On the Generalization Properties of Adversarial Training

Yue Xing
Department of Statistics
Purdue University
West Lafayette, IN 47907, USA

Qifan Song
Department of Statistics
Purdue University
West Lafayette, IN 47907, USA

Guang Cheng
Department of Statistics
Purdue University
West Lafayette, IN 47907, USA

Abstract

Modern machine learning and deep learning models are shown to be vulnerable when testing data are slightly perturbed. Theoretical studies of adversarial training algorithms mostly focus on their adversarial training losses or local convergence properties. In contrast, this paper studies the generalization performance of a generic adversarial training algorithm. Specifically, we consider linear regression models and two-layer neural networks (with lazy training) using squared loss under both low-dimensional and high-dimensional regimes. In the former regime, the adversarial risk of the trained models will converge to the minimal adversarial risk. In the latter regime, we discover that data interpolation prevents the adversarial robust estimator from being consistent (i.e. converge in probability). Therefore, inspired by successes of the least absolute shrinkage and selection operator (LASSO), we incorporate the $\ell_1$ penalty in the high dimensional adversarial learning, and show that it leads to consistent adversarial robust estimation in both theory and numerical trials.

1 Introduction

Recent advances in deep learning and machine learning have led to breakthrough performance and are widely applied in practice. However, empirical experiments show that deep learning models can be fragile and vulnerable against adversarial input which is intentionally perturbed (Biggio et al., 2013; Szegedy et al., 2014). For instance, in image recognition, a deep neural network will predict a wrong label when the testing image is slightly altered, while the change is not recognizable by the human eye (Papernot et al., 2016b). To ensure the reliability of machine learning and deep learning when facing real-world inputs, the demand on the robustness is increasing. The related research efforts in adversarial learning include designing adversarial attacks in various applications (Papernot et al., 2016b,a; Moosavi-Dezfooli et al., 2016), detecting attacked samples (Tao et al., 2018; Ma and Liu, 2019), and modifications on the training process to obtain adversarial robust models, i.e., adversarial training (Shaham et al., 2015; Madry et al., 2018; Jalal et al., 2017; Balunovic and Vechev, 2020).

To introduce adversarial training, let $l$ denote the loss function and $f_{\theta}(x)$ be the model with parameter $\theta$. The (population) adversarial loss is defined as

$$R_f(\theta, \epsilon) := \mathbb{E} [ l(f_{\theta}[x + A_{\epsilon}(f_{\theta}, x, y)], y) ] ,$$

Preprint. Under review.
where $A_{\epsilon}$ is an attack of strength $\epsilon > 0$ and intends to deteriorate the loss in the following way

$$A_{\epsilon}(f_\theta(x), y) := \text{argmax}_{z \in \mathcal{R}(0,\epsilon)} \{l(f_\theta(x + z), y)\}. \quad (1)$$

In the above, $z$ is subject to the constraint $\mathcal{R}(0,\epsilon)$, i.e., an $L_2$ ball centered at 0 with radius $\epsilon$.

Given $n$ i.i.d. samples $\{(x_i, y_i)\}_{i=1}^n$, the adversarial training aims to minimize an empirical version of $R_f(\theta, \epsilon)$ w.r.t. $\theta$:

$$\hat{R}_f(\theta, \epsilon) = \frac{1}{n} \sum_{i=1}^n l(f_\theta[x_i + A_\epsilon(f_\theta(x_i, y_i)], y_i)]$$

and

$$\theta = \text{argmin}_{\theta} \hat{R}_f(\theta, \epsilon). \quad (2)$$

The minimization in (2) is often implemented through an iterative two-step (min-max) update. In the $t$-th iteration, we first calculate the adversarial sample $x_i^{(t)} = x_i + A_\epsilon(f_{\theta_{t-1}}, x_i, y_i)$ based on the current $\theta_{t-1}$, and then update $\theta_{t+1}$ based on the gradient of the adversarial training loss while fixing $x_i^{(t)}$'s; see Algorithm 1. This generic algorithm and its variants have been studied in Shaham et al., 2015; Madry et al., 2018; Jalal et al., 2017; Balunovic and Vechev, 2020; Sinha et al., 2018; Wang et al., 2019 among others. Note that for some loss function $l$ or model $f_\theta$, there may not be an analytic form for $A_{\epsilon}$ (e.g., deep neural networks). In this case, an additional iterative optimization is needed to calculate $A_{\epsilon}$ at each step; see Wang et al., 2019.

**Algorithm 1 A General Form of Adversarial Training**

**Input:** data $(x_1, y_1), ..., (x_n, y_n)$, attack strength $\epsilon$, number of steps $T$, initialization $\theta^{(0)}$, step size $\eta$.

**for** $t = 1$ to $T$ **do**

**for** $i = 1$ to $n$ **do**

Calculate the attack for the $i$th sample and get $x_i^{(t-1)} = x_i + A_\epsilon(f_{\theta_{t-1}}, x_i, y_i)$.

**end for**

Fixing $x_i^{(t-1)}$'s, update $\theta^{(t)}$ from $\theta^{(t-1)}$ through

$$\theta^{(t)} = \theta^{(t-1)} - \eta \nabla_\theta \hat{R}_f(\theta^{(t-1)}, \epsilon) = \theta^{(t-1)} - \eta \nabla_\theta \left[ \frac{1}{n} \sum_{i=1}^n l(f_{\theta_{t-1}}(x_i^{(t-1)}), y_i) \right]. \quad (3)$$

**end for**

**Output:** $\theta^{(T)}$.

In the literature, there are three strands of theoretical studies related to this work. The first strand studies statistical properties or generalization performance of adversarial robust estimators. Javanmard et al. (2020) focuses on the statistical properties of $R_f(\hat{\theta}, \epsilon)$, without specifying how to obtain the exact/approximate global minimizer $f_{\hat{\theta}}$. Some other works concerned with the generalization bounds of adversarial loss (Yin et al., 2019; Raghunathan et al., 2019), or how to improve the generalization performance (Schmidt et al., 2018; Najafi et al., 2019; Zhai et al., 2019; Hendrycks et al., 2019). The second strand is interested in the adversarial training loss, say $\hat{R}_f(\theta^{(t)}, \epsilon)$ for $t = 1, ..., T$. For instance, Gao et al. (2019); Zhang et al. (2020) showed that for over-parameterized neural networks, the empirical adversarial loss can be arbitrarily close to the minimum value in a local region near initialization. The third strand studies the convergence of adversarial training (e.g., Sinha et al., 2018; Wang et al., 2019) in terms of local optimum. In summary, most literature focused on either the generalization performance $R_f(\hat{\theta}, \epsilon)$ by assuming that $\hat{\theta}$ is attainable, the training performance $\hat{R}_f(\theta^{(T)}, \epsilon)$ or $\hat{R}_f(\theta^{(t)}, \epsilon)$, or $R_f(\theta^{(T)}, \epsilon) - \hat{R}_f(\theta^{(T)}, \epsilon)$ for local optima. Recently, Allen-Zhu and Li (2020) studied both training performance and (standard/adversarial) testing performance of adversarial training in two-layer ReLU network when data is generated from a sparse coding model. However, their theorems assume $\epsilon$ is small and converges to zero as $d \to \infty$.

In this paper, we focus on the generalization ability of the adversarial training procedure in Algorithm 1 by studying $\hat{R}_f(\theta^{(T)}, \epsilon)$. Specifically, we consider linear regression as the generative model:

$$y = \theta_0^T x + \epsilon, \quad (4)$$
where $x$ is a $d$-dimensional Gaussian vector with mean 0 and variance $\Sigma (\theta_0$ is not perpendicular to $\Sigma$), and $\varepsilon$ is a Gaussian noise (independent of $x$) with variance $\sigma^2 < \infty$. As $d$ diverges, we assume both maximum and minimum eigenvalues of $\Sigma$ are finite and bounded away from 0. In addition, $\|\theta_0\|$ and $\sigma^2$ are allowed to increase in $d$, but the signal-to-noise ratio $\|\theta_0\|_2 / \sigma$ is large, say bounded away from zero.

We consider two working models: linear regression and two-layer neural networks, and assume the squared loss $l(f_\theta(x), y) = (f_\theta(x) - y)^2$ under the $\mathcal{L}_2$ attack. It is shown that the adversarial loss $R_{\ell_2}(\theta, \varepsilon)$ is not differentiable at the origin, which motivates us to propose a surrogate loss indexed by $\xi$. Under the low dimensional setup, say $d/n \to 0$, we show that under proper algorithm and network conditions, the iterative estimate $\theta^{(T)}_t$ trained from the surrogate adversarial loss asymptotically achieves the minimum adversarial risk.

However, under the high-dimensional setup where $d/n \to \infty$, the adversarial training acts rather differently. Specifically, we show that for any $\xi > 0$, due to the existence of data interpolation, $\hat{R}_{\ell_2}(\theta^{(T)}_0, \varepsilon)$ goes to zero, and $R_{\ell_2}(\theta^{(T)}_0, \varepsilon)$ converges to the adversarial risk of the null model (i.e., $R_{\ell_2}(0, \varepsilon)$). As a result, there always exists a gap between $R_{\ell_2}(\theta^{(T)}_0, \varepsilon)$ and the minimal adversarial risk.

To close the gap, we penalize the (surrogate) adversarial training loss using LASSO. The resulting adversarial robust estimator and adversarial risk are both consistent when $\theta^*$ defined in (6) is sparse.

## 2 Low dimensional asymptotics

In this section, we consider linear regression models and two-layer neural network models (training the first layer weights) under the low dimensional scenario. In particular, we examine the landscape of the adversarial loss, and further investigate the testing performance of the estimator $\theta^{(T)}_t$. In what follows, we rewrite $R_{\ell_2}$ as $R_L$ for linear models and as $R_N$ for two-layer neural networks.

### 2.1 Linear regression model

Consider the linear regression model: $f_\theta(x) = \theta^\top x$. By the definition of $A_\varepsilon(f_\theta, x, y)$ under the $\mathcal{L}_2$ ball constraint and the fact that $x$ and $\varepsilon$ are both Gaussian, $R_L$ has an analytical form as

$$
R_L(\theta, \varepsilon) = \|\theta - \theta_0\|_2^2 + \sigma^2 + 2c_0\|\theta\|_2 \sqrt{\|\theta - \theta_0\|_2^2 + \sigma^2},
$$

(5)

where $\|a\|_2 := a^\top \Sigma a$, $\|a\|_2 := \|a\|_2$ for any vector $a$, and $c_0 := \sqrt{2/\pi}$. Given any $\varepsilon \geq 0$, define

$$
\theta^*(\varepsilon) := \arg\min_{\theta} R_L(\theta, \varepsilon) \quad \text{and} \quad R^*(\varepsilon) := \min_{\theta} R_L(\theta, \varepsilon).
$$

(6)

When no confusion arises, we will rewrite $\theta^*(\varepsilon)$ and $R^*(\varepsilon)$ as $\theta^*$ and $R^*$ for simplicity.

We first analyze $R_L(\theta, \varepsilon)$ based on the form (5).

**Proposition 1.** Assume the attack strength $\varepsilon > 0$ and the data generation follows (4). The adversarial loss $R_L$ is differentiable w.r.t. $\theta$ when (1) $\sigma^2 > 0$ and $\theta \neq 0$, or (2) $\sigma = 0$, $\theta = 0$, and $\theta \neq \theta_0$. In addition, $R_L$ is convex w.r.t. $\theta$.

From Proposition 1, there exists some $\theta$ such that $R_L(\theta, \varepsilon)$ is not differentiable (e.g., $\theta = 0$). This makes it difficult to track the trajectory of gradient descent algorithms. Therefore, for both models under consideration, we introduce $A_{\varepsilon, \xi}$ that approximates the adversarial attack $A_{\varepsilon}$:

$$
A_{\varepsilon, \xi}(f_\theta, x, y) = \|\partial_l f_\theta(x, y) / \partial x\| \sqrt{\|\partial_l f_\theta(x, y) / \partial x\|^2 + \xi^2} A_\varepsilon(f_\theta, x, y).
$$

When $\xi = 0$, $A_{\varepsilon, \xi}$ is reduced to $A_\varepsilon$. With $A_{\varepsilon, \xi}$, we define the empirical and population versions of surrogate loss as:

$$
\hat{R}_{L, \xi}(\theta, \varepsilon) := \frac{1}{n} \sum_{i=1}^n l(f_\theta(x_i + A_{\varepsilon, \xi}(f_\theta, x_i, y_i), y_i) \quad \text{and} \quad R_{L, \xi}(\theta, \varepsilon) := \mathbb{E} l(f_\theta(x_i + A_{\varepsilon, \xi}(\theta, x_i, y_i), y_i).
$$

In this case, $R_{L, \xi}(\theta, \varepsilon)$ is smooth and convex everywhere. Accordingly, Algorithm 1 is modified by replacing $A_\varepsilon$ with $A_{\varepsilon, \xi}$. Note that a smaller value of $\xi$, which leads to a closer approximation to $R_{L, 0}$,
We consider the two-layer neural network with an activation function \( \phi \), say
\[
f_\theta(x) = \frac{1}{\sqrt{h}} \sum_{j=1}^h \phi(x^T \theta_j) a_j,
\]
where \( a_j \)'s are known values and \( \theta = (\theta_1, ..., \theta_h) \in \mathbb{R}^{d \times h} \) is the quantity to be trained, i.e., lazy training. We adopt a vanishing initialization scheme (Ba et al., 2020):
\[
\theta^{(0)}_{\xi,j} \sim N(0, I_d/d^{1+\delta})
\]
for some \( \delta > 0 \).

Remark 1. An alternative way to deal with the non-differentiability of \( R_L(\theta, \epsilon) \) at \( \theta = 0 \) is to use the sub-gradient method with a changing learning rate (e.g. Boyd et al., 2003). However, theoretical analysis of sub-gradient descent requires stringent conditions, such as the loss function is Lipschitz continuous, which usually do not hold for \( R_L(\theta, \epsilon) \) for \( \theta \in \mathbb{R}^d \).

Theorem 1. Assume the data generation follows (4) and let \( \epsilon \) be a fixed constant. If there exists some constant \( B_0 \) such that \( \|\theta^{(0)}_\xi\| \leq B_0 \epsilon \), and the dimension growth rate satisfies \( \log n/\sqrt{d/n} \to 0 \), then with probability tending to 1, \( R_{L,\xi}(\theta^{(t)}_\xi, \epsilon) \) reduces in each iteration, and
\[
\frac{R_{L,\xi}(\theta^{(t)}_\xi, \epsilon) - R^*}{v^2} \to 0, \quad \text{and} \quad \frac{\|\theta^{(t)}_\xi - \theta^*\|}{v} \to 0,
\]
given \( v = \xi/(vL) \) for some large constant \( L \), \( T = (v \log n)/\xi \) and \( \xi = v\sqrt{d/n} \log n \).

The proof of Theorem 1 is postponed to Appendix C. The idea is that, under the low-dimensional regime, the sample gradient converges to its expectation in probability in each iteration. So when \( T \) is not so large (in the sense that it diverges more slowly than \( n \)), such a convergence holds uniformly for all iterations up to \( T \). Consequently, \( \theta^{(T)}_\xi \) converges to \( \theta^* \) asymptotically.

The results of Theorem 1 actually hold for a wide range of \( (\xi, \eta, T) \) (as elaborated in the Appendix C). In general, the choice of \( (\eta, T) \) can be invariant to \( v \), while a larger \( v \) implies a wider range of possible \( \xi \). In terms of tuning \( \theta^{(0)}_\xi \) and \( \xi \), we can estimate \( v^2 \) using the sample variance of \( y_1, ..., y_n \). Note that the above discussions apply to Theorems 2, 3, 4, 5, and 6 as well.

Additionally, Theorem 1 confirms the phenomenon that “adversarial training hurts standard estimation” (Raghunathan et al. (2019)) as follows. Denote \( \hat{\theta}_{\text{OLS}} \) as the ordinary least square estimator for uncorrupted data (i.e. without attack), and denote \( \theta^{(t)}_{\xi}(\epsilon) \) as the \( t \)-th iterative estimate under the attack strength \( \epsilon \). Theorem 1 implies that the standard risk of \( \theta^{(T)}_\xi(\epsilon) \) is strictly larger than that of \( \hat{\theta}_{\text{OLS}} \), i.e.
\[
\frac{R_L(\theta^{(T)}_\xi(\epsilon), 0) - R_L(\hat{\theta}_{\text{OLS}}, 0)}{v^2} \to c(\epsilon) > 0,
\]
where the function \( c(\epsilon) \) increases in \( \epsilon \), and converges to \( \|\theta_0\|_2^2/v^2 \) as \( \epsilon \) diverges.

Remark 2. It is noteworthy to point out that Theorem 1 is different from the existing literature reviewed in the introduction section. Theorem 1 works on the generalization performance of \( \theta^{(T)}_\xi \), while other studies focused on either performance of adversarial training, or testing performance assuming that \( \hat{\theta} \) is attainable.

### 2.2 Two-layer neural networks

We consider the two-layer neural network with an activation function \( \phi \), say
Theorem 3. Under the generative model (4), assume the activation function \( \phi \) in model (7) is twice continuously differentiable, \( \phi'(0) \neq 0 \), and \( \phi(0) = 0 \). Take \( \xi/v = -\log \log n/\log((d \log n)/\nu \lor \sqrt{(d \log n)/\nu}) \). Set \( \eta = \nu h/(vL||a||^2) \) for some constant \( L \). If \( a \) and \( h \) satisfy \( ||a||_\infty = O(1) \), and \( (d \log n)||a||_\infty \nu \sqrt{h}/||a||^2 \rightarrow 0 \), then with probability tending to 1, for \( T = (v \log \log n)/\xi \), we have

\[
\frac{R_{N,\xi}(\theta^{(T)}_\xi, \epsilon) - R^*}{\nu^2} \rightarrow 0, \tag{8}
\]

where \( R^* \) is the exactly the same as that in Theorem 1.

The detailed proof is postponed to Appendix D.

Remark 3. The proof of Theorem 2 is similar to Ba et al. (2020): as the number of hidden nodes \( h \) grows, the trajectories of optimization using linear network (with zero initialization) and nonlinear network (with vanishing initialization) are slightly different, while the convergence result of the former one can be simply extended from linear models. Different from Ba et al. (2020), we specify the learning rate as well as the number of iterations as functions of \( d, a, n, h \), while Ba et al. (2020) utilized gradient flow, which is not applicable in our setup. In addition, compared with Ba et al. (2020), the relationship of \( \xi, a, \eta \) is revealed in our result when \( \nu = O(1) \) and we do not require \( a \) to be generated from \( Unif(-1, 1) \).

Similarly, when \( \phi \) represents ReLU activation function, we have the following result:

Theorem 3. Under the generative model (4), assume the activation function \( \phi \) is ReLU function with no bias. Take \( \xi/v = -\log \log n/\log((d \log n)/\nu \lor \sqrt{(d \log n)/h^2}) \). Set \( \eta = \nu h/(vL||a||^2) \) for some constant \( L \). Denote \( a^+ \) as a vector such that \( a^+_j = a_j \{a_j > 0\} \), and similarly define \( a^- \). If \( a \) satisfy \( ||a^+||/||a^-|| = 1 \) and \( ||a||_\infty = O(1) \), then with probability tending to 1, for \( T = (v \log \log n)/\xi \), (8) holds.

Compared with Theorem 2, the constrains on \( a \) and \( h \) in Theorem 3 are weaker, except for the extra condition on \( ||a^+||/||a^-|| \). To explain such a difference, the ReLU-activate neural network can be treated as two (partial) linear networks for \( a^+ \) and \( a^- \) respectively, thus we do not need \( \theta_j \) to be close enough to zero (which is needed in smooth \( \phi \) for Taylor expansion). However, the two linear networks should be balanced, so we require \( ||a^+||/||a^-|| = 1 \).

Remark 4. Due to page limit, numerical results of linear model, linear network, non-linear network with sigmoid / ReLU activation under low dimensional settings are postponed to Appendix B.1.

3 High dimensional asymptotics

In this section, we focus on the high dimensional regime where \( d/n \rightarrow \infty \). It is first revealed that the adversarial training suffers from the classical interpolation effect, i.e., near-zero (surrogate) adversarial training loss but high generalization error. As a potential remedy, we penalize the adversarial training loss using LASSO. When \( \theta^* \) is sparse, the resulting estimate is consistent.

3.1 Effect of interpolation in adversarial training

It is well known that interpolation may occur under high dimensionality. For instance of linear regression, if a gradient descent with zero initialization is applied to minimize the squared loss when \( d/n \rightarrow \infty \), then the solution converges to

\[
\theta(y) := X^T (XX^T)^{-1} y,
\]

where \( X = (x_1, x_2, ..., x_n)^T \) and \( y = (y_1, y_2, ..., y_n)^T \), given a sufficiently small learning rate. This perfectly interpolated estimator \( \theta(y) \) is proven to be inconsistent to \( \theta_0 \) and lead to a large generalization error in Hastie et al. (2019); Belkin et al. (2019).

Our first result shows that \( \theta(y) \) also induces the same effect of interpolation in adversarial learning.
Lemma 1. Assume data generation follows (4). When \( \theta(\mathbf{y}) \neq 0 \) and \( d/n \to \infty \), we have \( \|\theta(\mathbf{y})\|^2/v^2 = O_p(n/d) \), and with probability tending to 1, for any \( \xi \geq 0 \), it holds that
\[
\hat{R}_{L,\xi}(\theta(\mathbf{y}), \mathbf{e}) \to 0 \quad \text{and} \quad R_{L,\xi}(\theta(\mathbf{y}), \mathbf{e}) \to 1.
\]

We next show that \( \theta^{(T)}(\xi) \) shares the same properties as \( \theta(\mathbf{y}) \). The core idea is that the training trajectory \( \{\theta^{(t)}(\xi)\}_{t=1}^{\infty} \) can be sufficiently close to that in the standard training, when both are initialized from zero. Since the latter converges to \( \theta(\mathbf{y}) \), the surrogate adversarial training loss and testing loss of \( \theta^{(T)}(\xi) \) act in a similar way as those for \( \theta(\mathbf{y}) \) respectively.

Theorem 4. Under the same assumptions as in Lemma 1, when \( (\log n)/\sqrt{d/n} \to 0 \), take \( \eta \) small enough such that the largest eigenvalue of \( \eta \mathbf{X}^\top \mathbf{X} \) is smaller than 1. Use zero initialization, and denote \( T := \min\{t \in \mathbb{Z}^+ : ||\mathbf{X}\theta^{(t)} - \mathbf{y}||_2/(v\sqrt{n}) < 1/\sqrt{\log n}\} \), then with probability tending to 1, for any \( \xi > 0 \), we have \( T < \infty \) and
\[
\hat{R}_{L,\xi}(\theta^{(T)}(\xi), \mathbf{e}) \to 0, \quad \text{and} \quad R_{L,\xi}(\theta^{(T)}(\xi), \mathbf{e}) \to 1.
\]

The proof of Theorem 4 is postponed to Appendix E. Compared with Theorem 1, Theorem 4 no longer requires \( \xi \) to be associated with \( (d, n) \). A crucial reason for this difference is that under high dimensionality, when \( t \leq T \), the smoothness of \( \hat{R}_{L,\xi} \) along the training trajectory (i.e., the gradient of \( \hat{R}_{L,\xi}(\theta^{(t)}, \mathbf{e}) \)) is always dominated by a term that is only determined by the eigenvalues of high dimensional design matrix \( \mathbf{X} \), regardless of how small \( \xi \) is (refer to equation (14) in Appendix E for details). This is in contrast to the low dimensional case.

Theorem 4 shows that \( R_{L,\xi}(\theta^{(T)}(\xi), \mathbf{e})/v^2 \) does not converge to \( R^*/v^2 \). Similar results can be established for two-layer neural networks:

Theorem 5. For the two-layer neural network (7), under the same conditions on \( \phi \) as in Theorem 2, \( (\log n)/\sqrt{d/n} \to \infty \), take zero/vanishing initialization and \( \eta = \eta_{\text{linear}}, h/\|a\|^2 \). Assume \( \|a\| = O(1), \sqrt{\log(n)}\|a\|_\infty v/\sqrt{n}||a||_2^2 \to 0, \delta > (2 + c)\log_h(d) \) for some \( c > 0 \). Denote \( T := \min\{t \in \mathbb{Z}^+ : ||\mathbf{X}\theta^{(t)} - \mathbf{y}||_2/(v\sqrt{n}) < 1/\sqrt{\log n}\} \), then with probability tending to 1, for any \( \xi > 0 \), we have \( T < \infty \) and
\[
\hat{R}_{N,\xi}(\theta^{(T)}(\xi), \mathbf{e}) \to 0, \quad \text{and} \quad R_{N,\xi}(\theta^{(T)}(\xi), \mathbf{e}) \to 1.
\]

For ReLU network, we have the following result:

Theorem 6. For the two-layer neural network (7), \( (\log n)/\sqrt{d/n} \to \infty \), assume the activation function \( \phi \) is ReLU function with no bias. If \( \|a^+\|/\|a^-\| = 1, \|a\|_\infty = O(1), \) and \( \delta > (2 + c)\log_h(d) \) for some \( c > 0 \). Denote \( T := \min\{t \in \mathbb{Z}^+ : ||\mathbf{X}\theta^{(t)} - \mathbf{y}||_2/(v\sqrt{n}) < 1/\sqrt{\log n}\} \), then for any \( \xi > 0 \), with probability tending to 1, we have \( T < \infty \), and (9) holds.

A numerical experiment was conducted to verify the above theoretical result of Theorem 4. We choose \( n = 20, d = 1000, \) and \( \sigma^2 = 1 \). The underlying true model \( \theta_0 \) is all-zero except for its first 10 elements being 1. The attack intensity is \( \epsilon = 0, 0.01, 0.1 \). Learning rate is taken as 0.001 with zero initialization and \( \xi = 0.5 \). The curves in Figure 1 represent means of respective statistics, and the shaded areas represent mean ± one standard deviation, based on 100 replications. Figure 1 shows that, the surrogate adversarial training loss keeps decreasing to around zero for all the choices of \( \epsilon \), while the adversarial testing loss converges to some nonzero constant. Note that the three adversarial training losses in the middle plot of Figure 1 overlap with each other. The right plot in Figure 1 calculates \( ||\theta^{(t)}(\xi) - \theta^{(t)}(0)||/||\theta^{(t)}(0)|| \), which is tiny. It validates our argument that the trajectories between adversarial training and standard training are negligible. More experiments for larger \( d \) and \( \xi = 0 \) (with a change when \( \theta = 0 \)) are postponed to Appendix B.2.1. In addition, we also postpone experiments for neural networks to Appendix B.2.2.
Theorem 7. Assume data generation follows (4), \( \hat{\theta}_\varepsilon \) is sparse and \( \Sigma = I \). Take \( \xi/v \to 0 \), \( \lambda/v = o(1) \) and \( \lambda/\varepsilon \geq c\sqrt{s \log d/n} \) for some large constant \( c \) and \( a_n \to \infty \). If \( \varepsilon < \sqrt{\sigma^2 \log d/n} \), then \( \hat{\theta}_\varepsilon \neq 0 \), and with probability tending to 1, we have

\[
\frac{R_{L,\xi}(\hat{\theta}_\varepsilon, \epsilon) - R^*}{\sigma^2} \to 0 \quad \text{and} \quad \frac{\Vert \hat{\theta}_\varepsilon - \theta^* \Vert_1}{\lambda} \to 0.
\]

The proof of Theorem 7 is similar to the traditional LASSO analysis as in Bickel et al. (2009); Belloni and Chernozhukov (2013) but with an important modification. In the literature, the Hessian of the standard training loss, i.e., \( X^T X/n \), is usually required to satisfy the so-called restricted eigenvalue condition. However, in adversarial setting, the Hessian changes as \( \theta \), thus it takes some more steps to verify the above condition. From the condition of \( \lambda \) in Theorem 7, we only require \( (s, d, n) \) satisfying \( s \log d/n \to 0 \), which indicates \( s \) can slowly increase as well.
In the end, we conduct some empirical study to explore the potential application of LASSO in the adversarial training of neural networks. Similar experiments under large-sample regime can be found for adversarial training Sinha et al. (2018); Wang et al. (2019); Raghunathan et al. (2019), and pruning in adversarial training Ye et al. (2019); Li et al. (2020).

**Numerical experiment** We conduct a small numerical experiment to verify the performance of $L_1$ penalized adversarial training in a neural network. The program was modified from a repository in Github\footnote{https://github.com/louis2889184/pytorch-adversarial-training}. A simple two-layer neural network is constructed with 1024 hidden nodes and ReLU as activation. We use MNIST dataset to distinguish between digits 0 and 1, and randomly select some samples of 0 and 1 from the training dataset to create a high-dimension scenario. The attack level is set to be 3 in $L_2$ (the results for $L_\infty$ attack are postponed to Appendix B.3). We observe similar results when $\xi = 0$, and the details are postponed to Appendix B.3. We trained 2000 epochs to ensure the convergence of the algorithms and repeat the experiment for 30 times to draw a boxplot. After training 2000 epochs, for both $\lambda = 0$ (No penalty) and $0.001$ (LASSO), the training accuracies for clean data and adversarial data both reach 100%. The penalty $\lambda = 0.001$ was chosen such that the penalty is comparable with loss. The results are summarized in Figure 2.

![Figure 2: Comparison on standard (upper)/adversarial (lower) test accuracies between training with/without $L_1$ penalty under $L_2$ attack with $\xi = 10^{-4}$. Attack strength $\epsilon = 3$.](image)

For adversarial accuracy, as shown in Figure 2, the results with two different $\lambda$’s are significantly different. As a reference, we also plot the standard accuracy (i.e. prediction accuracy for un-corrupted data), even though the objective function minimized is the (penalized) adversarial training loss. Figure 2 shows, for both $\lambda = 0$ and 0.001, it approaches 99% quickly in $n$.

There are two gaps between our theorem and this numerical experiment in Section 3.2: first, Theorem 7 studies linear models while the empirical study works on a neural network; second, Theorem 7 focuses on the statistical properties of $\hat{\theta}_\xi$ but not $\theta^{(T)}_\xi$. The gaps are beyond the scope of this paper, and we leave them as future studies.

In addition to MNIST dataset, we also conduct experiment using CIFAR-10 with Resnet34. We use all 50,000 training samples and 10,000 testing samples in this task. The $L_\infty$ attack satisfies $\epsilon = 2/255$. We use fast gradient sign method (FGSM) in adversarial training and train 100 epochs. The mean / standard error of the resulting standard / adversarial testing accuracy are summarized in Table 1. From Table 1, when using $L_1$ penalty as regularization, both standard and adversarial testing accuracy are enhanced. On the other hand, introducing $\xi > 0$ into adversarial training makes the real attack length less than $\epsilon$, thus the standard testing accuracy is higher, while the adversarial testing accuracy is lower. Note that Theorem 7 reveals that LASSO results in a consistent estimator when $d = \exp(O(n))$, but it does not essential that LASSO outperforms when $d \gg n$. Whether LASSO is better depends on the data generation model as well as neural network structure.
Table 1: Adversarial training in CIFAR-10

| std / adv acc (%) | no LASSO          | LASSO (\(\lambda = 10^{-4}\)) |
|-------------------|-------------------|---------------------------------|
| \(\xi = 0\)      | 84.596(0.187) / 76.114(0.2424) | 87.072(0.139) / 79.33(0.305)    |
| \(\xi = 0.0001\) | 85.583(0.057) / 74.808(0.160)   | 87.735(0.181) / 78.253(0.356)   |

4 Conclusion and future works

In this paper, we study the convergence properties of adversarial training in linear models and two-layer neural networks (with lazy training). In the low-dimensional regime, using adversarial training with surrogate attack, the adversarial risk of the trained model converges to the minimal value. In high-dimensional regime, data interpolation causes the adversarial training loss close enough to zero, while the generalization is poor. One potential solution is to add \(L_1\) penalty in the adversarial training, which results in both consistent adversarial estimate and risk in high dimensional sparse models.

There are several future directions. First, the landscape of population risk may change (e.g. no unique risk minimizer) if the distributional assumption on \(x\) is relaxed. The relaxed assumption may better reflect the reality where \(x\) does not necessarily follow Gaussian distribution. Besides, we may focus to classification tasks as a future work. In regression, the adversarial robust model generally outputs smaller-in-magnitude predictions, which is not practical in classification. One may be interested in how adversarial training works in classification. Finally, as our main goal of the surrogate attack is to ensure the smoothness of the training process, and from numerical experiment it does reduce adversarial testing accuracy. Therefore, the surrogate attack should be improved so as to be applied in real application.

References

Allen-Zhu, Z. and Li, Y. (2020), “Feature Purification: How Adversarial Training Performs Robust Deep Learning,” arXiv preprint arXiv:2005.10190.

Ba, J., Erdogdu, M., Suzuki, T., Wu, D., and Zhang, T. (2020), “Generalization of two-layer neural networks: an asymptotic viewpoint,” in 8th International Conference on Learning Representations.

Bai, Z.-D. and Yin, Y.-Q. (2008), “Limit of the smallest eigenvalue of a large dimensional sample covariance matrix,” in Advances In Statistics, World Scientific, pp. 108–127.

Balunovic, M. and Vechev, M. (2020), “Adversarial training and provable defenses: bridging the gap,” in 8th International Conference on Learning Representations.

Belkin, M., Hsu, D., and Xu, J. (2019), “Two models of double descent for weak features,” arXiv preprint arXiv:1903.07571.

Belloni, A. and Chernozhukov, V. (2013), “Least squares after model selection in high-dimensional sparse models,” Bernoulli, 19, 521–547.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. (2009), “Simultaneous analysis of Lasso and Dantzig selector,” The Annals of Statistics, 37, 1705–1732.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013), “Evading adversarial attacks against machine learning at test time,” in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, pp. 387–402.

Boyd, S., Xiao, L., and Mutapcic, A. (2003), “Subgradient methods,” lecture notes of EE392o, Stanford University, Autumn Quarter, 2004, 2004–2005.

Gao, R., Cai, T., Li, H., Hsieh, C., Wang, L., and Lee, J. D. (2019), “Convergence of adversarial training in overparametrized neural networks,” in Advances in Neural Information Processing Systems, pp. 13009–13020.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. (2019), “Surprises in high-dimensional ridgeless least squares interpolation,” arXiv preprint arXiv:1903.08560.
Hendrycks, D., Lee, K., and Mazeika, M. (2019), “Using pre-training can improve model robustness and uncertainty,” in Proceedings of the 36th International Conference on Machine Learning, vol. 97 of Proceedings of Machine Learning Research, pp. 2712–2721.

Jalal, A., Ilyas, A., Daskalakis, C., and Dimakis, A. G. (2017), “The robust manifold defense: Adversarial training using generative models,”.

Javanmard, A., Soltanolkotabi, M., and Hassani, H. (2020), “Precise tradeoffs in adversarial training for linear regression,” arXiv preprint arXiv:2002.10477.

Li, B., Wang, S., Jia, Y., Lu, Y., Zhong, Z., Carin, L., and Jana, S. (2020), “Towards Practical Lottery Ticket Hypothesis for Adversarial Training,” arXiv preprint arXiv:2003.05733.

Ma, S. and Liu, Y. (2019), “Nic: Detecting adversarial samples with neural network invariant checking,” in Proceedings of the 26th Network and Distributed System Security Symposium.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018), “Towards deep learning models resistant to adversarial attacks,” in 6th International Conference on Learning Representations.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016), “Deepfool: a simple and accurate method to fool deep neural networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582.

Najafi, A., Maeda, S.-i., Koyama, M., and Miyato, T. (2019), “Robustness to adversarial perturbations in learning from incomplete data,” in Advances in Neural Information Processing Systems, pp. 5542–5552.

Papernot, N., McDaniel, P., Swami, A., and Harang, R. (2016a), “Crafting adversarial input sequences for recurrent neural networks,” in Military Communications Conference, MILCOM 2016-2016 IEEE, IEEE, pp. 49–54.

Papernot, N., McDaniel, P. D., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016b), “The limitations of deep learning in adversarial settings,” in IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016, IEEE, pp. 372–387.

Raghunathan, A., Xie, S. M., Yang, F., Duchi, J. C., and Liang, P. (2019), “Adversarial training can hurt generalization,” arXiv preprint arXiv:1906.06032.

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

Shaham, U., Yamada, Y., and Negahban, S. (2015), “Understanding adversarial training: Increasing local stability of neural nets through robust optimization,” arXiv preprint arXiv:1511.05432.

Sinha, A., Namkoong, H., and Duchi, J. C. (2018), “Certifying some distributional robustness with principled adversarial training,” in 6th International Conference on Learning Representations.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2014), “Intriguing properties of neural networks,” in 2nd International Conference on Learning Representations.

Tao, G., Ma, S., Liu, Y., and Zhang, X. (2018), “ Attacks meet interpretability: Attribute-steered detection of adversarial samples,” in Advances in Neural Information Processing Systems, pp. 7717–7728.

Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., and Gu, Q. (2019), “On the convergence and robustness of adversarial training,” in International Conference on Machine Learning, pp. 6586–6595.

Ye, S., Xu, K., Liu, S., Cheng, H., Lambrechts, J.-H., Zhang, H., Zhou, A., Ma, K., Wang, Y., and Lin, X. (2019), “Adversarial robustness vs. model compression, or both,” in The IEEE International Conference on Computer Vision (ICCV), vol. 2.
Yin, D., Ramchandran, K., and Bartlett, P. L. (2019), “Rademacher complexity for adversarially robust generalization,” 97, 7085–7094.

Zhai, R., Cai, T., He, D., Dan, C., He, K., Hopcroft, J., and Wang, L. (2019), “Adversarially robust generalization just requires more unlabeled data,” arXiv preprint arXiv:1906.00555.

Zhang, Y., Plevrakis, O., Du, S. S., Li, X., Song, Z., and Arora, S. (2020), “Over-parameterized Adversarial Training: An Analysis Overcoming the Curse of Dimensionality,” arXiv preprint arXiv:2002.06668.

The structure of appendix is as follows. In Section A, we present results w.r.t the effect of interpolation in high-dimensional case using $L_\infty$ attack. In Section B, we provide more numerical experiments (Section B.1 for low-dimensional case, Section B.2 for high-dimensional dense case, Section B.3 for high-dimensional sparse learning). Section C presents the proof of Theorem 1. Section D presents the proof of Theorems 2, 3, 5 and 6. Section E shows the proof for Lemma 1 and Theorem 4. Section F is for proof of $L_\infty$ attack. And finally Section G is for high-dimensional sparse model (Theorem 7).

A More theoretical results

A.1 $L_\infty$ Attack in high-dimensional Setup

As mentioned in Remark 5, for $L_\infty$ attack, the attack still has little effect when $d/n \to \infty$:

**Theorem 8.** Define the $L_\infty$ attack as $A^\infty_\epsilon(f_\theta, x, y) = \arg\max_{\|\tilde{x} - x\|_\infty \leq \epsilon} \{l(f_\theta(\tilde{x}), y) - x\}$, then for linear models, assuming data generation follows (4), when $d/n \to \infty$, with probability tending to 1,

$$\min_\theta R_{\mathcal{L}, \xi}(\theta, \epsilon) \rightarrow 0.$$ 

A simple simulation for Theorem 8 is shown in Figure 3. We use $\theta(y)$ to calculate the adversarial training loss under $L_2$ and $L_\infty$ attack with $\epsilon = 1$. The covariance $\Sigma = I$, coefficients of the standard model satisfies $\|\theta_0\| = 1$ and $\sigma$ is set to be $\sqrt{0.1}$. The training sample size is 50, and we take 100 repetitions for each $(d, \sigma^2)$ to take an average. The shade area in Figure 3 represents the region of log(mean ± one standard deviation). From Figure 3, the adversarial training loss decreases in $d$ for both two attacks.

![Figure 3: Effect of data dimension on adversarial training loss](image)

B More numerical results

B.1 Low-dimensional linear models

To verify Theorem 1 and the statement that “adversarial training hurts standard testing performance”, we run a linear model this experiment. The model is set to be $d = 10$ with $\theta_{0,i} = 1$ for $i = 1, \ldots, 10$. The covariance $\Sigma$ is $I$, and for noise, $\sigma^2 = 1$.

For adversarial training, we use zero initialization, $\xi = 0.1$, and $\eta = 0.01$. We repeat 100 times to get mean and standard deviation. The results are summarized in Figure 4. From Figure 4, one can
find that the adversarial testing loss is closed to $R^*(\epsilon)$, while the standard testing loss is away from 1 when $\epsilon = 0.5$.

![Figure 4: Adversarial training in linear regression under low data dimension. Dashed line in the middle panel: $R^*(\epsilon)$.

B.2 Low-dimensional two-Layer networks

Here we present a numerical experiment to verify our results in 2, and 3 on lazy training.

We take $d = 3$, $\theta_0 = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$, $\Sigma = I$, $\sigma^2 = 1$ for the data generation model, and $n = 100$. For the two-layer neural network, we take $h = 50$, and $a_j \sim \text{Unif}[-1, 1]$. For a network in Theorem 2, we take $\phi(x) = 1/(1 + e^{-x}) - 1/2$ and $\eta = 0.2$. To match the same $\phi'(0)$ and learning rate for all three models, we take $\phi(x) = x/4$ and $\eta = 0.2$ for linear network, and $\phi(x) = x/4 \{x > 0\}/4$ and $\eta = 0.8$ for ReLU network. For ReLU, we adjust negative $a_j$’s so that $\|a_j\| = \|a_j^+\|$.

For initialization, we take $\delta = 0.5$. For nonlinear networks, we use fast gradient method to approximate $A_{\epsilon, \xi}$ for both training and testing, and for surrogate loss, we take $\xi = 0.01$. We run the optimization for 4000 iterations, and repeat 50 times to get mean and standard deviation for the (population) adversarial risk. To estimate the adversarial risk, we randomly simulate 10000 samples and calculate the sample adversarial loss.

The results are shown in Figure 5. Since we match $\phi'(0)$ and learning rate for all the three networks, the adversarial risks decrease in the same speed and all converges to $R^*(\epsilon)$. However, due to the existence of $\xi$, linear network cannot reach an adversarial testing loss as $R^*(\epsilon)$. For finite $h$, sigmoid networks and ReLU networks have higher adversarial testing loss than linear networks.

![Figure 5: Adversarial training in three two-Layer neural networks (with lazy training) under low data dimension. Dashed line : $R^*(\epsilon)$.

B.3 High-dimensional dense models

B.3.1 Linear model

Besides the experiment in Figure 1, we further run some experiments with larger $d$ and $\xi = 0$. Figure
6 shows the experiment with the same setting as in Figure 1 but with $\xi = 0$. When $\theta = 0$, since the adversarial training loss is not differentiable, we do not impose attack. From Figure 6, the adversarial training / testing loss have similar performance as when $\xi = 0.1$, while the difference between the gradients of adversarial training and standard training becomes larger than the case when $\xi > 0$. To explain this, since $\|\theta^{(t)}\| \to 0$ when $d/n \to \infty$, the introduction of positive $\xi$ will leads to a surrogate attack with strength almost zero.

Figures 7 and 8 show the experiment with the same setting as in Figure 1 but with $\eta = 0.0002$ and $d = 5000$. In addition to the adversarial training / testing loss, from Figures 7 and 8, one can observe that the difference between the gradients of adversarial training and standard training gets smaller.

### B.3.2 Two-layer neural networks

Similar as in Section B.2, we conduct experiment on neural network with three different activation functions. The data generation follows those for Figure 1 with $\sigma^2 = 0.1$, $d = 1000$, $n = 20$. For neural networks, we take $h = 50$, and $a_j \sim \text{Unif}[-1,1]$. For ReLU, we adjust negative $a_j$’s so that $\|a^+\| = \|a^-\|$. Initialization takes $\delta$ such that $h^{\delta} = a^{0.6}$. The learning rate is taken as 0.16 for linear
and sigmoid networks, and 0.64 for ReLU network so that the convergence pattern is clear in the first 100 iterations. For adversarial surrogate loss, we take $\xi = 0.1$.

The results are summarized in Figure 9 for adversarial training loss and 10 for adversarial testing loss. For all three neural networks, the adversarial training loss decreases as fast as standard loss, while the adversarial testing loss are as higher than 1.

Figure 9: Adversarial training loss of adversarial training in three two-Layer neural networks (with lazy training) under high data dimension.

Figure 10: Adversarial testing loss of adversarial training in three two-Layer neural networks (with lazy training) under high data dimension.

### B.4 High-dimensional sparse models

In addition to the $L_2$ attack with $\xi = 0$ as in Figure 2, we also conduct experiments of $L_\infty$ attack with $\xi = 0$ and the two attacks with $\xi = 1e-04$ below. When $\xi > 0$, the adversarial testing accuracy is based on the original attack, i.e. not the surrogate one. As mentioned in Figure 6, under high-dimensional setup, a positive $\xi$ leads to the adversarial training getting closer to standard training when $n/d \to 0$. As a result, we choose a small enough $\xi$ such that the adversarial testing performance is closed to the case when $\xi = 0$. Similar as Figure 13, all the experiments in Figures 11, 12, and 13 shows a better performance using LASSO.

### C Proofs for low dimension linear model

In the proof of Theorem 1, we assume $v^2$ is a constant number, which implies that $\|\theta^*\| \leq B_0$ for some $B_0$. After investigating the results for bounded $v$, we then use a trick to extend to the case when $v$ is changing.

**Lemma 2.** Under the model in (4), when $\|\theta_0\| \leq B_0$ for some constant $B_0 > 0$, there exists some positive constants $c_1$ and $c_2$ such that for some function $g_1(\delta, \xi, d, n)$,

$$P \left( \sup_{\|\theta\| \leq B_0} \left\| \nabla \hat{R}_{L, \xi}(\theta, \epsilon) - \nabla R_{L, \xi}(\theta, \epsilon) \right\| > \delta \right) \leq e^{-g_1(\delta, \xi, d, n)},$$

where $\nabla \hat{R}_{L, \xi}(\theta, \epsilon)$ and $\nabla R_{L, \xi}(\theta, \epsilon)$ denote the gradient of $\theta$ after fixing $A_{\xi, \xi}$ (so it is not $\partial \hat{R}_{L, \xi}/\partial \theta$ or $\partial R_{L, \xi}/\partial \theta$ if $\xi > 0$). $c_d$ is some constant only depending on $d$. 

14
Figure 11: Comparison on standard (upper)/adversarial (lower) test accuracies between training with/without $L_1$ penalty under $L_2$ attack. Attack strength $\epsilon = 3$.

Figure 12: Comparison on standard (upper)/adversarial (lower) test accuracies between training with/without $L_1$ penalty under $L_\infty$ attack with $\xi = 1e^{-0.04}$. Attack strength $\epsilon = 0.3$.

**Proof of Lemma 2.** Assume $\theta \in B(0, B_0)$. For any sample $i$, the gradient of $\theta$ on surrogate adversarial loss is

$$2x_i (x_i^\top \theta - y_i) + 2\epsilon x_i \frac{\theta}{\sqrt{\|\theta\|^2 + \xi^2}} |x_i^\top \theta - y_i| + 2\epsilon x_i \frac{\|\theta\|^2}{\sqrt{\|\theta\|^2 + \xi^2}} \text{sgn}(x_i^\top \theta - y_i) + 2\epsilon^2 \theta \frac{\|\theta\|^2}{\|\theta\|^2 + \xi^2},$$

which has length $O(1)$ as long as $\theta \in B(0, B_0)$. Therefore, by Bernstein inequality, it holds that

$$P \left( \left\| \nabla R_{L,\xi}(\theta, \epsilon) - \nabla R_{L,\xi}(\theta, \epsilon) \right\| > \delta \right) \leq e^{-c_1 n \delta^2}.$$

For the ball $B(0, B_0)$, we use balls with radius $1/M$ to cover it. Then there are total $c_d B_0^d M^d$ balls for some constant $c_d$ which only depends on $d$. Denote $\theta_k$ as the center of the $k$th ball. The worst
Figure 13: Comparison on standard (upper)/adversarial (lower) test accuracy between training with/without $L_1$ penalty under $L_\infty$ attack.

case among the $c_d M^d$ centers of balls satisfies

$$P \left( \sup_k \left\| \nabla \hat{R}_{L,\xi}(\theta_k, \epsilon) - \nabla R_{L,\xi}(\theta_k, \epsilon) \right\| > \delta \right) \leq c_d B_0^d M^d e^{-c_1 n \delta^2}.$$ 

For any $\theta \in B(0, B_0)$, the distance from $\theta$ to its nearest $\theta_k$ is at most $1/M$, thus there exists some constant $c_2$ such that

$$\inf_k \left\| \nabla R_{L,\xi}(\theta_k, \epsilon) - \nabla R_{L,\xi}(\theta, \epsilon) \right\| \leq c_2 / M \xi.$$ 

In terms of $\nabla \hat{R}_{L,\xi}(\theta_k, \epsilon) - \nabla \hat{R}_{L,\xi}(\theta, \epsilon)$, we have

$$\nabla \hat{R}_{L,\xi}(\theta_k, \epsilon) - \nabla \hat{R}_{L,\xi}(\theta, \epsilon) = \frac{2}{n} \sum_{i=1}^n x_i x_i^\top (\theta_k - \theta) + \frac{2\epsilon}{n} \sum_{i=1}^n \left( \frac{\theta_k}{\sqrt{\|\theta_k\|^2 + \xi^2}} x_i^\top \theta_k - y_i \right) - \frac{\theta}{\sqrt{\|\theta\|^2 + \xi^2}} x_i^\top \theta - y_i,$$

$$+ \frac{2\epsilon}{n} \sum_{i=1}^n \left( \frac{\|\theta_k\|}{\sqrt{\|\theta_k\|^2 + \xi^2}} x_i \text{sgn}(x_i^\top \theta_k - y_i) - \frac{\|\theta\|}{\sqrt{\|\theta\|^2 + \xi^2}} x_i \text{sgn}(x_i^\top \theta - y_i) \right),$$

$$+ 2\epsilon^2 \theta_k \frac{\|\theta_k\|^2}{\|\theta_k\|^2 + \xi^2} - 2\epsilon^2 \theta \frac{\|\theta\|^2}{\|\theta\|^2 + \xi^2}.$$
thus taking \( \hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^\top \), and denoting \( \|A\| \) as the operator norm matrix \( A \),

\[
\left\| \nabla \tilde{R}_L(\theta_k, \epsilon) - \nabla \tilde{R}_L(\theta, \epsilon) \right\|
\leq 2\|\Sigma\| \|\theta_k - \theta\| + \left\| \frac{c}{n} \sum_{i=1}^{n} \left( \frac{\theta_k}{\sqrt{\|\theta_k\|^2 + \xi^2}} |x_i^\top \theta_k - y_i| - \frac{\theta}{\sqrt{\|\theta\|^2 + \xi^2}} |x_i^\top \theta - y_i| \right) \right\|
\leq 2\frac{c}{n} \|\theta_k - \theta\| \left( \frac{1}{n} \sum_{i=1}^{n} |x_i^\top \theta_k - y_i| \right)
\leq 2\frac{c}{n} \|\theta_k - \theta\| c_2'(\sup_i \|x_i\|).
\]

\[
A_2 = \left\| \frac{c}{n} \sum_{i=1}^{n} \left( \frac{\theta_k}{\sqrt{\|\theta_k\|^2 + \xi^2}} |x_i^\top \theta_k - y_i| - \frac{\theta}{\sqrt{\|\theta\|^2 + \xi^2}} |x_i^\top \theta - y_i| \right) \right\|
\leq 2\frac{c}{n} \|\theta_k - \theta\| c_2'(\sup_i \|x_i\|).
\]

\[
A_3 = \left\| \frac{\theta}{\sqrt{\|\theta\|^2 + \xi^2}} \frac{c}{n} \sum_{i=1}^{n} \left( |x_i^\top \theta_k - y_i| - |x_i^\top \theta - y_i| \right) \right\|
\leq 2\frac{c}{n} \|\theta_k - \theta\| c_2'(\sup_i \|x_i\|).
\]

\[
A_4 \leq 2\frac{c}{n} \|\theta_k - \theta\| c_2'(\sup_i \|x_i\|).
\]

\[
A_6 \leq c_2' \|\theta_k - \theta\|.
\]

For \( A_1 \), with probability tending to 1, we have

\[
A_1 \leq 2 \sqrt{\frac{d \log n}{n}} \|\theta_k - \theta\|.
\]

For \( A_5 \), the case is a little bit complicated.

\[
A_5 \leq 2\frac{c}{n} \sum_{i=1}^{n} \left( |x_i^\top \theta_k - y_i| - |x_i^\top \theta - y_i| \right)
\leq 2\frac{c}{n} \sum_{i=1}^{n} |x_i^\top \theta_k - y_i| \leq \|x_i\| \|\theta_k - \theta\|.
\]

For any \( \theta_k, x_i^\top \theta_k - y_i \), follows a Gaussian distribution with some mean and the variance is finite. As a result, for some \( c_3 > 0 \), we have

\[
P \left( \frac{1}{n} \sum_{i=1}^{n} |x_i| \mathbb{1} \{ |x_i^\top \theta_k - y_i| \leq \kappa \} - \mathbb{E} ( |x| \mathbb{1} \{ |x^\top \theta_k - y| \leq \kappa \} ) > \delta \right) \leq e^{-n c_3 \delta^2 / \kappa}.
\]
Note that $\mathbb{E}\left(|x|1\{|x^T\theta_k - y| \leq \kappa\}\right) = O(\kappa)$.

Furthermore, since $x_i$'s follows Gaussian, we have for some constant $c_4$,

$$P\left(\sup_i \|x_i\| \leq c_2\varepsilon\right) \leq n e^{-c_4\varepsilon^2/\sqrt{d}},$$

thus with probability at least $1 - (1/n)^{c_3\varepsilon^2-1} - c_4 B_0^d M^d e^{-c_3\varepsilon^2\sqrt{d} \log n}/M$ (take $\delta = \kappa = c_2\sqrt{d}\log n}/M$ in $A_3$), we have for some large $c_5$,

$$\inf_k \|\nabla \tilde{R}_{\xi,\varepsilon}(\theta_k, \varepsilon) - \nabla \tilde{R}_{\xi,\varepsilon}(\theta, \varepsilon)\| \leq c_5\left(\sqrt{d} \log n}/\xi M\right).$$

As a result, assume $\text{argmin}_j \|\theta - \theta_j\| = k$, then

$$\|\nabla \tilde{R}_{\xi,\varepsilon}(\theta, \varepsilon) - \nabla R_{\xi,\varepsilon}(\theta, \varepsilon)\| \leq \|\nabla \tilde{R}_{\xi,\varepsilon}(\theta, \varepsilon) - \nabla \tilde{R}_{\xi,\varepsilon}(\theta_k, \varepsilon)\| + \|\nabla \tilde{R}_{\xi,\varepsilon}(\theta_k, \varepsilon) - \nabla \tilde{R}_{\xi,\varepsilon}(\theta, \varepsilon)\|,$$

thus we have

$$P\left(\sup_{\|\theta\| \leq B_0} \|\nabla \tilde{R}_{\xi,\varepsilon}(\theta, \varepsilon) - \nabla R_{\xi,\varepsilon}(\theta, \varepsilon)\| > c_6 \sqrt{\log n}/\xi M\right)$$

$$\leq 1 - (1/n)^{c_3\varepsilon^2-1} - c_4 B_0^d M^d e^{-c_3\varepsilon^2\sqrt{d} \log n}/M - c_4 B_0^d M^d e^{-c_3\varepsilon^2\sqrt{d} \log n}/M + 1/n^{C_2-1}.$$ 

Rewrite $\sqrt{d} \log n}/M$ as $c_5 \delta$ (or $M = \sqrt{d} \log n}/(c_5 \delta)$) and constant $c_3$'s, since both $\xi$ and $\delta$ converges to zero, and $M \to \infty$, we have

$$P\left(\sup_{\|\theta\| \leq B_0} \|\nabla \tilde{R}_{\xi,\varepsilon}(\theta, \varepsilon) - \nabla R_{\xi,\varepsilon}(\theta, \varepsilon)\| > \delta\right)$$

$$\leq c_4 B_0^d e^{-c_3 n(d - d \log (\delta\xi) + d \log(d \log n))/2} + c_4 B_0^d e^{-c_3 n(d - d \log (\delta\xi) + d \log(d \log n))/2} + 1/n^{C_2-1}.$$ 

$$:= e^{-g_1(\delta, \xi, d, n)}.$$

\[\square\]

**Proof of Theorem 1.** Rewrite $\theta^{(t)}$ as $\theta^{(t)}$ for simplicity. Denote $\tilde{R}_{\xi,\varepsilon}(\theta, \varepsilon)$ as $\tilde{R}_{\xi,\varepsilon}(\theta, \varepsilon) = \|\theta - \theta_0\|_2^2 + \sigma^2 + 2c_0\varepsilon \sqrt{\|\theta\|^2 + \xi^2} \sqrt{\|\theta - \theta_0\|_2^2 + \sigma^2 + \varepsilon^2} \|\theta\|^2 + \xi^2$,

then taking gradient on $\tilde{R}_{\xi,\varepsilon}$ w.r.t $\theta$, it becomes

$$\frac{\partial \tilde{R}_{\xi,\varepsilon}}{\partial \theta} = 2\Sigma(\theta - \theta_0) + 2c_0\varepsilon \theta \sqrt{\|\theta\|^2 + \xi^2} \sqrt{\|\theta - \theta_0\|_2^2 + \sigma^2} + 2c_0\varepsilon \sqrt{\|\theta\|^2 + \xi^2} \sum(\theta - \theta_0) \sqrt{\|\theta - \theta_0\|_2^2 + \sigma^2} + 2\varepsilon^2 \theta$$

$$= 2\Sigma(\theta - \theta_0) \left(1 + c_0\varepsilon \frac{\sqrt{\|\theta\|^2 + \xi^2} \sqrt{\|\theta - \theta_0\|_2^2 + \sigma^2}}{\sqrt{\|\theta - \theta_0\|_2^2 + \sigma^2}}\right) + 2\theta \left(c_0\varepsilon \frac{\sqrt{\|\theta - \theta_0\|_2^2 + \sigma^2}}{\sqrt{\|\theta\|^2 + \xi^2}} + \varepsilon^2 \sqrt{\|\theta\|^2 + \xi^2}\right).$$

For $\nabla R_{\xi,\varepsilon}(\theta, \varepsilon)$, we have

$$\nabla R_{\xi,\varepsilon}(\theta, \varepsilon) = 2\Sigma(\theta - \theta_0) \left(1 + c_0\varepsilon \frac{\|\theta\|^2}{\sqrt{\|\theta\|^2 + \xi^2} \sqrt{\|\theta - \theta_0\|_2^2 + \sigma^2}}\right) + 2\theta \left(c_0\varepsilon \frac{\sqrt{\|\theta - \theta_0\|_2^2 + \sigma^2}}{\sqrt{\|\theta\|^2 + \xi^2}} + \varepsilon^2 \sqrt{\|\theta\|^2 + \xi^2}\right).$$

We assume $\theta \in B(0, B_1)$, then there exists $B_2$ such that both $\|\nabla R_{\xi,\varepsilon}(\theta, \varepsilon)\|$ and $\|\partial \tilde{R}_{\xi,\varepsilon}(\theta, \varepsilon)/\partial \theta\|$ are bounded by some large constant $B_2 > 0$. 

18
To compare the difference between \( \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta} \) and \( \nabla R_{L, \xi}(\theta, \epsilon) \), we have

\[
\begin{align*}
&\left\| \nabla R_{L, \xi}(\theta, \epsilon) - \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta} \right\| \\
= &\left\| 2\epsilon_0 \left( \sqrt{||\theta||^2 + \xi^2} - \frac{||\theta||^2}{\sqrt{||\theta||^2 + \xi^2}} \right) \frac{\Sigma(\theta - \theta_0)}{\sqrt{||\theta - \theta_0||_2^2 + \sigma^2}} + 2\epsilon^2 \theta - \frac{\xi^2}{||\theta||^2 + \xi^2} \right\| \\
= &\left\| 2\epsilon_0 \epsilon \frac{\xi^2}{\sqrt{||\theta||^2 + \xi^2} \sqrt{||\theta - \theta_0||_2^2 + \sigma^2}} + 2\epsilon^2 \theta - \frac{\xi^2}{||\theta||^2 + \xi^2} \right\|.
\end{align*}
\]

As a result, there exists some \( B_3 > 0 \) such that

\[
\left\| \nabla R_{\xi}(\theta, \epsilon) - \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta} \right\| \leq \frac{\xi}{B_3} \left\| \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta} \right\|.
\]

Therefore the gradient of \( \tilde{R}_{L, \xi} \) is the dominant term when updating \( \theta \) in each iteration.

Next we show that \( \tilde{R}_{L, \xi}(\theta^{(t+1)}, \epsilon) \) is smaller than \( \tilde{R}_{L, \xi}(\theta^{(t)}, \epsilon) \) in probability. From the definition of \( \tilde{R} \), similar with Proposition 1, there exists \( L \) such that

\[
\tilde{R}_{L, \xi}(\theta^{(t+1)}, \epsilon) \leq \tilde{R}_{L, \xi}(\theta^{(t)}, \epsilon) + \left( \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta(t)} \right)^T (\theta^{(t+1)} - \theta^{(t)}) + \frac{L}{\xi} \| \theta^{(t+1)} - \theta^{(t)} \|^2.
\]

From Lemma 2, with probability at least \( 1 - e^{-\delta, \xi, d, \alpha} \),

\[
\left\| \nabla \tilde{R}_{L, \xi}(\theta, \epsilon) - \nabla R_{L, \xi}(\theta, \epsilon) \right\| < \delta,
\]

thus

\[
\left( \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta(t)} \right)^T (\theta^{(t+1)} - \theta^{(t)}) = -\eta \left( \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta(t)} \right)^T \nabla \tilde{R}_{L, \xi}(\theta^{(t)}, \epsilon) \leq -\eta \left\| \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta(t)} \right\|^2 (1 - \xi / B_3) + \eta \delta B_2,
\]

and

\[
\| \theta^{(t+1)} - \theta^{(t)} \|^2 = \eta^2 \| \nabla \tilde{R}_{L, \xi}(\theta^{(t)}, \epsilon) \| \leq \eta^2 \| \nabla R_{L, \xi}(\theta^{(t)}, \epsilon) \| + 2\eta^2 \delta B_2 + \eta^2 B_2^2 \delta^2
\]

\[
\leq \eta^2 \left\| \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta(t)} \right\|^2 (1 + \xi / B_3)^2 + 2\eta^2 \delta B_2 + o,
\]

where \( A + o \) represents \( A + \alpha \) where \( \alpha / A \rightarrow 0 \) for expressions \( a \) and \( A \).

Therefore,

\[
\tilde{R}_{L, \xi}(\theta^{(t+1)}, \epsilon) \leq \tilde{R}_{L, \xi}(\theta^{(t)}, \epsilon) - \eta \left\| \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta(t)} \right\|^2 (1 - \xi / B_3) + \eta \delta B_2 + \frac{L \eta^2}{\xi} \left\| \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta(t)} \right\|^2 (1 + 2\xi / B_3) + o.
\]

When taking \( \eta = \frac{\xi}{2L} \),

\[
\tilde{R}_{L, \xi}(\theta^{(t+1)}, \epsilon) \leq \tilde{R}_{L, \xi}(\theta^{(t)}, \epsilon) - \frac{\xi}{4L} \left\| \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta(t)} \right\|^2 + \frac{\xi^2}{2L} B_2 + \frac{\xi \delta B_2}{2L} + o.
\]

From (10), when taking \( \xi \rightarrow 0 \) and \( \delta \rightarrow 0 \), we know that \( \tilde{R}_{\xi}(\theta^{(t)}, \epsilon) \) will decrease in \( t \) at the beginning of training; thus based on the shape of \( \tilde{R}_{\xi} \) and \( \| \theta^{(0)} \| \leq B_0 \), one can use induction to show the existence of \( B_1 \), i.e. \( \theta(t) \in B(0, B_1) \) for any \( t \).

To bound the difference \( \tilde{R}_{L, \xi}(\theta^{(t+1)}, \epsilon) \) and \( \tilde{R}_{\xi}(\theta^*, \epsilon) \), since \( \tilde{R}_{\xi} \) is a convex function, we have

\[
\tilde{R}_{L, \xi}(\theta^{(t)}, \epsilon) \leq \tilde{R}_{L, \xi}(\theta^*, \epsilon) + \left( \frac{\partial \tilde{R}_{L, \xi}}{\partial \theta(t)} \right)^T (\theta^{(t)} - \theta^*).
\]
We first bound the difference among (1) and (2) assuming \( \xi \). We consider three optimization problems: (1) a linear network using zero initialization, (2) a nonlinear network using zero initialization, and (3) a nonlinear network with vanishing initialization. Extending from Theorem 1, we know how (1) works, then we bound the difference between (1), (2), and (3).

As a result, summing up from \( t = 1 \) to \( T \),

\[
\sum_{t=1}^{T} \tilde{R}_{L,\xi}(\theta(t), \epsilon) - \tilde{R}_{L,\xi}(\theta^*, \epsilon) \leq \frac{L}{\xi} \|\theta(0) - \theta^*\|^2 + \frac{T\|\theta(0) - \theta^*\|^2}{\xi \epsilon B_1 B_2 + o}. \]

When \( \|\theta(t) - \theta^*\|^2 / \xi \to \infty \) (take \( \xi / \delta \to \infty \)), the algorithm reduces the loss in every iteration in probability, thus with probability at least \( 1 - e^{-\alpha(\delta, \xi, d, n)} \),

\[
\tilde{R}_{L,\xi}(\theta(t), \epsilon) - \tilde{R}_{L,\xi}(\theta^*, \epsilon) \leq \frac{L}{\xi} \|\theta(0) - \theta^*\|^2 + \frac{\xi B_1 B_2}{B_3} + o. \]

If there exists some \( \theta(t) \) such that \( \|\nabla \tilde{R}_{L,\xi}(\theta(t), \epsilon)\|^2 / (4\xi / B_3) < 1 \), then \( \theta^{(T)} \) will also be closed enough to \( \theta^* \). Thus choose \( t \) such that \( \xi t \to 0 \).

When \( \xi \to 0 \), we have \( \tilde{R}_{L,\xi}(\theta, \epsilon) - R_{L,\xi}(\theta, \epsilon) \to 0 \) and \( R_{L,\xi}(\theta, \epsilon) - R_0(\theta, \epsilon) \to 0 \), thus through taking suitable choice of \( (\xi, t, \eta) \), we can get \( R_{L,\xi}(\theta^{(T)}, \epsilon) \to R_0(\theta^*, \epsilon) \).

The proof of \( \tilde{R}_{L,\xi}(\theta^{(T)}, \epsilon) - \tilde{R}_{L,\xi}(\theta^{(T)} \epsilon) \to 0 \) in probability follows a concentration bound similar as in Lemma 2.

Finally we relax the condition of \( \|\theta_0\| \leq B_0 \). For \( y_i = \theta_0^\top x_i + \varepsilon_i \) with \( \|\theta_0\|^2 + \sigma^2 = 1, \sigma > \xi \), denote \( y'_i := v y_i = (v \theta_0)^\top x_i + \varepsilon_i := (\theta'_0)^\top x_i + \varepsilon'_i \) for some \( v \to \infty \). In this case, if the initialization satisfy \( (\theta^{(0)}) = v \theta^{(0)} \) and \( \xi' = v \xi \), then for any \( t = 1, ..., T \), we always have \( (\theta^{(t)}) = v \theta^{(t)} \).

Therefore, since \( \theta^{(T)} \to \theta^* \), we also have \( \theta^{(T)} / v \to \theta^* \), which is just the minimizer of population adversarial loss for \( y_i' \).

\[ \square \]

## D Proofs for low-dimensional nonlinear network

**Proof of Theorem 2.** In the proof, we directly consider taking gradient on loss w.r.t \( x \) to get the attack direction. When \( \sqrt{n} \to \infty \), the difference between the gradient and the true attack vanishes.

We consider three optimization problems: (1) a linear network using zero initialization, (2) a nonlinear network using zero initialization, and (3) a nonlinear network with vanishing initialization. Extending from Theorem 1, we know how (1) works, then we bound the difference between (1), (2), and (3).

We first bound the difference among (1) and (2) assuming \( \xi \) and \( t \) are any arbitrary number. And finally choose some suitable \( \xi \) and \( t \) to ensure the consistency.

Denote \( A_{\epsilon, \xi, \OP}(f, x, y) = \epsilon \sgn(f(x) - y) \frac{\sum_{j=1}^{h} a_j \phi'(0) \theta_j}{\sqrt{\left|\sum_{j=1}^{h} a_j \phi'(0) \theta_j\right|^2 + \xi^2}}, \)

and for the weight \( \theta_j \) for the \( j \)th node,

\[
g_{j, \epsilon, \OP}(\theta, \epsilon) = \frac{2}{n} \sum_{i=1}^{n} \frac{\phi(0)' a_j}{\sqrt{h}} \left[ x_i + A_{\epsilon, \xi, \OP}(f, x_i, y_i) \left[ \frac{1}{\sqrt{R}} \sum_{k=1}^{h} \phi(0)' x_i \theta_k a_k + \phi'(0) A_{\epsilon, \xi, \OP}(f, x_i, y_i)^\top \theta_k a_k \right] - y_i \right].
\]

\[
g_{j, \epsilon, \OP}(\theta, \epsilon) = \frac{2}{n} \sum_{i=1}^{n} \frac{\phi'(0) (x + A_{\epsilon, \xi}(f, x, y))^\top \theta_j}{\sqrt{h}} a_j \left[ x_i + A_{\epsilon, \xi}(f, x_i, y_i) \left[ \frac{1}{\sqrt{R}} \sum_{j=1}^{h} \phi((x + A_{\epsilon, \xi}(f, x, y))^\top \theta_j) a_j - y_i \right] \right].
\]
Denote $\theta^{OP1}(t)$ as the weight obtained using linear network with zero initialization, i.e. for each hidden node $j = 1, \ldots, h$,

$$
\theta_j^{OP1}(0) = 0,
\theta_j^{OP1}(t + 1) = \theta_j^{OP1}(t) - \eta g_j^{OP1}(\theta_j^{OP1}(t), \epsilon).
$$

Also denote $\theta^{OP2}(t)$ as the weight obtained using nonlinear network with nonzero initialization, i.e.

$$
\theta_j^{OP2}(0) = 0,
\theta_j^{OP2}(t + 1) = \theta_j^{OP2}(t) - \eta g_j^{OP2}(\theta_j^{OP2}(t), \epsilon).
$$

For the original problem we consider (i.e. nonlinear network with vanishing initialization), define

$$
\theta_j^{OP3}(0) \sim N \left( \frac{1}{\sqrt{d+h+\delta^f}} \mathbf{I} \right), \quad \theta_j^{OP3}(t + 1) = \theta_j^{OP2}(t) - \eta g_j^{OP2}(\theta_j^{OP2}(t), \epsilon).
$$

**Difference between (1) and (2)** At $(t + 1)$th step, we have

$$
\theta_j^{OP1}(t + 1) - \theta_j^{OP2}(t + 1) = \theta_j^{OP1}(t) - \theta_j^{OP2}(t) - \eta g_j^{OP1}(\theta_j^{OP1}(t), \epsilon) + \eta g_j^{OP2}(\theta_j^{OP2}(t), \epsilon)
$$

$$
= \theta_j^{OP1}(t) - \theta_j^{OP2}(t) - \eta \left[ g_j^{OP1}(\theta_j^{OP1}(t), \epsilon) - g_j^{OP1}(\theta_j^{OP2}(t), \epsilon) \right] - \eta \left[ g_j^{OP2}(\theta_j^{OP2}(t), \epsilon) - g_j^{OP2}(\theta_j^{OP2}(t), \epsilon) \right].
$$

Denote

$$
\Delta_1(\theta, x, y) = A_{\epsilon, \xi}(f, x, y) - A_{\epsilon, \xi, OP1}(f, x, y),
$$

$$
\Delta_2(\theta, x, y) = \frac{1}{\sqrt{h}} \sum_{j=1}^{h} \phi \left( (x + A_{\epsilon, \xi}(f, x, y))^\top \theta_j \right) a_j
$$

$$
- \frac{1}{\sqrt{h}} \sum_{j=1}^{h} \phi(0)^\top \theta_j a_j + \phi(0) A_{\epsilon, \xi, OP1}(f, x, y)^\top \theta_j a_j,
$$

and

$$
\Delta_3,(\theta, x, y) = \phi^\prime \left( (x + A_{\epsilon, \xi}(f, x, y))^\top \theta_j \right) - \phi^\prime(0).
$$

Assume all $\Delta_{1,j}(\theta, x, y)$, $\Delta_{2,j}(\theta, x, y)$, $\Delta_{3,j}(\theta, x, y)$ converges to zero for any $j$ and $(x, y) \in \{(x_i, y_i)\}_{i=1,\ldots,n}$, then

$$
g_j^{OP1}(\theta, \epsilon) - g_j^{OP2}(\theta, \epsilon)
$$

$$
= -\frac{2}{n} \sum_{i=1}^{n} \phi^\prime \left( (x + A_{\epsilon, \xi}(f, x, y))^\top \theta_j \right) a_j [x_i + A_{\epsilon, \xi}(f, x, y)] \left[ \frac{1}{\sqrt{h}} \sum_{k=1}^{h} \phi \left( (x_i + A_{\epsilon, \xi}(f, x, y)) \theta_k \right) a_k - y_i \right]
$$

$$
+ \frac{2}{n} \sum_{i=1}^{n} \phi(0)^\top a_j \left[ x_i + A_{\epsilon, \xi, OP1}(f, x, y) \right] \left[ \frac{1}{\sqrt{h}} \sum_{k=1}^{h} (\phi(0)^\top \theta_k a_k + \phi^\prime(0) A_{\epsilon, \xi, OP1}(f, x, y)^\top \theta_k a_k) - y_i \right]
$$

$$
= -\frac{2}{n} \sum_{i=1}^{n} \frac{a_j \Delta_{3,j}(\theta, x, y_i)}{\sqrt{h}} [x_i + A_{\epsilon, \xi, OP1}(f, x, y_i)] \left[ \frac{1}{\sqrt{h}} \sum_{k=1}^{h} (\phi(0)^\top \theta_k a_k + \phi^\prime(0) A_{\epsilon, \xi, OP1}(f, x, y_i)^\top \theta_k a_k) - y_i \right]
$$

$$
- \frac{2}{n} \sum_{i=1}^{n} \frac{\phi(0)^\top a_j \Delta_1(\theta, x, y_i)}{\sqrt{h}} \left[ \frac{1}{\sqrt{h}} \sum_{k=1}^{h} (\phi(0)^\top \theta_k a_k + \phi^\prime(0) A_{\epsilon, \xi, OP1}(f, x, y_i)^\top \theta_k a_k) - y_i \right]
$$

$$
- \frac{2}{n} \sum_{i=1}^{n} \frac{\phi(0)^\top a_j}{\sqrt{h}} [x_i + A_{\epsilon, \xi, OP1}(f, x, y_i)] \Delta_2(\theta, x, y_i) + o,
$$

21
for some remainder term \( o \), thus

\[
\frac{1}{\sqrt{h}} \sum_{i=1}^{h} a_j g_{j,i}^{OP1}(\theta, \epsilon) - \frac{1}{\sqrt{h}} \sum_{i=1}^{h} a_j g_{j,i}^{OP2}(\theta, \epsilon) = - \frac{2}{n} \sum_{j=1}^{n} \frac{\phi(0)' x_{ij} \theta a_k + \phi'(0) A_{t,j,OP1}(f, x_i, y_i)^\top \theta a_k}{h} \Delta_1(\theta, x_i, y_i)
\]

\[
= - \frac{2}{n} \sum_{j=1}^{n} \phi(0)' \parallel a \parallel^2 \Delta_1(\theta, x_i, y_i) \left[ \frac{1}{\sqrt{h}} \sum_{k=1}^{h} (\phi(0)' x_{i} \theta_k a_k + \phi'(0) A_{t,j,OP1}(f, x_i, y_i)^\top \theta_k a_k) - y_i \right]
\]

\[
= - \frac{2}{n} \sum_{j=1}^{n} \phi(0)' \parallel a \parallel^2 \Delta_1(\theta, x_i, y_i) \left[ \frac{1}{\sqrt{h}} \sum_{k=1}^{h} (\phi(0)' x_{i} \theta_k a_k + \phi'(0) A_{t,j,OP1}(f, x_i, y_i)^\top \theta_k a_k) - y_i \right]
\]

When \( \parallel \theta^\top a / \sqrt{h} \parallel \leq B_1 \), we have

\[
\left\| \frac{1}{\sqrt{h}} \sum_{j=1}^{h} a_j g_{j,i}^{OP1}(\theta, \epsilon) - \frac{1}{\sqrt{h}} \sum_{j=1}^{h} a_j g_{j,i}^{OP2}(\theta, \epsilon) \right\| = O\left(v \max_i \left( \sum_{j=1}^{h} a_j^2 \Delta_{s,j}(\theta, x_i, y_i) \right) \frac{\parallel a \parallel^2}{h} + v \max_i \parallel \Delta_i(\theta, x_i, y_i) \parallel + \max_i |\Delta_2(\theta, x_i, y_i)|\right).
\]

We know that with probability at least \( 1 - 2e^{-g(\delta, \xi, d, n)} \),

\[
\left\| \frac{1}{\sqrt{h}} \sum_{j=1}^{h} a_j g_{j,i}^{OP1}(\theta, \epsilon) - \frac{1}{\sqrt{h}} \sum_{j=1}^{h} a_j g_{j,i}^{OP2}(\theta', \epsilon) \right\| \leq 2\delta + \frac{\parallel a \parallel^2}{h} \| \nabla R_{t,j,\xi}(\theta^\top a / \sqrt{h}, \gamma) - \nabla R_{t,j,\xi}((\theta')^\top a / \sqrt{h}, \gamma) \|
\]

\[
\leq 2\delta + \frac{\parallel a \parallel^2}{h} \frac{L}{\sqrt{h}} \parallel \theta - (\theta') \parallel a. 
\]

As a result, with probability at least \( 1 - 2e^{-g(\delta, \xi, d, n)} \),

\[
\left\| \frac{1}{\sqrt{h}} \theta^{OP1}(t + 1)^\top a - \frac{1}{\sqrt{h}} \theta^{OP2}(t + 1)^\top a \right\| \leq \left\| \frac{1}{\sqrt{h}} \theta^{OP1}(t)^\top a - \frac{1}{\sqrt{h}} \theta^{OP2}(t)^\top a \right\| \left( 1 + \frac{\parallel a \parallel^2}{h} \frac{L_y}{\xi} \right) + 2\eta\delta + O\left( \eta v \max_i \left( \sum_{j=1}^{h} a_j^2 \Delta_{s,j}(\theta^{OP2}(t), x_i, y_i) \right) \frac{\parallel a \parallel^2}{h} \right)
\]

\[
+ \eta v \max_i \parallel \Delta_1(\theta^{OP2}(t), x_i, y_i) \parallel + \eta \max_i |\Delta_2(\theta^{OP2}(t), x_i, y_i)|. 
\]

**Linear network in (1)** To further figure out the difference, we need some knowledge on \( \theta^{OP1}(t) \), i.e. how \( \theta^{OP1}(t) \) affects \( \Delta_1, \Delta_2, \) and \( \Delta_3 \).

Observe that \( \frac{\phi(0)' \theta^{OP1}(t)^\top a}{h} \) is just the coefficients of a linear model, so using zero initialization on \( \theta^{OP1}(t) \), we have for any \( j, k \) in \( 1, ..., h \),

\[
\frac{\parallel \theta^{OP1}(t) \parallel}{\parallel \theta^{OP1}(t) \parallel} = \frac{|a_j|}{|a_k|}.
\]

Recall that \( \theta_0 \) is the true model (i.e. a vector, not a matrix). With probability tending to 1,

\[
\max_i \left| \sum_{j=1}^{h} a_j^2 \Delta_{s,j}(\theta^{OP1}(t), x_i, y_i) \right| = O \left( \sqrt{d \log n} \max_i \parallel \theta^{OP1}(t) \parallel \right) = O \left( \sqrt{d \log n} \frac{\parallel a \parallel_{\infty}}{\parallel a \parallel^2} \sqrt{h} \parallel \theta_0 \parallel \right).
\]
Similarly, for $\Delta_1$,
\[
\|A_{\epsilon, \xi}(f, x) - A_{\epsilon, \xi, \delta, \xi}(f, x, y)\| = O \left( \frac{\|x\| \sum_{j=1}^{h} |a_j| \|\theta_j\|^2}{\|\sum_{j=1}^{h} a_j \theta_j\|} \right).
\]

With probability tending to 1,
\[
\max_{i, t} \|\Delta_1(\theta^{OP_1}(t), x_i, y_i)\| = O \left( \sqrt{d \log n \|a\|_\infty \|\theta_0\| / \|a\|^2} \right).
\]

Based on $\Delta_1$, for $\Delta_2$, when $\sqrt{d \log n \|a\|_\infty \|\theta_0\| / \|a\|^2} \to 0$, with probability tending to 1,
\[
\max_{i, t} |\Delta_2(\theta^{OP_1}(t), x_i, y_i)|
\]
\[
= O \left( \frac{1}{\sqrt{h}} \sum_{j=1}^{h} a_j (x + A_{\epsilon, \xi}(f^{OP_1}(f, x, y)^\top \theta^{OP_1}(T))^2 + \frac{\phi'(0) \|\Delta_1(\theta, x, y)\|}{\sqrt{h}} \sum_{j=1}^{h} a_j \|\theta_j^{OP_1}(T)\|) \right)
\]
\[
+ O \left( (d \log n) \|a\|_\infty \|\theta_0\|^2 / \|a\|^2 + \|\Delta_1(\theta^{OP_1}(T), x, y)\| \|\theta_0\| \right).
\]

Therefore we require $(d \log n) \sqrt{h} \|a\|_\infty \|\theta_0\| / \|a\|^2 \to 0$.

**Return to the difference between (1) and (2)**  Now we use the property that $\theta^{OP_2}(0) = 0$. Similar as $\theta^{OP_1}$, if $a_j, a_k > 0$, then
\[
\frac{\|\theta_j^{OP_2}(t)\|}{\|\theta_k^{OP_2}(t)\|} = \frac{|a_j|}{|a_k|}.
\]

Therefore, we define $\theta_+^{OP_2}(t) = \sum_{a_j > 0} a_j \theta_j^{OP_2}(t)$, and similarly define $\theta_-^{OP_2}(t), \theta_0^{OP_1}(t), \theta^{OP_1}(t)$.

One can obtain similar result for (11) to (13) when considering $\theta_+^{OP_1}(t) - \theta_-^{OP_2}(t)$ and $\theta^{OP_1}(t) - \theta_-^{OP_2}(t)$. When $\eta(\delta + (d \log n) \sqrt{h} \|a\|_\infty / \|a\|^2)(1 + \|a\|^2 L \eta / (\sqrt{h} \xi)) \to 0$, with probability $1 - 2T e^{-\delta (\delta, \xi, d, n)}$, we have $\|\theta_+^{OP_1}(T) - \theta_+^{OP_2}(T)\| / \|\theta_0\|$ and $\|\theta_-^{OP_1}(T) - \theta_-^{OP_2}(T)\| / \|\theta_0\|$ converges to zero.

**Difference between (2) and (3)**
\[
\|\theta^{OP_3}(t + 1) - \theta^{OP_2}(t + 1)\|_F
\]
\[
\leq \|\theta^{OP_3}(t) - \theta^{OP_2}(t)\|_F + \eta \left( \sum_{j=1}^{h} \left| a_j \theta_j^{OP_3}(t, \epsilon) - a_j \theta_j^{OP_2}(t, \epsilon) \right|^2 \right)^{1 / 2}
\]
\[
= \|\theta^{OP_3}(t) - \theta^{OP_2}(t)\|_F O \left( 1 + \frac{L \eta \|\theta_0\| \|a\|_\infty}{\sqrt{h} \xi} \right).
\]

Therefore,
\[
\|\theta^{OP_3}(T) - \theta^{OP_2}(T)\|_F = O \left( \|\theta^{OP_3}(0)\|_F \left( 1 + \frac{L \eta \|\theta_0\| \|a\|_\infty}{\sqrt{h} \xi} \right)^T \right) = O_p \left( \frac{1}{h^{3/2}} \left( 1 + \frac{L \eta \|\theta_0\| \|a\|_\infty}{\sqrt{h} \xi} \right)^T \right).
\]

As a result, we require $\|\theta^{OP_3}(T) - \theta^{OP_2}(T)\|_F \sqrt{d \log n} \to 0$ so that with probability tending to 1, for all nodes, $\|x_i^{OP_3}(T)\| \to 0$. In this case,
\[
\|\theta^{OP_3}(T) a - \theta^{OP_2}(T) a\| \leq \|a\|_\infty \|\theta^{OP_3}(T) - \theta^{OP_2}(T)\|_F.
\]
Deciding proper choice of $\xi$. In Theorem 1, since $\frac{\phi'(0)}{\eta}$ is a linear model, denote $\eta_{\text{linear}}$ as the learning rate for linear model, then the corresponding $\eta$ in linear network is $\eta = \eta_{\text{linear}} h/\|a\|^2 \phi'(0)$. We require $T\eta_{\text{linear}} \to \infty$, $\xi/v \to 0$, $\eta_{\text{linear}} = \xi/(2Lv)$, and $\delta/\xi \to 0$ in Theorem 1.

Now we list all the assumptions we made on $(\xi, \eta, T, \delta)$ during derivation.

- Difference between $\theta^{OP1}(t)$ and $\theta^{OP2}(t)$: $T e^{-\eta (\delta, \xi, d, n)} \to 0$, $\|a\|^2 L \eta / (h \xi)^T / v \to 0$.

- Difference between $\theta^{OP2}(t)$ and $\theta^{OP3}(t)$: $\sqrt{\frac{d \log n}{n}} \Theta(n \log n, \|a\|^2 / \|a\|^2) \to 0$.

So we take $\delta = \sqrt{(d \log n)/n}$, and

$$\xi / v = - \log \log n / \log \left( \sqrt{\frac{d \log n}{n}} \sqrt{(d \log n) \sqrt{\|a\|^2 / \epsilon_\tilde{\delta}}} \right) \sqrt{\frac{d \log n}{h \delta}}.

\qed$

Proof of Theorem 5. The proof is similar as the one for Theorem 3 through replacing $\eta_1$ with $\theta(y)$. Based on Lemma 1, we know that $\|\theta(y)\| / v = O_p(n/d)$, thus in the step “Linear Network in (1)”, to ensure $\|f_\theta(y)(x_i) - y\|$ decreases in each iteration, we require $\sqrt{d \log n (n \log n) \sqrt{\|a\|^2 / \epsilon_\tilde{\delta}}} \to 0$.

On the other hand, for the step “Difference between (2) and (3)”, we take $a$ and $\delta'$ such that $\sqrt{d \log n (\theta^{OP3}(t) a - \theta^{OP2}(t) a)} = o(1)$.

Proof of Theorem 3 and 6. Assume $X = [X_0, -X_0]^T$. Since $\|a^+\| = \|a^-\|$, one can check that all $\Delta_1, \Delta_2$, and $\Delta_3$ are zeros. Take $\phi'(0) = 1$. Denote $S^+(\theta)$ as $\{j \mid f(x + A_{1, \xi}(f, xi, y_i) > 0\}$, and similarly denote $S^-(\theta)$. Then one can observe that, using zero initialization,

$$\frac{1}{\sqrt{h}} \sum_{j \in S^+(\theta)} a_j g_{j, \xi}(\theta, \epsilon) + \frac{1}{\sqrt{h}} \sum_{j \in S^-(\theta)} a_j g_{j, \xi}(\theta, \epsilon) = 0,$

and hence ReLU-activated neural networks with zero initialization perform the same as linear networks when the learning rate for ReLU-activated neural is twice as the one for linear networks.

For $\|\theta^{OP3}(t) a - \theta^{OP2}(t) a\|$, for low-dimensional regime, we require it is in $o(1)$, and for high-dimensional regime, we require it is in $O(\|\theta(y)\|)$, thus $\delta$ in Theorem 3 and 6 are different.

\qed

E  Proofs for high-dimensional dense model

Proof of Lemma 1. Denote $\theta^+ = \theta(y)$. When $v$ is a constant, $\|\theta^+\|$ satisfies

$$\|\theta^+\|^2 = y^T (XX^T)^{-1} y = \theta_0^T X^T (XX^T)^{-1} X_0 + \epsilon^T (XX^T)^{-1} \epsilon \to \theta_0^T X^T (XX^T)^{-1} X_0 + \sigma^2 tr((XX^T)^{-1}).$$

Furthermore, similar as Belkin et al. (2019), we have

$$\left(\theta_0^T \Sigma^{1/2} \Sigma^{-1/2} X^T (XX^{-1} X^T)^{-1} X \Sigma^{-1/2} (\Sigma^{1/2} \theta_0) - \frac{n}{d} \|\theta_0 \Sigma^{1/2}\|^2. \right.$$

Therefore, since $XX^{-1} X^T - \frac{1}{\sigma_{\max}} XX^T$ is positive definite for the smallest eigenvalue $\lambda_{\min}$ of $\Sigma$ and, we have

$$\theta_0^T (XX^T)^{-1} X_0 \leq \frac{1}{2 \lambda_{\max}} (\theta_0^T \Sigma^{1/2}) \Sigma^{-1/2} (XX^T)^{-1} X \Sigma^{-1/2} (\Sigma^{1/2} \theta_0) \to \frac{n}{d} \|\theta_0 \Sigma^{1/2}\|^2.$$

Finally, if $v \to \infty$, we obtain that $\theta_0^T (XX^T)^{-1} X_0 v^2 \to 0.$
Lemma 3. When \((\log n)\sqrt{n/d} \to 0\), with probability tending to 1, the smallest eigenvalue of \(XX^T\) is in \(\Theta(d)\).

Proof of Lemma 3. Assume \(\Sigma = I\) for simplicity. Denote \(b \in (0, 1)\). Since \((\log n)\sqrt{n/d} \to 0\), we append \((bd - n)\) i.i.d samples of \(x\) after \(X\) and denote the new data matrix as \(Z\). Based on Bai and Yin (2008), the smallest eigenvalue of \(ZZ^T/d\) converges to \((1 - \sqrt{b})^2\), and the largest eigenvalue converges to \((1 + \sqrt{b})^2\). Since \(\lambda_{\min}(XX^T/d) \geq \lambda_{\min}(ZZ^T/d)\) and \(\lambda_{\max}(XX^T/d) \leq \lambda_{\max}(ZZ^T/d)\), we conclude that \(\lambda_{\min} = \Theta(d)\) in probability.

Proof of Theorem 4. Assume \(\nu\) is constant for simplicity. Denote \(\theta^{OP1}(t)\) and \(\theta^{OP2}(t)\) satisfy

\[
\begin{align*}
\theta^{OP1}(0) &= 0, \\
\theta^{OP1}(t + 1) &= \theta^{OP1}(t) - \eta \nabla \hat{R}_\xi(\theta^{OP1}(t), 0),
\end{align*}
\]

and

\[
\begin{align*}
\theta^{OP2}(0) &= 0, \\
\theta^{OP2}(t + 1) &= \theta^{OP2}(t) - \eta \nabla \hat{R}_\xi(\theta^{OP2}(t), \epsilon).
\end{align*}
\]

For \(\theta^{OP1}(t)\), i.e. standard training, when \(\eta\) is small enough such that the largest eigenvalue of \(|XX^T| \) is smaller than 1, then

\[
\begin{align*}
\theta^{OP1}(t) &= X^T(I - (1 - \eta XX^T/n)^t)(XX^T)^{-1}y, \\
X\theta^{OP1}(t) &= y - (1 - \eta XX^T/n)^t y,
\end{align*}
\]

which means that \(|X\theta^{OP1}(t) - y|\) monotonically decreases in \(t\). Solving \(|(I - \eta XX^T/n)^t y|/\sqrt{n} = 1/\sqrt{\log n}\), we obtain \(\eta T = O(\frac{1}{n^2} \log \log n)\).

Denote \(z = X\theta\), and assume \(\theta = \theta^{OP1}(t)\) for some \(t > 0\) and \(|\theta| = O(\sqrt{n/d})\). Since \(z - y = (I - \eta XX^T/n)^t y, |X^T(z - y)| \to c(p, n, t)\sqrt{d}||z - y||\) for some function \(c(p, n, t)\) which is finite and bounded away from zero. As a result, for \(\epsilon\) with probability tending to 1, for the difference between \(\nabla \hat{R}_\xi(\theta, \epsilon)\) and \(\nabla \hat{R}_\xi(\theta, 0)\), it becomes

\[
\begin{align*}
\|\nabla \hat{R}_\xi(\theta, \epsilon) - \nabla \hat{R}_\xi(\theta, 0)\| &\leq \frac{2\epsilon}{n} \frac{|\theta|^2}{\sqrt{|\theta|^2 + \xi^2}} X^T \text{sgn}(z - y) + \frac{2\epsilon}{n} \frac{\theta}{\sqrt{|\theta|^2 + \xi^2}} ||z - y||_1 \\
&+ \frac{2\epsilon^2}{\sqrt{|\theta|^2 + \xi^2}} \\
&= \left\| \nabla \hat{R}_\xi(\theta, 0) \right\| \left( \sqrt{\frac{n}{d}} + \frac{n^{3/2}/d}{\||z - y|| + ||z - y||} \right) \\
&= \left\| \nabla \hat{R}_\xi(\theta, 0) \right\| \left( \frac{n^{1/2}/d}{||z - y||} \right). \tag{14}
\end{align*}
\]

Using induction, we have

\[
\left\| \theta^{OP2}(t) - \theta^{OP1}(t) \right\| = O \left( \frac{t\eta \sqrt{n}}{||X\theta^{OP1}(t) - y||} \right).
\]

When \((\log n)\sqrt{n/d} \to 0\), for any \(t \leq T\),

\[
\left\| \theta^{OP2}(t) - \theta^{OP1}(t) \right\| = o(||\theta^{OP1}(t)||).
\]

Finally,

\[
\begin{align*}
X\nabla \hat{R}_\xi(\theta, \epsilon) &= \frac{2XX^T(z - y)}{n} + \frac{2\epsilon}{n} \frac{|\theta|^2}{\sqrt{|\theta|^2 + \xi^2}} XX^T \text{sgn}(z - y) \\
&+ \frac{2\epsilon}{n} \frac{z}{\sqrt{|\theta|^2 + \xi^2}} ||z - y||_1 + 2\epsilon^2 z \frac{|\theta|^2}{||\theta|^2 + \xi^2}.
\end{align*}
\]
Since with probability tending to 1, \( \|XX^T \text{sgn}(z - y)\| = O(d/\sqrt{n}) \). Thus, when \( \|XX^T (z - y)\| = \Theta(d/\sqrt{n}) \) and \( \|\theta\| = O(\sqrt{n/d}) \), with probability tending to 1, we have
\[
\left\lVert \frac{2\epsilon}{n} \sqrt{\frac{\|\theta\|^2 + \xi^2}{2}} \|z - y\|_1 + 2\epsilon^2 \frac{\|\theta\|^2}{\|\theta\|^2 + \xi^2} \right\rVert = o\left( \left\lVert \frac{XX^T (z - y)}{n} \right\rVert \right),
\]
and
\[
\left\lVert \frac{2\epsilon}{n} \frac{\|\theta\|^2}{\|\theta\|^2 + \xi^2} XX^T \text{sgn}(z - y) \right\rVert = \left\lVert \frac{XX^T (z - y)}{n} \right\rVert O\left( \sqrt{\frac{n}{d}} \|z - y\| \right).
\]
The statement \( \|XX^T (z - y)\| = \Theta(d/\sqrt{n}) \) for any \( z \) holds based on Lemma 3. As a result, one can use induction to show that \( \|X_{\theta^{OPT}}(t) - y\| \) decreases in \( t \) when \( \|z - y\| > \sqrt{n/\log n} \) and \((\log n) \sqrt{n/d} \to 0 \), and the statement in Theorem 4 holds.

\[\square\]

F Proof for \( \mathcal{L}_\infty \) Attack

Proof of Theorem 8. based on \( \mathcal{L}_\infty \) norm, the loss becomes
\[
\mathcal{L}_\infty(\theta) = \sum_{i=1}^{n} l_i \left( x_i^T \theta + \delta A_i^\infty (\theta, x_i, y_i) \right) \|\theta\|_\infty, y_i \right).
\]
We fix \( \theta_1 \), then the weight with minimum \( \mathcal{L}_2 \) norm becomes
\[
\theta^*_1(y) = X_{\theta^*_1}^T \left(X_{\theta^*_1}X_{\theta^*_1}^T\right)^{-1}(y - X_1).
\]
Thus
\[
\min_{X_{\theta^*_1} = y} \|\theta\|_\infty, v \leq \min_{\theta_1} \frac{\|\theta^*_1(y - \theta_1X_1)\|_2^2 + \theta_1^2 + 2\|\theta_1\|\|\theta^*_1(y - \theta_1X_1)\|_2}{\|\theta_1\|_v} = \min_{\theta_1} \|\theta_1\| + 2\|\theta^*_1(y - \theta_1X_1)\|_2 + \frac{\|\theta^*_1(y - \theta_1X_1)\|_2^2}{\|\theta_1\|_v} \leq c\|\theta^*_1(y)\|_2/v
\]
for some \( c > 0 \). Therefore, based on Lemma 1 we have \( \min_{X_{\theta = y}} \|\theta\|_\infty, v \to 0 \).

\[\square\]

G Proofs for high-dimensional sparse model

Proof of Theorem 7. Assume \( v = O(1) \) first. Denote
\[
\frac{1}{n} \sum_{i=1}^{n} l_i(f_\theta(x_i + A_\epsilon(f_\theta, x_i, y_i)), y_i) + \lambda \|\theta\|_1 := \frac{1}{n} \sum_{i=1}^{n} l_i(\theta, x_i, y_i) + \lambda \|\theta\|_1.
\]
Since \( \hat{\theta} \) minimizes the empirical penalized loss function, take \( \Delta = \hat{\theta} - \theta^* \), we have
\[
\frac{1}{n} \sum_{i=1}^{n} l_i(\hat{\theta}, x_i, y_i) - \frac{1}{n} \sum_{i=1}^{n} l_i(\theta^*, x_i, y_i) \leq \lambda \|\theta^*\|_1 - \lambda \|\hat{\theta}\|_1 \leq \lambda (\|\Delta_S\|_1 - \|\Delta_{S^c}\|_1).
\]
Moreover, the structure of \( l \) implies that it is a convex function, thus
\[
\Delta^T \frac{1}{n} \sum_{i=1}^{n} l'_i(\theta^*, x_i, y_i) \leq \lambda (\|\Delta_S\|_1 - \|\Delta_{S^c}\|_1).
\]
Further,
\[
\lambda (\|\Delta_S\|_1 - \|\Delta_{S^c}\|_1) \geq - \Delta^T \frac{1}{n} \sum_{i=1}^{n} l'_i(\theta^*, x_i, y_i) \geq - \|\Delta\|_1 \left\lVert \frac{1}{n} \sum_{i=1}^{n} l'_i(\theta^*, x_i, y_i) \right\rVert_\infty.
\]
Since $\theta^*$ is fixed, we can figure out that with probability tending to 1,
\[ \left\| \frac{1}{n} \sum_{i=1}^{n} l_i'(\theta^*, x_i, y_i) \right\|_\infty \leq c_1 \sqrt{\frac{s \log d}{n}} \]
for some constant $c_1 > 0$. Consequently, as our choice of $M$ is large enough, with probability tending
to 1,
\[ \| \Delta S \|_1 \geq \frac{\lambda - c_1}{\lambda + c_1} \| \Delta S^* \|_1. \]
As a result, from Lemma 4, we know that with probability tending to 1, for some constant $c_2 > 0$,
\[ \lambda \| \Delta S \|_1 \geq \lambda \| \Delta S \|_1 - \lambda \| \Delta S^* \|_1 \]
\[ \geq \Delta^\top \frac{1}{n} \sum_{i=1}^{n} l_i'(\theta^*, x_i, y_i) - c_2 \| \Delta \|_1 \sqrt{\frac{s \log d}{n}} + c^2 \left( 1 - \frac{2}{\pi} \right) \| \Delta \|^2. \]
Therefore,
\[ \lambda \| \Delta S \|_1 + (c_1 + c_2) \| \Delta \|_1 \sqrt{\frac{s \log d}{n}} \geq c^2 \left( 1 - \frac{2}{\pi} \right) \| \Delta \|^2 \]
\[ \geq c^2 \left( 1 - \frac{2}{\pi} \right) \| \Delta S \|_1^2, \]
hence with probability tending to 1,
\[ \| \Delta \|_1 = O(\lambda), \| \Delta \|_2 = O(\lambda). \]
Finally, using similar steps as in (15), we can observe that
\[ l_i(\hat{\theta}, x_i, y_i) - l_i(\theta^*, x_i, y_i) \]
\[ \leq \Delta^\top \frac{\partial l_i(\theta^*, x_i, y_i)}{\partial \theta^*} + (\Delta^\top x_i)^2 + c^2 \| \Delta \|^2 \]
\[ + 2c(\| \theta^* + \Delta \| - \| \theta^* \|) \Delta^\top x_i + 2c \| \theta^* + \Delta \| \| \theta^* \| - (\theta^* + \Delta)^\top \theta^* \| y_i - x_i^\top \theta^* \|. \]
Taking expectation on $(x_i, y_i)$, it becomes
\[ \mathbb{E} \left( l_i(\hat{\theta}, x_i, y_i) - l_i(\theta^*, x_i, y_i) \right) \]
\[ \leq (1 + c^2) \| \Delta \|^2 + O \left( \| \theta^* + \Delta S \| \| \theta^* \| - (\theta^* + \Delta)^\top \theta^* \right) + O (\| \Delta S^* \|) = O(\| \Delta \|). \]
Therefore, if $\lambda \to 0$, with probability tending to 1,
\[ R_0(\hat{\theta}, \epsilon) - R_0(\theta^*, \epsilon) \to 0. \]
To extend for general $v$, similar with Theorem 1, the change on $v$ does not affect the convergence
property of $\hat{\theta}$ after adjustment w.r.t. $v$. \qed

**Lemma 4.** Under the conditions in Theorem 7, when $\| \Delta S^* \|_1 \leq c \| \Delta S \|_1$, with probability tending
to 1, for some $c_3 > 0$,
\[ \frac{1}{n} \sum_{i=1}^{n} l_i(\hat{\theta}, x_i, y_i) - l(\theta^*, x_i, y_i) \]
\[ \geq \frac{1}{n} \sum_{i=1}^{n} \Delta^\top \frac{\partial l_i}{\partial \theta} + (\Delta^\top x_i)^2 - \| \Delta \|^2 + c^2 (1 - 2/\pi) \| \Delta \|^2 - c_3 \| \Delta \|_1 \sqrt{\frac{s \log d}{n}}, \]
and
\[ \frac{1}{\| \Delta S \|_1^2} \left| \frac{1}{n} \sum_{i=1}^{n} (\Delta^\top x_i)^2 - \| \Delta \|^2 \right| \to 0. \]
Proof. Assume $x^T \theta_2 - y > 0$, then for any $\theta_1$,

\[
(x^T \theta_1 - y)^2 + 2\epsilon \|\theta_1\|_2 (x^T \theta_1 - y) + \epsilon^2 \|\theta_1\|^2 - (x^T \theta_2 - y)^2 - 2\epsilon \|\theta_2\|(x^T \theta_2 - y) - \epsilon^2 \|\theta_2\|^2 \\
\geq (x^T \theta_1 - y)^2 + 2\epsilon \|\theta_1\|_2 (x^T \theta_1 - y) + \epsilon^2 \|\theta_1\|^2 - (x^T \theta_2 - y)^2 - 2\epsilon \|\theta_2\|(x^T \theta_2 - y) - \epsilon^2 \|\theta_2\|^2 \\
= 2(\theta_1 - \theta_2)^T x(x^T \theta_2 - y) + (\theta_1 - \theta_2)^T xx^T (\theta_1 - \theta_2) + 2\epsilon^2 (\theta_1 - \theta_2)^T \theta_2 + \epsilon^2 \|\theta_1 - \theta_2\|^2 \\
\quad + 2\epsilon \|\theta_1\|(x^T \theta_1 - y) - 2\epsilon \|\theta_2\||(x^T \theta_2 - y) - 2\epsilon (\theta_1 - \theta_2) \left( \frac{\theta_2}{\|\theta_2\|} (x^T \theta_2 - y) + \|\theta_2\| x \right) \\
\quad + 2\epsilon (\theta_1 - \theta_2) \left( \frac{\theta_2}{\|\theta_2\|} (x^T \theta_2 - y) + \|\theta_2\| x \right),
\]

where

\[
\|\theta_1\|(x^T \theta_1 - y) - \|\theta_2\|(x^T \theta_2 - y) - (\theta_1 - \theta_2) \left( \frac{\theta_2}{\|\theta_2\|} (x^T \theta_2 - y) + \|\theta_2\| x \right)
\]

\[
= \|\theta_1\|(x^T \theta_2 - y) + \|\theta_1\|((\theta_1 - \theta_2)^T x - \theta_2) \left( \frac{\theta_2}{\|\theta_2\|} (x^T \theta_2 - y) + \|\theta_2\| x \right) + \theta_2^T x \|\theta_2\|
\]

\[
= \|\theta_1\|\|\theta_2\| - \|\theta_1\|\theta_2 \left( x^T \theta_2 - y \right) + ((\|\theta_1\| - \|\theta_2\|))(\theta_1 - \theta_2)^T x
\]

\[
\geq (\|\theta_1\| - \|\theta_2\|)(\theta_1 - \theta_2)^T x
\]

Thus if $x_i^T \theta^* - y_i > 0$,

\[
l_c(\theta, x_i, y_i) - l_c(\theta^*, x_i, y_i) \geq \Delta^T \frac{\partial l_c(\theta^*, x_i, y_i)}{\partial \theta^*} + (\Delta^T x_i)^2 + \epsilon^2 \|\Delta\|^2 + 2\epsilon \|\theta^* + \|\theta^*\| - \|\theta^*\|)\Delta^T x_i (15)
\]

When $x^T \theta_2 - y < 0$,

\[
\|\theta_1\|(y - x^T \theta_1) - \|\theta_2\|(y - x^T \theta_2) - (\theta_1 - \theta_2) \left( \frac{\theta_2}{\|\theta_2\|} (y - x^T \theta_2) + \|\theta_2\| x \right)
\]

\[
= \|\theta_1\|(y - x^T \theta_2) - \|\theta_1\|((\theta_1 - \theta_2)^T x - \theta_2) \left( \frac{\theta_2}{\|\theta_2\|} (y - x^T \theta_2) + \|\theta_2\| x \right) + \theta_2^T x \|\theta_2\|
\]

\[
= \|\theta_1\|\|\theta_2\| - \|\theta_1\|\theta_2 \left( y - x^T \theta_2 \right) - ((\|\theta_1\| - \|\theta_2\|))(\theta_1 - \theta_2)^T x
\]

\[
\geq -((\|\theta_1\| - \|\theta_2\|))(\theta_1 - \theta_2)^T x.
\]

Thus for any $x_i$,

\[
l_c(\theta, x_i, y_i) - l_c(\theta^*, x_i, y_i) \geq \Delta^T \frac{\partial l_c(\theta^*, x_i, y_i)}{\partial \theta^*} + (\Delta^T x_i)^2 + \epsilon^2 \|\Delta\|^2 + 2\epsilon \|\theta^* + \|\theta^*\| - \|\theta^*\|)\Delta^T x_i.
\]

8
As a result,
\[
\frac{1}{n} \sum_{i=1}^{n} l_{e}(\tilde{\theta}, x_i, y_i) - l(\theta^{*}, x_i, y_i) \\
\geq \frac{1}{n} \sum_{i=1}^{n} \Delta^{\top} \frac{\partial l_{e}}{\partial \theta} + (\Delta^{\top} x_i)^2 + \varepsilon^2 \|\Delta\|^2 + 2\varepsilon(\|\theta^{*} + \Delta\| - \|\theta^{*}\|)\Delta^{\top} x_i \text{sgn}(x_i^{\top} \theta^{*} - y_i) \\
= \frac{1}{n} \sum_{i=1}^{n} \Delta^{\top} \frac{\partial l_{e}}{\partial \theta} + (\Delta^{\top} x_i)^2 + \varepsilon^2 \|\Delta\|^2 + 2\varepsilon(\|\theta^{*} + \Delta\| - \|\theta^{*}\|)(\Delta_{S}^{\top} x_{i,S} + \Delta_{S}^{\top} x_{i,S^{c}}) \text{sgn}(x_i^{\top} \theta^{*} - y_i) \\
\geq \frac{1}{n} \sum_{i=1}^{n} \Delta^{\top} \frac{\partial l_{e}}{\partial \theta} + (\Delta^{\top} x_i)^2 + \varepsilon^2 \|\Delta\|^2 \\
-2\varepsilon \|\theta^{*} + \Delta\| - \|\theta^{*}\| \left( \|\Delta_{S^{c}}\|_1 \left\| \frac{1}{n} \sum_{i=1}^{n} x_{i,S^{c}} \text{sgn}(x_i^{\top} \theta^{*} - y_i) \right\| \right) + O_{p} \left( \|\Delta_{S}\|_1 \sqrt{\frac{s \log s}{n}} \right) \\
+ 2\varepsilon(\|\theta^{*} + \Delta\| - \|\theta^{*}\|)\Delta_{S}^{\top} x_{i,S} \text{sgn}(x_i^{\top} \theta^{*} - y_i) \\
\geq \frac{1}{n} \sum_{i=1}^{n} \Delta^{\top} \frac{\partial l_{e}}{\partial \theta} + (\Delta^{\top} x_i)^2 + \varepsilon^2 \|\Delta\|^2 \\
-2\varepsilon \|\theta^{*} + \Delta\| - \|\theta^{*}\| \left( \|\Delta_{S^{c}}\|_1 \left\| \frac{1}{n} \sum_{i=1}^{n} x_{i,S^{c}} \text{sgn}(x_i^{\top} \theta^{*} - y_i) \right\| \right) + O_{p} \left( \|\Delta_{S}\|_1 \sqrt{\frac{s \log s}{n}} \right) \\
-2\varepsilon \|\Delta\| \sqrt{\frac{2}{\pi}} \left| \Delta^{\top} \frac{\partial}{\partial \theta} \sqrt{\|\theta - \theta_{0}\|^2 + \sigma^2} \right| \\
\geq \frac{1}{n} \sum_{i=1}^{n} \Delta^{\top} \frac{\partial l_{e}}{\partial \theta} + (\Delta^{\top} x_i)^2 - \|\Delta\|^2 + \varepsilon^2 (1 - 2/\pi) \|\Delta\|^2 - O_{p} \left( \|\Delta\|_1 \sqrt{\frac{s \log d}{n}} \right). \quad (16) \tag{16}
\]

From (16) to (17),
\[
\|\Delta\|^2 - 2\varepsilon\|\Delta\| \sqrt{\frac{2}{\pi}} \left| \Delta^{\top} \frac{\partial}{\partial \theta} \sqrt{\|\theta - \theta_{0}\|^2 + \sigma^2} \right| \\
\geq \left( \|\Delta\| - \varepsilon \sqrt{\frac{2}{\pi}} \left| \Delta^{\top} \frac{\partial}{\partial \theta} \sqrt{\|\theta - \theta_{0}\|^2 + \sigma^2} \right| \right)^2 - \varepsilon^2 \frac{2}{\pi} \left| \Delta^{\top} \frac{\partial}{\partial \theta} \sqrt{\|\theta - \theta_{0}\|^2 + \sigma^2} \right|^2 \\
\geq -\varepsilon^2 \frac{2}{\pi} \|\Delta\|^2.
\]

When \( \|\Delta_{S^{c}}\|_1 \leq \varepsilon \|\Delta_{S}\|_1 \), since \( \varepsilon_{d,n} := \max_{i,j} |\tilde{\Sigma}_{i,j} - I_{i,j}| \to 0 \) in probability, we have with probability tending to 1,
\[
\frac{1}{\|\Delta_{S}\|_1^2} \left| \frac{1}{n} \sum_{i=1}^{n} (\Delta^{\top} x_i)^2 - \|\Delta\|^2 \right| = \frac{1}{\|\Delta_{S}\|_1^2} \left| \Delta^{\top} (I - \tilde{\Sigma}) \Delta \right| \leq \frac{1}{\|\Delta_{S}\|_1^2} \|\Delta\|_1^2 \varepsilon_{d,n} \to 0.
\]

\( \square \)