we provide proof for the proposition. The inequality can be rewritten as: $-y^2 + by + c \geq 0$ for $b, c \geq 0$. For this type of inequality, it follows that:

$$y \leq \frac{b + \sqrt{b^2 + 4c}}{2}.$$

Therefore:

$$y^2 \leq \left(\frac{b + \sqrt{b^2 + 4c}}{2}\right)^2 \overset{(a)}{\leq} \frac{b^2 + (b^2 + 4c)}{2} = b^2 + 2c,$$

where (a) uses that: $(r + s)^2 \leq 2(r^2 + s^2)$ for any $r, s \geq 0$. $\square$

C.3 High-dimensional analysis

We follow the same development as [29, example 7.14].

Assumption. Assume $\boldsymbol{\varepsilon}$ has i.i.d. $\mathcal{N}(0, \sigma^2)$ entries and that the matrix $\boldsymbol{X}$ is fixed with maximum entry of $\boldsymbol{X}$ equals to R.

Satisfying the conditions of Theorem 3. Under our assumptions

$$\max_{j=1,\cdots,m} \frac{\|\boldsymbol{x}_j\|_2}{\sqrt{n}} \overset{(a)}{\leq} \max_{j=1,\cdots,m} \|\boldsymbol{x}_j\|_\infty \leq R.$$

where step (a) follows from the inequality between the norms. Now, $\|\frac{\boldsymbol{X}^\top \boldsymbol{\varepsilon}}{n}\|_\infty$ is a the maximum over p zero-mean Gaussian variable with variance at most $\frac{R^2 \sigma^2}{n}$. From standard Gaussian tail bounds:

$$\left\| \frac{\boldsymbol{X}^\top \boldsymbol{\varepsilon}}{n} \right\|_\infty \leq R\sigma \sqrt{\frac{2\log(p/\gamma)}{n}} \tag{S.18}$$

with probability greater than $1 - 2\gamma$. Hence, if we set $\lambda = KR\sigma\sqrt{\frac{\log p}{n}}$ for an appropriate constant K we will have with high probability that $\|\frac{\boldsymbol{X}^\top \boldsymbol{\varepsilon}}{n}\|_\infty \leq \lambda$, satisfying the the condition on Theorem 3.

Satisfying the conditions of Theorem 2. Now, we will analyze the condition for which the condition of Theorem 2 is satisfied. That is, what values of δ yield with high probability $\|\boldsymbol{X}^\top \boldsymbol{\varepsilon}\|_\infty \leq \delta\|\boldsymbol{\varepsilon}\|_1$. We have that

$$\mathbb{E}\left[\frac{1}{n}\|\boldsymbol{\varepsilon}\|_1\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[|\varepsilon_i|\right] \overset{(a)}{=} \sqrt{\frac{2}{\pi}}\sigma,$$

where step (a) relied on the fact that $|\varepsilon_i|$ is a rectified Gaussian variable. Moreover, since the rectified Gaussian is a sub-Gaussian variable with proxy-variance $2\sigma^2$, from a Hoeffding-type of bound

$$\frac{1}{n}\|\boldsymbol{\varepsilon}\|_1 \geq \left(\sqrt{\frac{2}{\pi}} - 2\sqrt{\frac{\log(1/\gamma)}{n}}\right)\sigma$$

with probability greater than $1 - 2\gamma$. We combine this result with the (S.18) to obtain that we can set: $\delta = KR\sqrt{\frac{\log p}{n}}$, for an appropriate constant K and we will (with high-probability) satisfy the condition $\|\boldsymbol{X}^\top \boldsymbol{\varepsilon}\|_\infty \leq \delta\|\boldsymbol{\varepsilon}\|_1$.

D Numerical experiments

Here we provide some additional descriptions of the numerical experiments.

1. **Isotropic Gaussian features** As mentioned in the main text, we consider Gaussian noise and co-variates: $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\boldsymbol{x}_i \sim \mathcal{N}(0, r^2 \boldsymbol{I}_p)$ and the output is computed as a linear combination of the features contaminated with additive noise: $y_i = \boldsymbol{x}_i^\mathsf{T}\boldsymbol{\beta} + \epsilon_i$. In the experiments, unless stated otherwise, we use the parameters $\sigma = 1$ and $r = 1$.

2. **Latent-space features model** The "latent space" feature model is described in Hastie *et al.* [26, Section 5.4]. The features $\boldsymbol{x}$ are noisy observations of a lower-dimensional subspace of dimension d. A vector in this *latent space* is represented by $\boldsymbol{z} \in \mathbb{R}^d$. This vector is indirectly observed via the features $\boldsymbol{x} \in \mathbb{R}^p$ according to

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{u},$$

where $\boldsymbol{W}$ is an $p \times d$ matrix, for $p \geq d$. We assume that the responses are described by a linear model in this latent space

$$y = \boldsymbol{\theta}^\top \boldsymbol{z} + \xi,$$

where $\xi \in \mathbb{R}$ and $\boldsymbol{u} \in \mathbb{R}^p$ are mutually independent noise variables. Moreover, $\xi \sim \mathcal{N}(0, \sigma_\xi^2)$ and $\boldsymbol{u} \sim \mathcal{N}(0, \boldsymbol{I}_p)$. We consider the features in the latent space to be isotropic and normal $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}_d)$ and choose $\boldsymbol{W}$ such that its columns are orthogonal, $\boldsymbol{W}^\top \boldsymbol{W} = \frac{p}{d}\boldsymbol{I}_d$, where the factor $\frac{p}{d}$ is introduced to guarantee that the signal-to-noise ratio of the feature vector $\boldsymbol{x}$ (i.e. $\frac{\|\boldsymbol{W}\boldsymbol{z}\|_2^2}{\|\boldsymbol{u}\|_2^2}$) is kept constant. In the experiments, unless stated otherwise, we use the parameters $\sigma_\xi = 1$ and the latent dimension fixed $d = 1$.

3. **Random Fourier features model** [39] The features are obtained by the nonlinear transformation $\boldsymbol{z} \mapsto \boldsymbol{x}$:

$$\boldsymbol{x} = \sqrt{\frac{2}{p}}\cos(\boldsymbol{W}\boldsymbol{z} + \boldsymbol{b}),$$

where each entry of W is independently sampled from a normal distribution $\mathcal{N}(0, \sigma_w)$ and each entry from b is sampled from a uniform distribution $\mathcal{U}[0, 2\pi)$ the pair. We apply the random Fourier feature map to inputs of the Diabetes dataset [18]. The outputs are kept unaltered. In the experiments, unless stated otherwise, we use the parameter $\sigma_w = 0.01$.

4. **Random projections model** [37] We consider Gaussian noise and covariates: $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and $x \sim \mathcal{N}(0, I_d)$ and the output is computed as a linear combination of the features contaminated with additive noise: $y = x^\top S^\top \beta + \epsilon$, where S is a random projection matrix of varying dimension. The entries of S are randomly sampled from Rademacher distribution as in the experiments in [37]. In the experiments, unless stated otherwise, we use the $\sigma = 1$.

5. **Phenotype prediction from genotype.** We consider Diverse MAGIC wheat dataset [40] from the National Institute for Applied Botany. The dataset contains the whole genome sequence data and multiple phenotypes for a population of 504 wheat lines. We use a subset of the genotype to predict one of the continuous phenotypes. We have integer input with values indicating whether each one of the 1.1 million nucleotides differs or not from the reference value. Closely located nucleotides tend to be correlated and we consider z a pruned version provided by [40]. To generate the features we subsample p from the sequence z, such that the input to the model is $x = Wz$, where W is a matrix containing ones or zeros, such that each row of $W \in \mathbb{R}^{p \times d}$ have p nonzero entries, i.e., $W\mathbf{1} = p\mathbf{1}$.

Examples 1, 2, 4 are synthetic datasets. Examples 3 and 5 are real datasets combined with a feature map strategy. We point out that example 4 requires a (slightly) different mathematical formulation, which we cover in Appendix A.4. The results for the random projection model are presented separately in Figure S.10. The other figures refer to examples 1, 2, 3, 5.
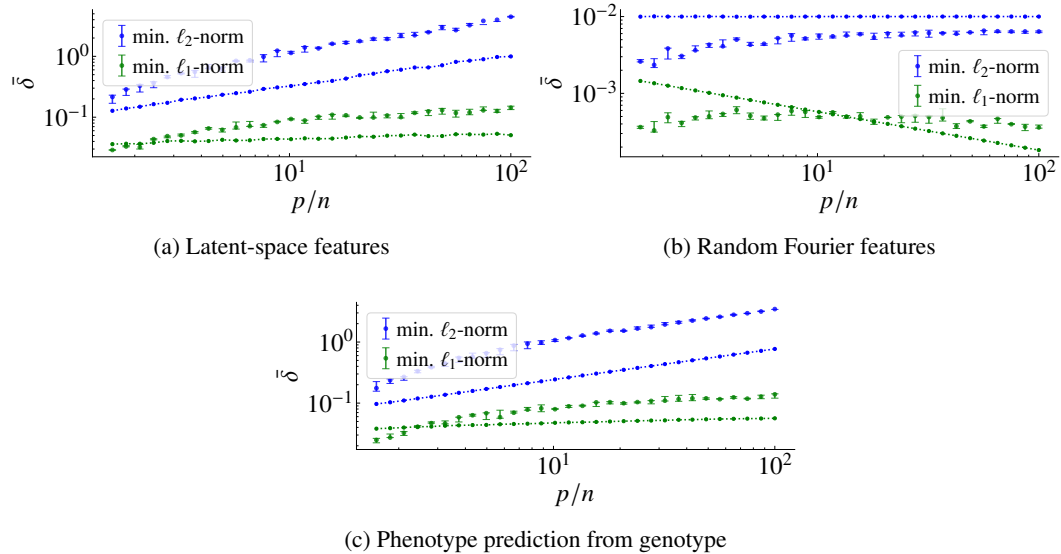

(a) Latent-space features

(b) Random Fourier features

(c) Phenotype prediction from genotype

Figure S.4: **Threshold $\bar{\delta}$ vs. number of features.**

(a) Gaussian features

(b) Latent-space features

(c) Random Fourier features
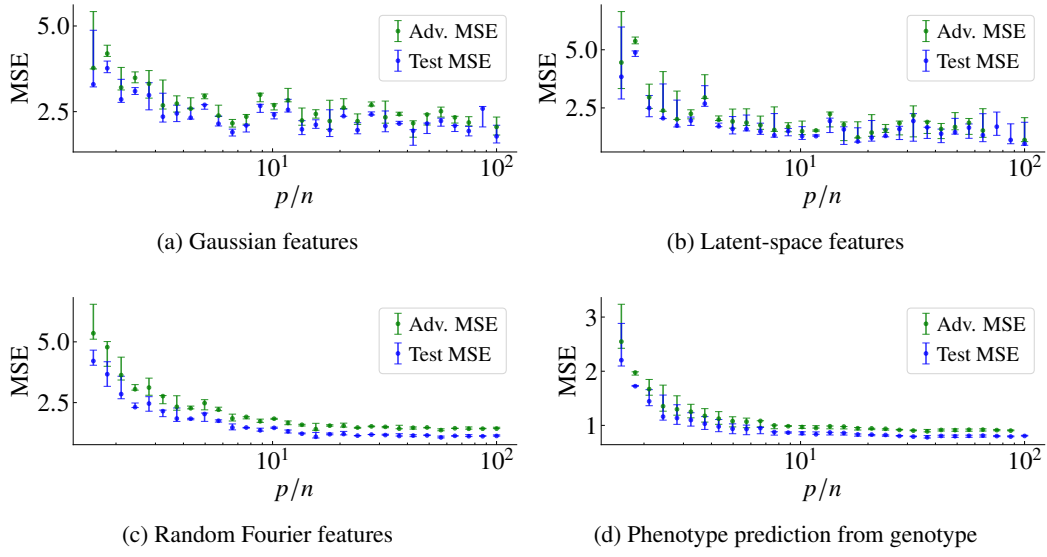
(d) Phenotype prediction from genotype

Figure S.5: (**Minimum ℓ_2-norm interpolator**) **MSE on test set *vs.* number of features.** We show both the MSE in the absence of an adversary (Test MSE), and in the presence of an ℓ_2-adversarial attack (Adv. MSE). The adversarial radius of the evaluation is $\delta_{\text{test}} = 0.01 \mathbb{E}\left[\|x\|_2\right]$
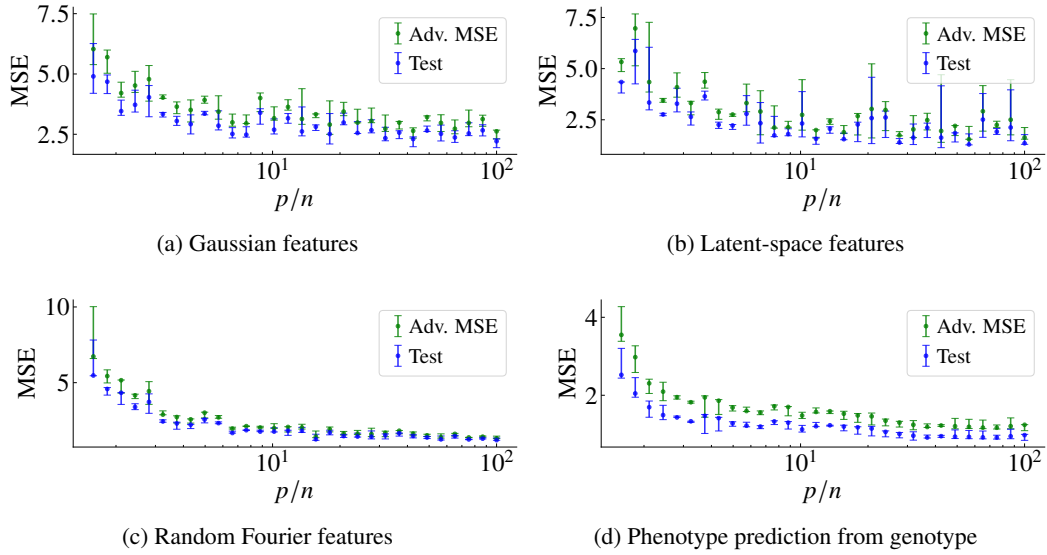

(a) Gaussian features

(b) Latent-space features

(c) Random Fourier features

(d) Phenotype prediction from genotype

Figure S.6: (**Minimum ℓ_1-norm interpolator**) **MSE on test set *vs.* number of features.** We show both the MSE in the absence of an adversary (Test MSE), and in the presence of an ℓ_∞-adversarial attack (Adv. MSE). The adversarial radius of the evaluation is $\delta_{\text{test}} = 0.01 \mathbb{E}\left[\|x\|_1\right]$
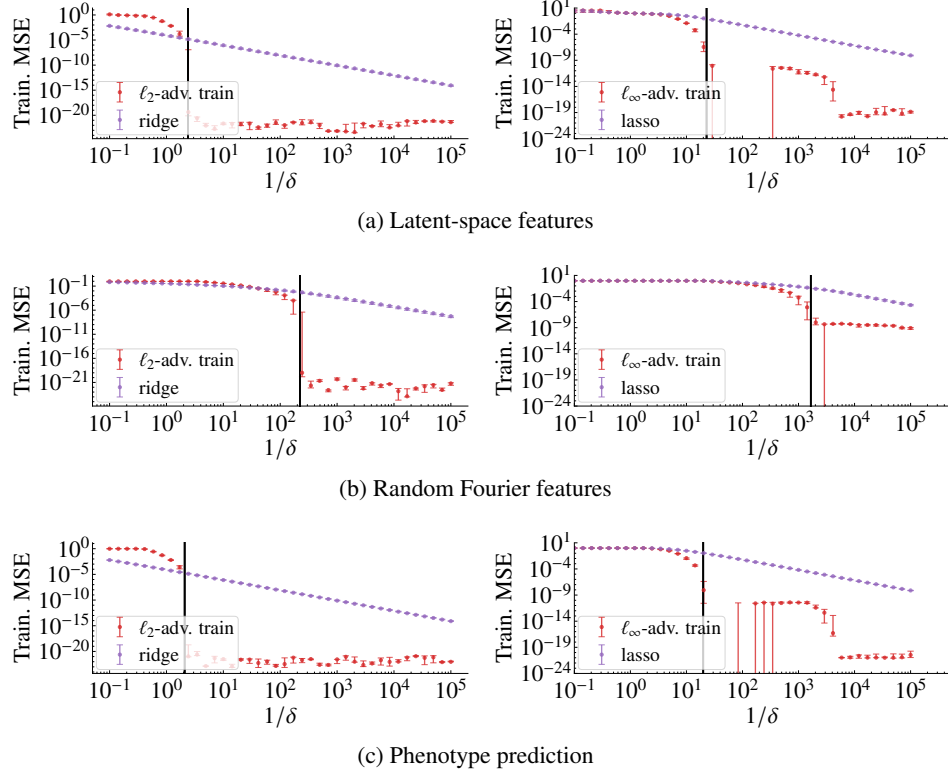
(a) Latent-space features

(b) Random Fourier features

(c) Phenotype prediction

Figure S.7: **Training MSE *vs* regularization parameter**. *Left:* for ridge and ℓ_2-adversarial training. *Right:* for Lasso and ℓ_∞-adversarial training The error bars give the median and the 0.25 and 0.75 quantiles obtained from numerical experiment (5 realizations).

(a) Gaussian features
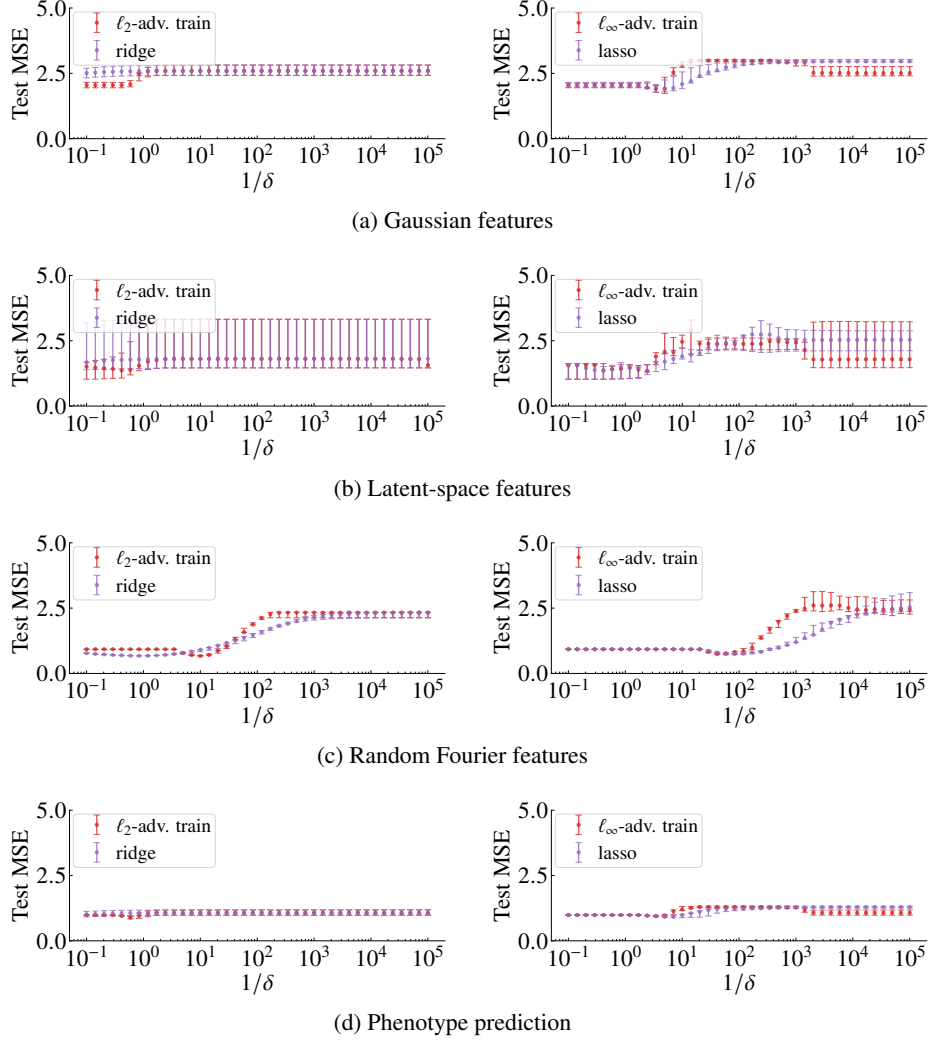
(b) Latent-space features

(c) Random Fourier features

(d) Phenotype prediction

Figure S.8: **Test MSE *vs* regularization parameter**. *Left:* for ridge and ℓ_2-adversarial training. *Right:* for Lasso and ℓ_∞-adversarial training. The error bars give the median and the 0.25 and 0.75 quantiles obtained from numerical experiment (5 realizations).
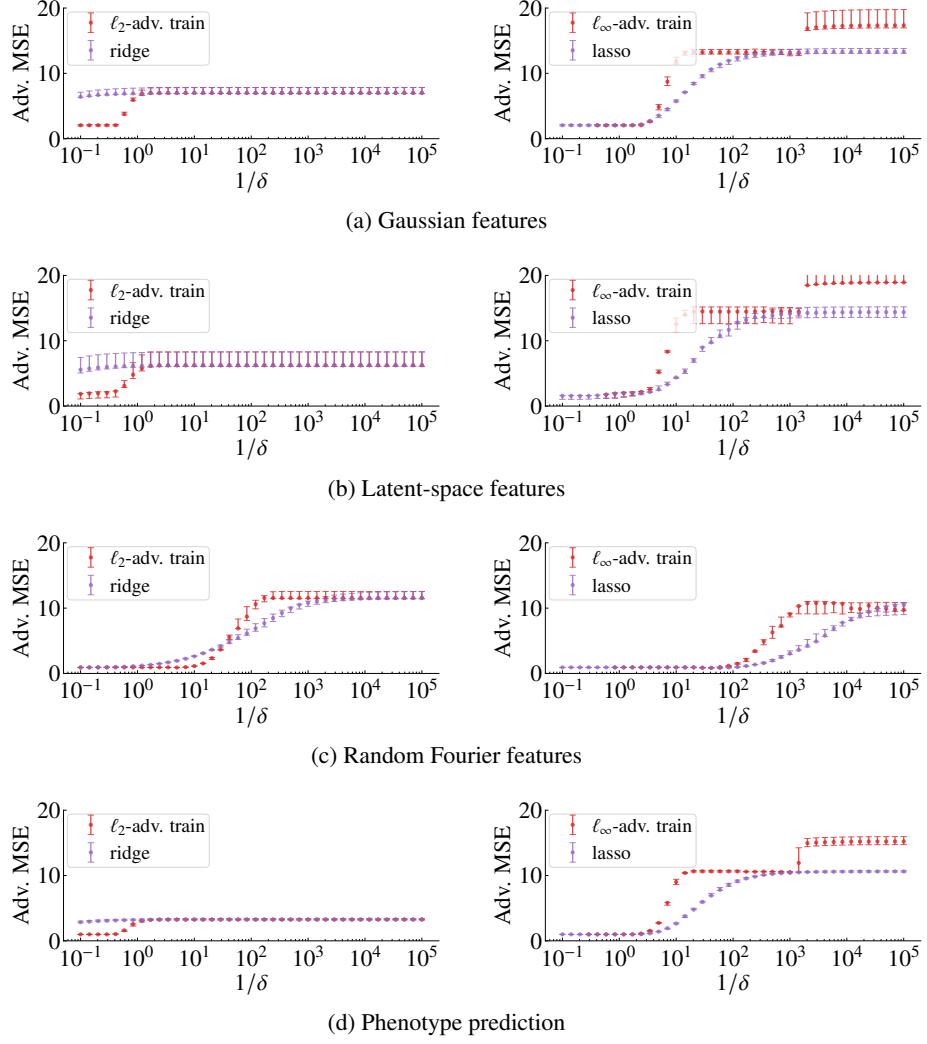
(a) Gaussian features

(b) Latent-space features

(c) Random Fourier features

(d) Phenotype prediction

Figure S.9: **Adversarial Test MSE *vs* regularization parameter**. *Left:* for ridge and ℓ_2-adversarial training. *Right:* for Lasso and ℓ_∞-adversarial training. The error bars give the median and the 0.25 and 0.75 quantiles obtained from numerical experiment (5 realizations).

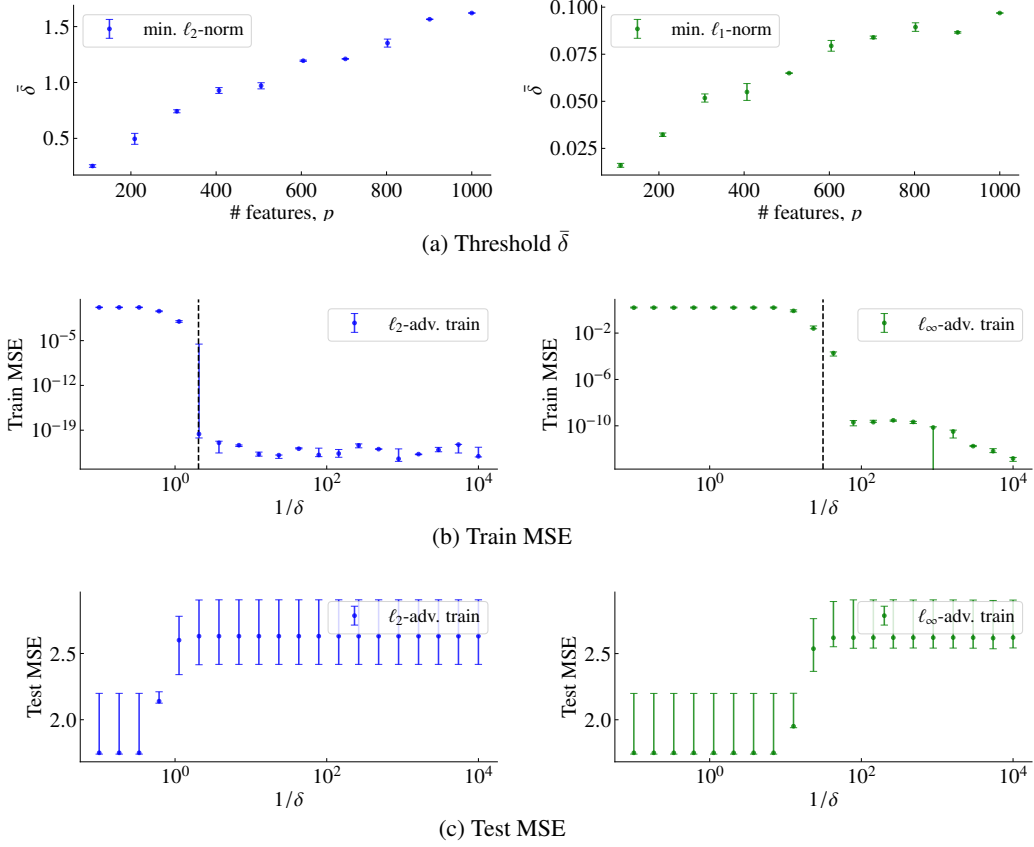(a) Threshold $\bar{\delta}$

(b) Train MSE

(c) Test MSE

Figure S.10: **Random projections** *Left:* the results for ℓ_∞-adversarial training. *Right:* the results for ℓ_2-adversarial attacks. In (a) we show the threshold as a function of the number of features. Unlike Figures 2 and S.4, we do not give a reference, that is because the input x is fixed, so it does make sense to consider δ in absolute terms. (b) the train MSE as a function of $1/\delta$ for the number of features fixed $p = 200$. (c) the test MSE as a function of $1/\delta$. We consider an input of dimension $d = 1000$.
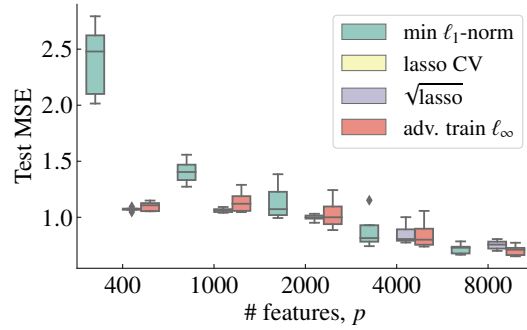

Figure S.11: **Test MSE (MSE) under fixed adversarial radius** in MAGIC dataset. We study a choice of adversarial radius inspired by Theorem 2. For ℓ_∞-adversarial training, we use $\delta = 0.5 \|X\xi\|_\infty / \|\xi\|_1$ for ξ a vector with zero-mean normal entries. We use a random ξ, since we do not know the true additive noise. Even with this approximation, ℓ_∞-adversarial training performs comparably with Lasso with the regularization parameter set using 5-fold cross-validation doing a full search among the hyperparameter space. We use a similar method for setting the square-root Lasso parameter, setting $\lambda = 0.1\|X\xi\|_\infty / \|\xi\|_2$. The value 0.5 and 0.1 were set empirically, after finding out that the value 3 used in the theorem was too conservative.

E   Results for general loss functions

E.1   Proof of Theorem 4

In the proof, we will use the Fenchel conjugate of L, where $L^* : \mathbb{R} \to \mathbb{R}$ is defined as:

$$L^*(u) \coloneqq \sup_z \{uz - L(z)\}$$

Proof. Using Fenchel–Moreau theorem the assumption on the loss imply that $L(z) = \sup_u \{uz - L^*(u)\}$ for all z. Let $z = \boldsymbol{x}^\top \boldsymbol{\beta}$,

$$
\begin{aligned}
\max_{\|\boldsymbol{\Delta x}\| \le \delta} \mathcal{L}((\boldsymbol{x} + \boldsymbol{\Delta x})^\top \boldsymbol{\beta}) &= \max_{\|\boldsymbol{\Delta x}\| \le \delta} \sup_u (u\boldsymbol{x}^\top \boldsymbol{\beta} + u\boldsymbol{\Delta x}^\top \boldsymbol{\beta} - L^*(u)) \\
&= \sup_u \left(u\boldsymbol{x}^\top \boldsymbol{\beta} + \max_{\|\boldsymbol{\Delta x}\| \le \delta} (u\boldsymbol{\Delta x}^\top \boldsymbol{\beta}) - L^*(u) \right) \\
&= \sup_u \left(u\boldsymbol{x}^\top \boldsymbol{\beta} + \delta \|\boldsymbol{\beta}\|_* |u| - L^*(u) \right) \\
&= \max_{s \in \{-1,1\}} \sup_u \left(u(\boldsymbol{x}^\top \boldsymbol{\beta} + s\delta \|\boldsymbol{\beta}\|_*) - L^*(u) \right)
\end{aligned}
$$

Applying Fenchel–Moreau theorem again we obtain the desired result. $\qquad\square$

E.2   Extension to linear maps

We consider here a matrix $\boldsymbol{S} \in \mathbb{R}^{p \times d}$ maps from the input space $\mathbb{R}^d$ to a feature space $\mathbb{R}^p$ (The setting described in Appendix A.4). And that the parameter vector is estimated in this features space. In this case, the following extension holds:

> **Theorem 7.** Let $\mathcal{L} : \mathbb{R} \to \mathbb{R}$ be convex and lower-semicontinuous, than for every δ
>
> $$\max_{\|\boldsymbol{\Delta x}\| \le \delta} \mathcal{L}((\boldsymbol{x} + \boldsymbol{\Delta x})^\top \boldsymbol{S}^\top \boldsymbol{\beta}) = \max_{s \in \{-1,1\}} \mathcal{L}\left(\boldsymbol{x}^\top \boldsymbol{S}^\top \boldsymbol{\beta} + \delta s \|\boldsymbol{S}^\top \boldsymbol{\beta}\|_* \right). \qquad (S.19)$$

the proof follows the same steps and is omitted here.

E.3   Application to dimensionality reduction

In this example, we discuss how to formulate an adversarial dimensionality reduction algorithm from principal component analysis (PCA) loss function. PCA finds a projection matrix $\boldsymbol{P}$ that transforms a given input $\boldsymbol{x} \in \mathbb{R}^m$ into a lower-dimensional representation $\boldsymbol{z} = \boldsymbol{P}\boldsymbol{x} \in \mathbb{R}^d$, with $d < m$. Conversely, given a lower-dimensional representation, the original input space can be reconstructed with the inverse transformation $\hat{\boldsymbol{x}} = \boldsymbol{P}^\top \boldsymbol{z}$. Principal components analysis can be formulated as the minimization of

$$\frac{1}{n} \sum_{i=1}^n \|\boldsymbol{x}_i + \boldsymbol{P}_d \boldsymbol{P}_d^\top \boldsymbol{x}_i\|_2^2. \qquad (S.20)$$

for $\boldsymbol{P} \in \mathbb{R}^{m \times d}$ a matrix with orthonormal columns $\boldsymbol{P}_d = [\boldsymbol{p}_1, \cdots, \boldsymbol{p}_d]$. Alternatively, it can be viewed as sequentially minimizing:

$$\frac{1}{n} \sum_{i=1}^n \|\tilde{\boldsymbol{x}}_i + \boldsymbol{p}_d \boldsymbol{p}_d^\top \tilde{\boldsymbol{x}}_i\|_2^2 \quad \text{subject to} \quad \boldsymbol{P}_{(d-1)} \boldsymbol{p}_d = 0 \text{ and } \|\boldsymbol{p}_d\|_2 = 1 \qquad (S.21)$$

where $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{P}_{(d-1)} \boldsymbol{P}_{(d-1)}^\top \boldsymbol{x}_i$. Moreover, minimizing (S.21) is equivalent to minimize $\frac{1}{n} \sum_{i=1}^n \mathcal{L}(\tilde{\boldsymbol{x}}_i^\top \boldsymbol{p}_d)$ for $\mathcal{L}(\tilde{\boldsymbol{x}}^\top \boldsymbol{p}_d) = -(\tilde{\boldsymbol{x}}^\top \boldsymbol{p}_d)^2$. With the formulation of PCA we just described, the adversarial extension follows naturally. One can obtain that for this loss function $s_* = -\mathrm{sign}(\tilde{\boldsymbol{x}}^\top \boldsymbol{p}_d)$ and that

$$\max_{\|\boldsymbol{\Delta x}\| \le \delta} \|(1 + \boldsymbol{p}_d \boldsymbol{p}_d^\top)(\tilde{\boldsymbol{x}} + \boldsymbol{\Delta x})\|_2^2 = -\left(|\tilde{\boldsymbol{x}}^\top \boldsymbol{p}_d| - \delta \|\boldsymbol{p}_d\|_* \right)^2.$$

For the special case of ℓ_2-adversarial disturbance, $\|\boldsymbol{p}_d\|_2 = 1$ hence:

$$\max_{\|\boldsymbol{\Delta x}\|_2 \leq \delta} \|(1 + \boldsymbol{p_d p_d}^\top)(\tilde{\boldsymbol{x}} + \boldsymbol{\Delta x})\|_2^2 = -\left(|\tilde{\boldsymbol{x}}^\top \boldsymbol{p_d}| - \delta\right)^2 .$$